



Système d'exploitation avancé

Système de fichiers

Pierre LEROY – leroy.pierre1@gmail.com

Sommaire

- I. Linux & généralités
 - II. Système de gestion de fichiers
 - III. Essentiel
 - IV. Conclusion
-

Définitions

- Un fichier se définit de la sorte :

ENSEMBLE

METADONNEES

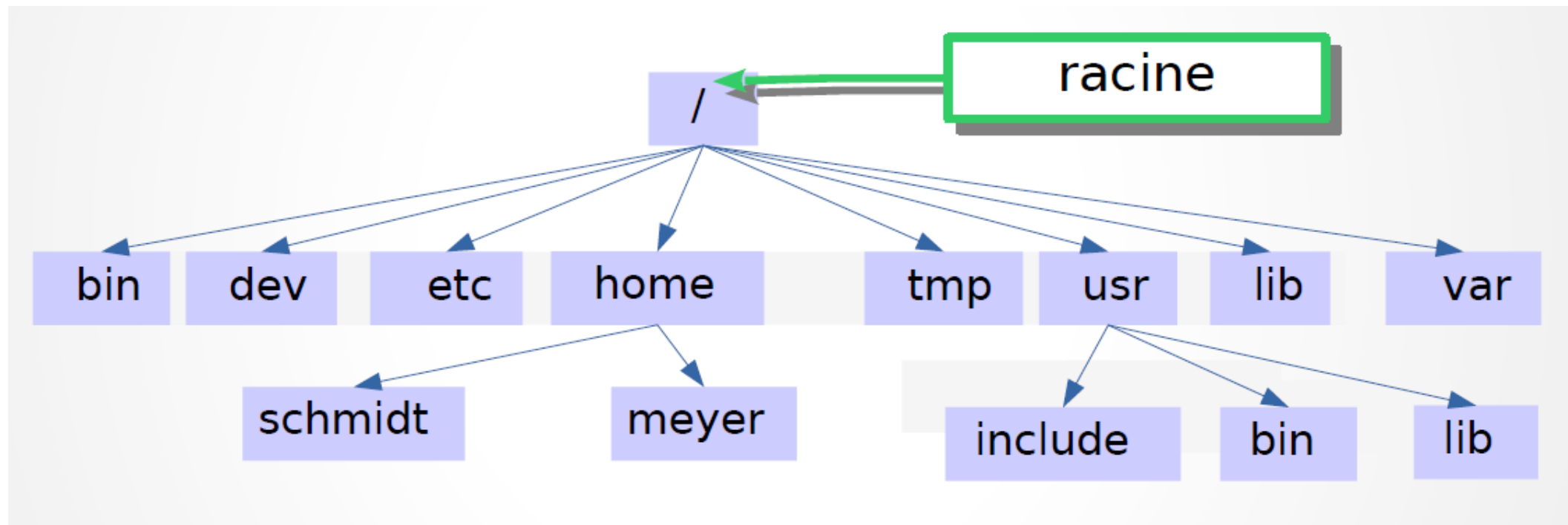
- Une suite d'octets
- Un ensemble d'attributs dépendant du (*FS*)
 - ✓ Nom
 - ✓ Taille
 - ✓ Type
 - ✓ Propriétaire
 - ✓ Permissions
 - ✓ ...
- Présent et persitable sur un support physique selon une méthodologie définie (*SGF*)
- Le stockage d'un fichier est réalisé par bloc

Définitions

- Le contenu d'un fichier est donc un sous-ensemble de blocs
 - <> Un sous-ensemble *précis* de blocs compose le contenu d'un fichier
- Le découpage en blocs sur un périphérique de stockage (disque dur, ram) implique:
 - ✓ La fragmentation interne du support : **non contiguïté des blocs de données**
 - ✓ La gestion de l'indexation des blocs
- L'ensemble des fichiers est structuré en arbre
 - ✓ L'élément racine de toute structuration étant : / (Linux) ou C:\ (Windows)

Généralités

- Partition principale sous Linux :
 - ✓ Respecte la *Filesystem Hierarchy Standard*



