

Système de Fichiers, inodes et liens

Gorgoumack SAMBE

Université de Ziguinchor

Version 1.0¹

¹Decembre 2012



Objectifs

A l'issue de ce chapitre, l'apprenant doit être capable de :

- décrire les **caractéristiques d'un fichier**
- distinguer les **types de fichier** sur Linux
- décrire la **structure physique des disques**
- distinguer **formatage** physique et logique d'un disque.
- décrire le **système de fichier virtuel** de Linux.
- comprendre le concept d'**inoeud (inode)**
- distinguer **liens durs** et **liens symboliques**



Plan

- 1 Fichier
- 2 Organisation physique d'un disque
- 3 Organisation logique d'un disque
- 4 Système de fichiers Virtuel(VFS)
- 5 Inodes, liens durs et liens symboliques



Plan

- 1 Fichier
- 2 Organisation physique d'un disque
- 3 Organisation logique d'un disque
- 4 Système de fichiers Virtuel(VFS)
- 5 Inodes, liens durs et liens symboliques



Fichier

Définition et caractéristiques

Définition

Collection d'informations numériques réunies sous un même nom, enregistrée sur un support de stockage(disque dur, CD-ROM, ...).

- **Caractéristiques d'un fichier :**

- ▶ **nom** : composé généralement du **nom** même du fichier et d'une **extension**.
Exemples : monfic.txt, tonfic.doc, prog.c, test.pas, test.exe.
- ▶ **type**
- ▶ **format**
- ▶ **taille**
- ▶ **droits** sur le fichier (**propriétaires**)
- ▶ ...



Nom de fichier

- **Windows** associe **extensions** de fichier et **format** de fichier
⇒ **logiciels par défaut**.
- **Linux** ne s'attache pas à l'extension des fichiers
(<http://librefan.eu.org/node/71>).
- Certains **environnements de bureaux** Linux fonctionne dans ce contexte comme Windows (KDE).

| Nommage des fichiers | | | |
|----------------------|---------------------|---------------|---------------------|
| SE | Sensible à la casse | Long max. nom | Long max. extension |
| UNIX/Linux | Oui | 255 | 255 |
| MS-DOS | Non | 8 | 3 |
| Windows 95 | Non | ... | ... |
| Windows 98 | Non | ... | ... |
| Windows NT | Non | 255 | 255 |
| Windows 2000 | Non | 255 | 255 |



Type de fichier

• Sur Windows

- 1 **répertoires** : fichiers conteneurs de références à d'autres fichiers.
- 2 **fichiers** : fichiers texte, fichiers exécutables ...

• Sur Linux

- 1 **fichiers répertoires** : fichiers conteneurs de références à d'autres fichiers.
- 2 **fichiers normaux** : fichiers ASCII, fichiers binaires.
- 3 **fichiers spéciaux(bloc /caractères)** : situés dans /dev, ce sont les points d'accès préparés par le système aux périphériques. Le montage va réaliser une correspondance de ces fichiers spéciaux vers leur répertoire "point de montage".
- 4 **fichiers liens symboliques** : fichiers qui ne contiennent qu'une référence (un pointeur) à un autre fichier.



Format de fichier

- Le format désigne la **manière d'encoder** le fichier.
 - ▶ **format propriétaire** : spécifications contrôlées par une entité privée.
 - **fermé** : spécifications non publiques.
Exemple : format word, format excel, wma, wmv, ...
 - **ouvert** : spécifications publiques.
Exemple : mp3, mpeg, postscript, PDF ...
 - ▶ **format libre** : spécifications techniques publiques, sans restrictions d'accès ni de mise en œuvre.
Exemple : OGM-vorbis, odf, HTML, ...
- Quelques ressources
 - ▶ http://fr.wikipedia.org/wiki/Format_ouvert
 - ▶ <http://www.halpanet.org/?q=content/formats-libres-propri-taires>
 - ▶ <http://fabiwan.kenobi.free.fr/pgid/formats.html>



Plan

- 1 Fichier
- 2 Organisation physique d'un disque
- 3 Organisation logique d'un disque
- 4 Système de fichiers Virtuel(VFS)
- 5 Inodes, liens durs et liens symboliques