Généralités

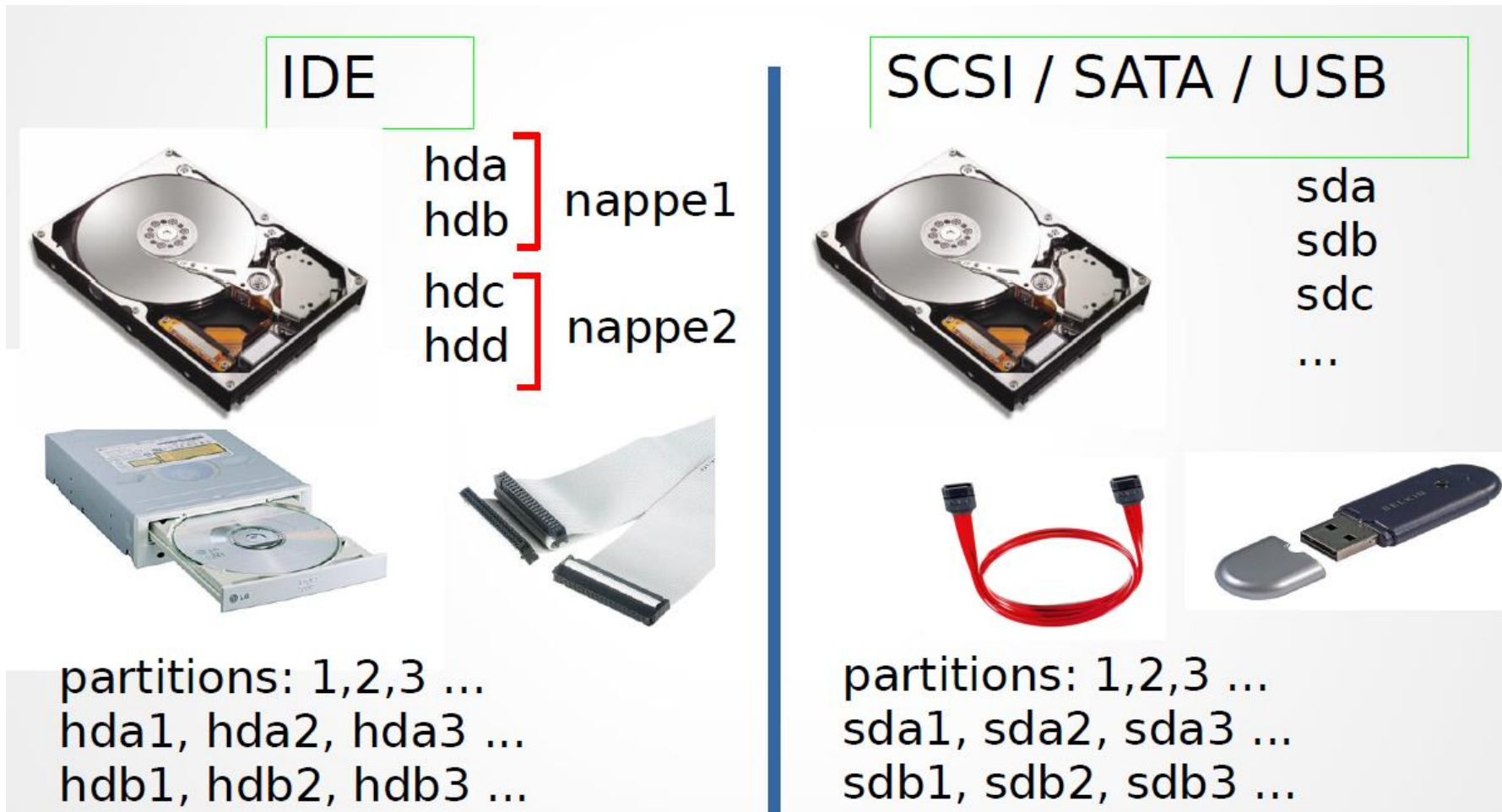
- Il existe de nombreux répertoires usuels dédiés à des tâches spécifiques :
 - ✓ **Fichiers de configuration** : /etc
 - ✓ **Commandes** : /bin, /sbin, /usr/bin, /usr/sbin ...
 - ✓ **Périphériques** : /dev (devices)
 - ✓ **Fichiers de log** : /var
 - ✓ **Applications** : /usr/local
 - ✓ **Librairies** : /lib, /usr/lib, ...
 - ✓ **Fichiers temporaires** : /tmp
 - ✓ **/boot** : fichiers de démarrage (noyau)
 - ✓ **/home** : fichiers des utilisateurs
 - ✓ **/root** : fichiers du super-administrateur
 - ✓ **/tmp** : répertoire temporaire
 - ✓ **/proc** : fichiers descriptifs du système

Généralités

- Il existe de nombreux fichiers usuels dédiés à des tâches spécifiques :
 - ✓ **Démarrage** : /etc/inittab, /etc/init.d, /etc/rcX.d
 - ✓ **Utilisateurs** : /etc/passwd, /etc/shadow, /etc/group
 - ✓ **Système de fichiers** : /etc/fstab, /etc/mtab
 - ✓ **Réseaux** : /etc/network/interfaces, /etc/services, /etc/inetd.conf, /etc/resolv.conf
 - ✓ **Librairies dynamiques** : /etc/ld.so.conf.d/*

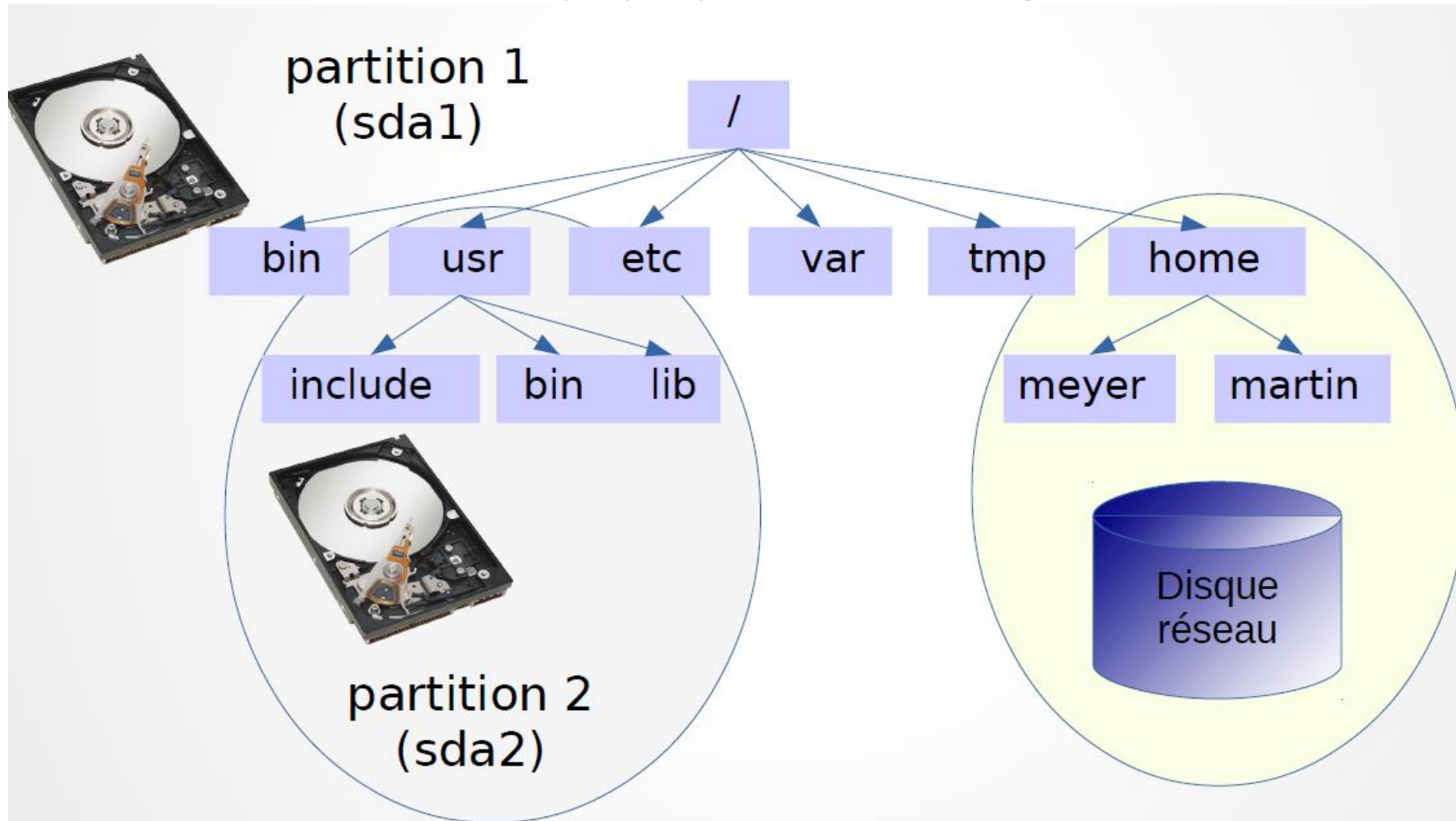
Généralités

- Représentation des périphériques de stockage sous linux : **/dev/xxx**



Généralités

- Corrélation éléments physiques de stockage & forêt de FS :



Sommaire

- I. Linux & généralités
 - II. Système de gestion de fichiers
 - III. Essentiel
 - IV. Conclusion
-