Structure physique

- Tout le mécanisme est enfermé dans un boîtier hermétique

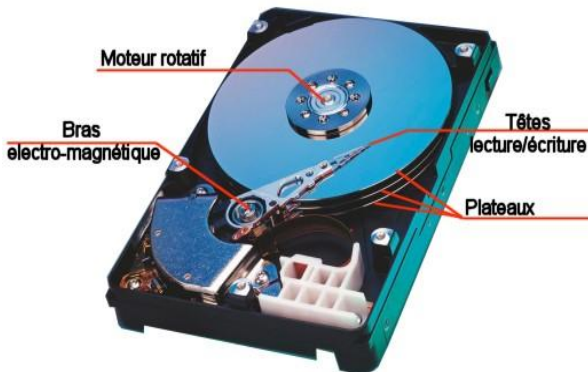


Figure: Le disque dur



Formatage de bas niveau

Plateaux, pistes et cylindres

Chaque **plateau** est constitué de **pistes** alignées de manière à former des **cylindres**.

formatage physique

- Inscription de **secteurs** sur les **pistes**.
- Un secteur contient 512 octets d'information
- Généralement **effectué en usine**

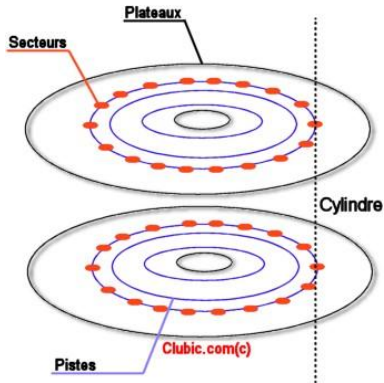


Figure: formatage physique



Formatage de bas niveau

Les secteurs

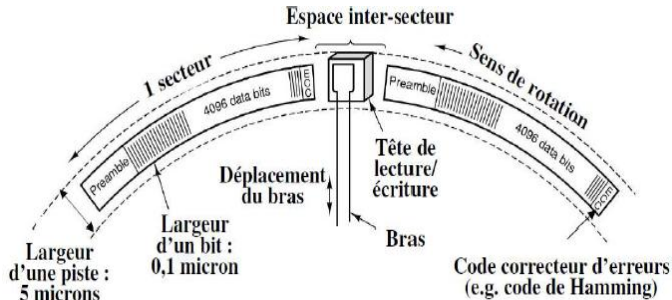
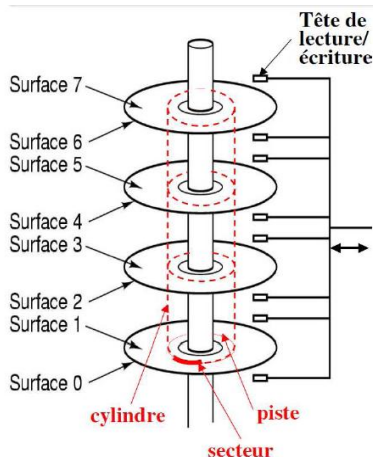


Figure: Secteurs



Fonctionnement du disque Winchester



- Les têtes se déplacent ensemble et se placent sur un cylindre.
- Le disque a la **densité** de 100 000 bits/cm et tourne en faisant 60-120 tours par seconde
- **Débit** typique : ~ 20 Mo/s
- Le **temps d'accès** est dominé par le temps de positionnement (placement de la tête sur la bonne piste + attente du début du bon secteur): ~ 10 ms
- Adresse d'un secteur: (n° tête, n° piste, n° secteur)



Plan

- 1 Fichier
- 2 Organisation physique d'un disque
- 3 Organisation logique d'un disque
- 4 Système de fichiers Virtuel(VFS)
- 5 Inodes, liens durs et liens symboliques



formatage de haut niveau

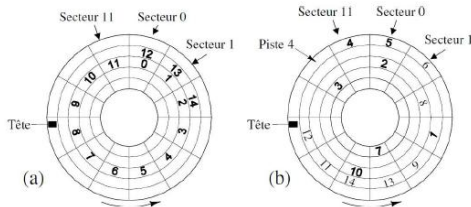
- Désigne l'**inscription d'un système de fichiers** sur un disque ou une partition.
- Une **partition** de disque dur désigne une partie indépendante des autres parties du disque dur et possédant un système de fichier.
- Chaque SE développe généralement son propre système de fichier \Rightarrow **le système de fichier dépend du SE utilisé.**

Système de fichiers

Façon de stocker et d'**organiser** les informations (fichiers) sur les supports de stockage.



Exemples d'organisation



- Chaque fichier est enregistré sur un nombre de secteurs suffisant, le dernier secteur n'étant pas souvent plein.
- Avec l'allocation consécutive (a) (CD-ROM par exemple) il suffit de connaître la taille et l'emplacement du début de chaque fichier.
- Avec l'allocation non consécutive (b) on a besoin d'une liste de secteurs pour chaque fichier.
- Grâce au Système d'Exploitation l'utilisateur ne voit qu'un fichier.



Exemples de système de fichier

| Système | Nom | Taille max.fichier | Taille max. partition | Journalisé? | Gestion des droits d'accès? |
|---------|---------------------|--------------------|-----------------------|-------------|-----------------------------|
| Linux | ext2fs ² | 2 TO | 4 TO | Non | Oui |
| | ext3fs | 2 TO | 4 TO | Oui | Oui |
| | ext4fs | 16 TO | 1 EO | Oui | Oui |
| Windows | FAT ³ | 2 GO | 2 GO | Non | Non |
| | FAT32 | 4 GO | 8 TO | Non | Non |
| | NTFS ⁴ | 16 TO | 256 TO | Oui | Oui |

- http://doc.ubuntu-fr.org/systeme_de_fichiers
- http://fr.wikipedia.org/wiki/Liste_des_syst%C3%A8mes_de_fichiers

²Extended File System

³File Allocation Table

⁴New Technology File System



Plan

- 1 Fichier
- 2 Organisation physique d'un disque
- 3 Organisation logique d'un disque
- 4 Système de fichiers Virtuel(VFS)**
- 5 Inodes, liens durs et liens symboliques



Système de fichiers Virtuel

Système de fichiers Virtuel

Désigne l'**organisation hiérarchique** de catégories(une catégorie peut contenir d'autres catégories) de **fichiers** appelées par analogie **répertoires** ou dossiers en vue de faciliter la localisation des fichiers pour l'utilisateur.

Système de fichiers / Système de fichiers virtuel

Le système de fichier désigne l'**organisation logique** des fichiers sur le **disque** (formatage logique) mais aussi la **vision de l'utilisateur** sur une organisation virtuelle des fichiers (Système de fichiers virtuel)



FileSystem Hierarchy Standard (FHS)

Qu'est ce que c'est?

Norme qui définit l'**arborescence** et le **contenu des principaux répertoires** des systèmes de fichiers GNU/Linux et de la plupart des systèmes UNIX.

- **quatre catégories de fichiers** : partageables / non partageables, statiques / variables.
 - ▶ données partageables : ce qui peut être partagé entre plusieurs machines différentes. Exemple : /home
 - ▶ données variables : tout ce qui change sans l'intervention de l'administrateur système.

| | partageables | non partageables |
|-----------|------------------------------|---------------------|
| statiques | /usr, /opt | /etc, /boot |
| variables | /var/mail, /var/spool/news / | /var/run, /var/lock |



Fichiers sur Linux

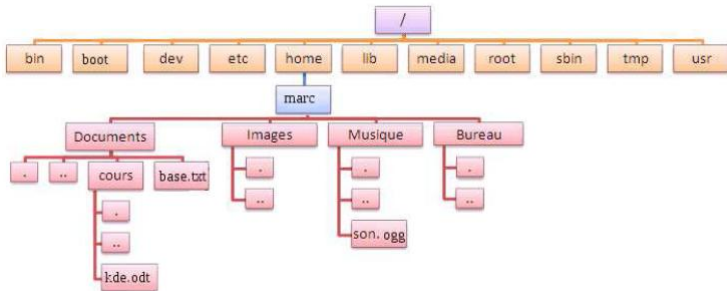
- Répertoires, fichiers normaux, fichiers spéciaux, liens symboliques.
- Le nom d'un fichier est une suite de caractères (sensible à la casse).
- Caractère ".": simple caractère dans le nom du fichier et non pas le caractère de séparation entre le nom et le type (notion d'extension)
- Les caractères invalides sous Dos (; + = [] , ' " * \ < > / ? : |) sont acceptés. Mais certains caractères sont à éviter dans les noms de fichiers: \, la tabulation, " ; " , caractères de contrôle (ESC , CTRL - C , etc.)