Définitions

- Un OS gère une forêt de fichier
 - ✓ La structuration de l'espace de stockage est appelé ***File System (FS)***
 - ✓ Multiples espaces de stockage implique potentiellement plusieurs FS différents
 - L'OS doit être capable de gérer plusieurs types répartis sur plusieurs support
- Un système de gestion de fichier gère
 - ✓ La **structuration** de l'espace de stockage appelé ***Formatage***
 - ✓ La **gestion** et l'**organisation** de l'espace de stockage : *Quid taille des blocs ?*
 - ✓ La **manipulation** des fichiers
 - ✓ La gestion des **métadonnées** des items
 - ✓ La **sécurité** des fichiers

Typologie

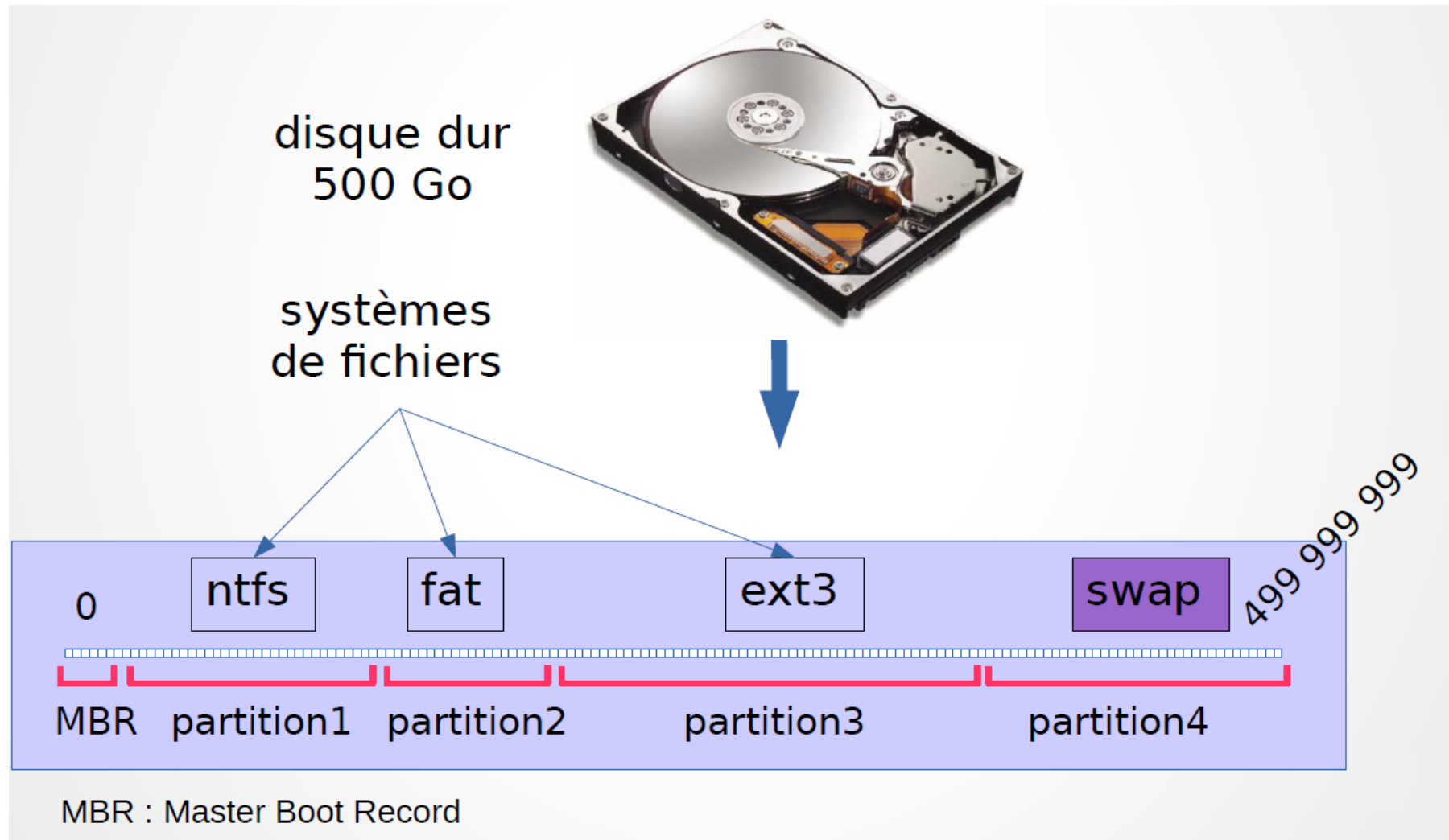
- Les systèmes de gestion de fichiers les plus courants sont