Arborescence

- Structure d'arbre.

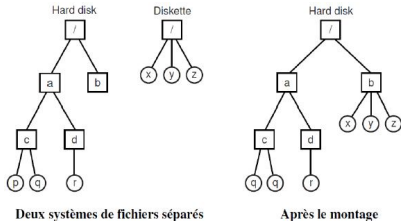
- ▶ La racine de l'arbre est la racine du système de fichier: /
- ▶ Les nœuds sont les répertoires et les feuilles sont les fichiers.



Arborescence

Montage

- De façon générale **tout est fichier**, **une seule arborescence**.
- Pas de notion de disque physique sous UNIX (au niveau utilisateur).
- Différents points de celle-ci sont rattachés à un système de fichiers : **Principe de montage**

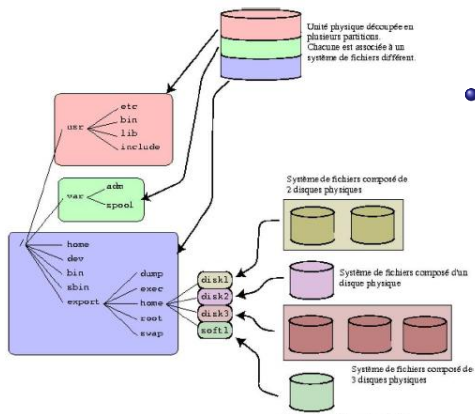


- mount -t iso9660 /dev/hdd /media/cdrom**
- Fichier **/etc/fstab**: montage automatique au démarrage du système.



Arborescence

Montage



• Le Système de Fichiers monté peut être:

- 1 Une **partition d'un disque physique**: une partie ou la totalité.
- 2 Un **ensemble de disques physiques**.
- 3 Un **système de fichier distant** résidant sur une autre machine (NFS).



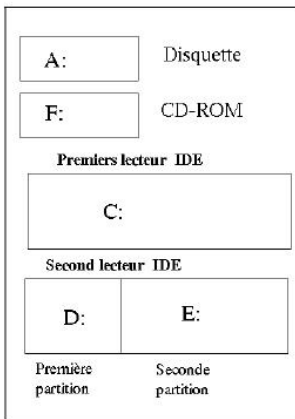
Principaux répertoires

- / - répertoire racine.
 - ▶ **/boot/** - Fichiers statiques du chargeur de démarrage(noyau,...).
 - ▶ **/dev/** - Fichiers de périphériques: disques, écran,....
 - ▶ **/etc/** - fichiers de configuration
 - ▶ **/home/** - Répertoires des utilisateurs
 - ▶ **/root/** - répertoire personnel de l'utilisateur root.
 - ▶ **/tmp/** - fichiers temporaires
 - ▶ **/mnt/** - point de montage des partitions temporaires
 - ▶ **/media/** - pour les points de montage
 - ▶ **/lib/** - Bibliothèques partagées essentielles et modules du noyau.
 - ▶ **/bin/** - Binaires de base pour tous les utilisateurs.
 - ▶ **/sbin/** - Binaires systèmes essentielles.
 - ▶ **/usr/** - hiérarchie secondaire.

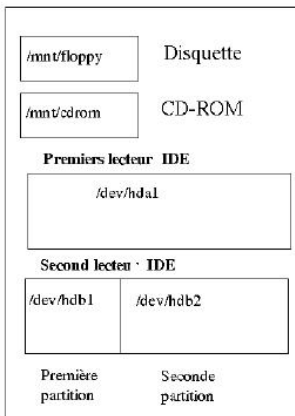


Comparatif avec Windows

Windows



Linux



Répertoire de travail

Répertoire de travail

- répertoire qui contient les fichiers dont se servent les commandes lorsque celles-ci ne spécifient aucun chemin (path) précis.
- A chaque moment, un répertoire joue le rôle du répertoire courant.
- Changer de répertoire signifie choisir un autre répertoire comme répertoire courant (working directory).
 - ▶ **cd** permet de changer de répertoire de travail.
 - ▶ **pwd** affiche à l'écran le répertoire de travail.

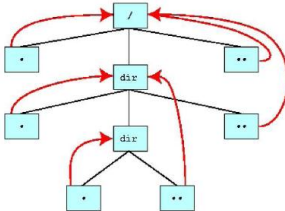


Répertoires particuliers

".", ".." et ~

Dans chaque répertoire du système, il y a toujours 2 sous-répertoires:

- . : Le point ".", fait référence au répertoire lui même.
- .. : Les deux points "..", font référence au répertoire père.



- Le symbole ~ (tilde) désigne le répertoire d'accueil(home) de l'utilisateur.
- Un fichier est caché si son nom commence par un point(".").



Chemin d'accès

Chemin absolu Vs chemin relatif

- On distingue deux types de chemins d'accès aux fichiers
 - ▶ Les **chemins d'accès absolus**: Pour tout fichier présent dans le système, il existe un chemin de la racine (/) et aboutissant à ce fichier. La liste des nœuds rencontrés identifie sans aucune ambiguïté le fichier. Le chemin ainsi parcouru s'appelle chemin absolu
 - **Commencent obligatoirement par /.**
 - ▶ Les **chemins d'accès relatifs**: La référence relative d'un fichier est la liste des nœuds rencontrés pour aboutir à ce fichier à partir de l'endroit où on se trouve (working directory).
 - **Ne commencent jamais par /.**

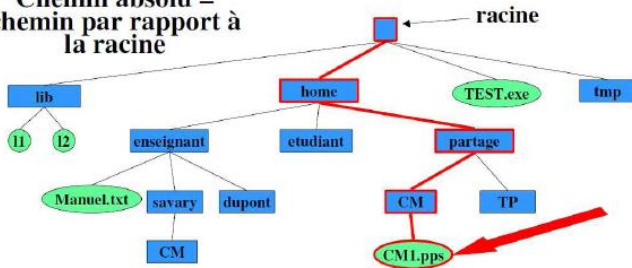


Chemin d'accès

Exemple : chemin absolu

- `/home/partage/CM/CM1.pps`

**Chemin absolu =
chemin par rapport à
la racine**



- Donner le chemin d'accès absolu du fichier Manuel.txt.
- Donner le chemin d'accès absolu du répertoire savary.

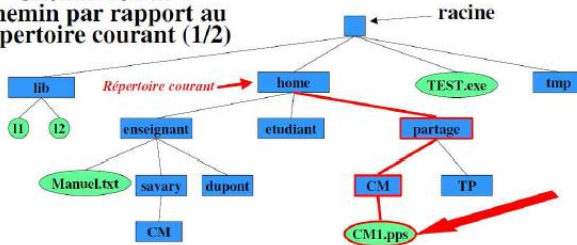


Chemin d'accès

Exemple : chemin relatif

- `./partage/CM/CM1.pps`

**Chemin relatif =
chemin par rapport au
répertoire courant (1/2)**



- Donner le chemin d'accès relatif du fichier Manuel.txt
- Donner le chemin d'accès relatif du dossier CM.

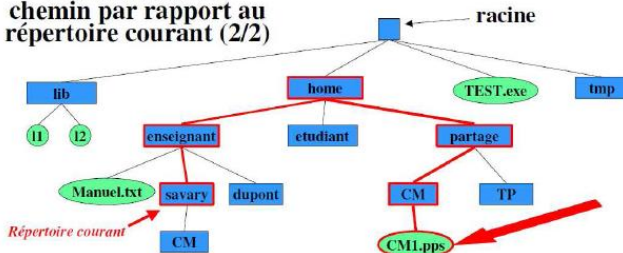


Chemin d'accès

Exemple : chemin relatif

- `../..partage/CM/CM1.pps`

**Chemin relatif =
chemin par rapport au
répertoire courant (2/2)**



- Donner le chemin d'accès relatif du fichier Manuel.txt.
- Donner le chemin d'accès relatif du fichier l1.



Plan

- 1 Fichier
- 2 Organisation physique d'un disque
- 3 Organisation logique d'un disque
- 4 Système de fichiers Virtuel(VFS)
- 5 Inodes, liens durs et liens symboliques



Qu'est ce qu'un i-noeud

- inode (index node, information node, i-noeud): structure de données contenant des **informations sur un fichier**.
- Á chaque **fichier** correspond un **numéro d'inode** (i-number) dans le SF dans lequel il réside, **unique au périphérique** sur lequel il est situé.
- Les informations stockées dans un inode sont :
 - ▶ Utilisateur propriétaire
 - ▶ Groupe propriétaire
 - ▶ Le type de fichier
 - "-": Fichier ordinaire. "d": Répertoire. "b": Fichier spécial mode bloc.
 - "c": Fichier spécial mode caractère. "n": Fichier spécial mode réseau.
 - "l": Lien symbolique. "p": Pour une fifo (pipe nommé, named pipe)
 - ▶ Droits d'accès (ugo * rwx)
 - ▶ Le nombre de références (liens physiques)
 - ▶ La taille du fichier
 - ▶ Adresses des blocs-disque contenant le fichier
 - ▶ Date de dernier accès
 - ▶ Date de dernière modification
- Toutes les informations fournies par la commande "**ls -l**" s'y trouvent, **SAUF** le nom.



Inode

Les fichiers

- Un **fichier** peut avoir plusieurs liens, donc **plusieurs noms**. Il suffit pour cela qu'ils pointent sur le même inode: celui du fichier.
- Le nom d'un fichier se trouve dans l'inode du répertoire conteneur.
- La racine du système de fichier (/) correspond à l'inode numéro 2.
- La commande `ls` affiche le contenu d'un répertoire
 - ▶ L'option `-i` affiche les numéros d'inode.
 - ▶ L'option `-a` affiche les fichiers cachés



Inode

Les répertoires

- Un répertoire est simplement un fichier, dont Le type est différent(**d**).
- Les blocs de données contiennent une **liste de couples**
(**nom**, **numéro d'inode**)
- Exemple : Si le répertoire contient deux fichiers **a.txt** et **b.sh**, il sera composé d'une liste de quatre couples:
(**.**, **43**), (**..**, **254**), (**a.txt**, **30**), (**b.sh**, **300**)



Liens

Un lien est un couple

(**nomDeFichier**, **numéro d'inode**)

1 Les liens physiques (durs)

- ▶ permet de donner plusieurs noms/chemin d'accès, à un même fichier **en pointant sur un même numéro d'inode** de fichier.
- ▶ Création d'un nouveau nom pour le même inode.
- ▶ Impossible de créer un lien physique vers un répertoire(boucle)
- ▶ Impossible de créer un lien physique vers un autre volume (partition)

2 Les liens symboliques (raccourcis)

- ▶ permet d'attribuer un autre nom/chemin d'accès à un fichier **en pointant sur un nom de fichier**.
- ▶ Création d'un nouvel inode de type lien symbolique qui contient le chemin vers le fichier "pointé".
- ▶ Il est possible d'avoir un lien symbolique vers un répertoire
- ▶ Il est possible d'avoir un lien symbolique vers une autre partition



Liens

1 Les liens physiques

- ▶ Un **compteur de références** permet de savoir combien de noms pointent sur le même inode.
- ▶ le fichier n'est supprimé que lorsque le dernier nom est effacé.
- ▶ commande de création de lien : *`ln cible nomDuLien`*

2 Les liens symboliques

- ▶ La suppression du fichier référencé n'entraîne pas la suppression du lien (mais produit un lien mort).
- ▶ la suppression du lien symbolique n'entraîne pas la suppression du fichier.
- ▶ commande de création de lien : *`ln -s cible nomDuLien`*