WIN. ✓ FAT32
✓ NTFS

LINUX ✓ EXT : *ext2/ext3/ext4*
✓ NFS
✓ ZFS
✓ BTRFS

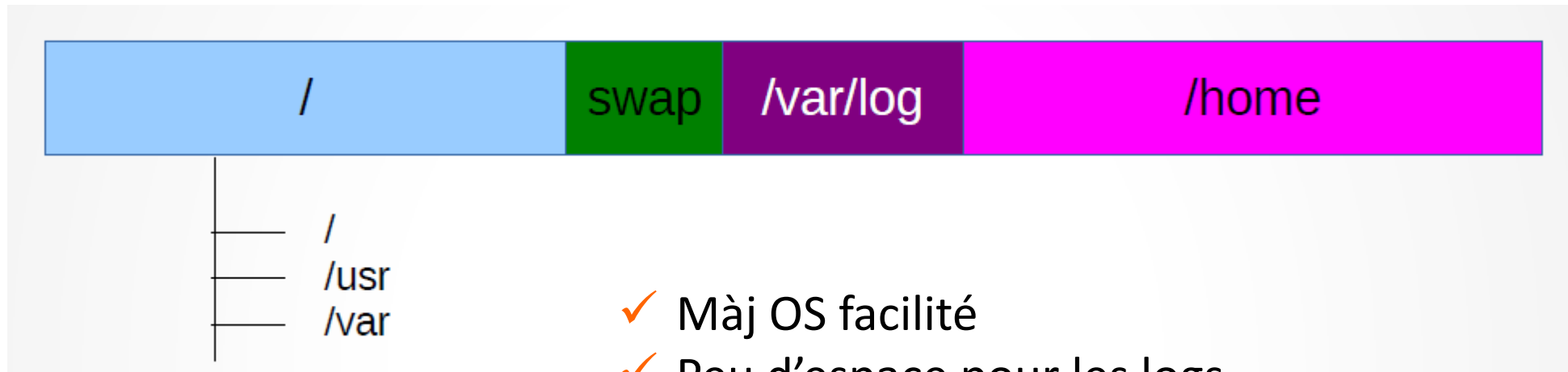
- Chaque système adresse des problématiques différentes :
 - ✓ **Journalisation / Snapshot / Réseau / Virtualisation**

Partitionnement



Stratégie de partitionnement

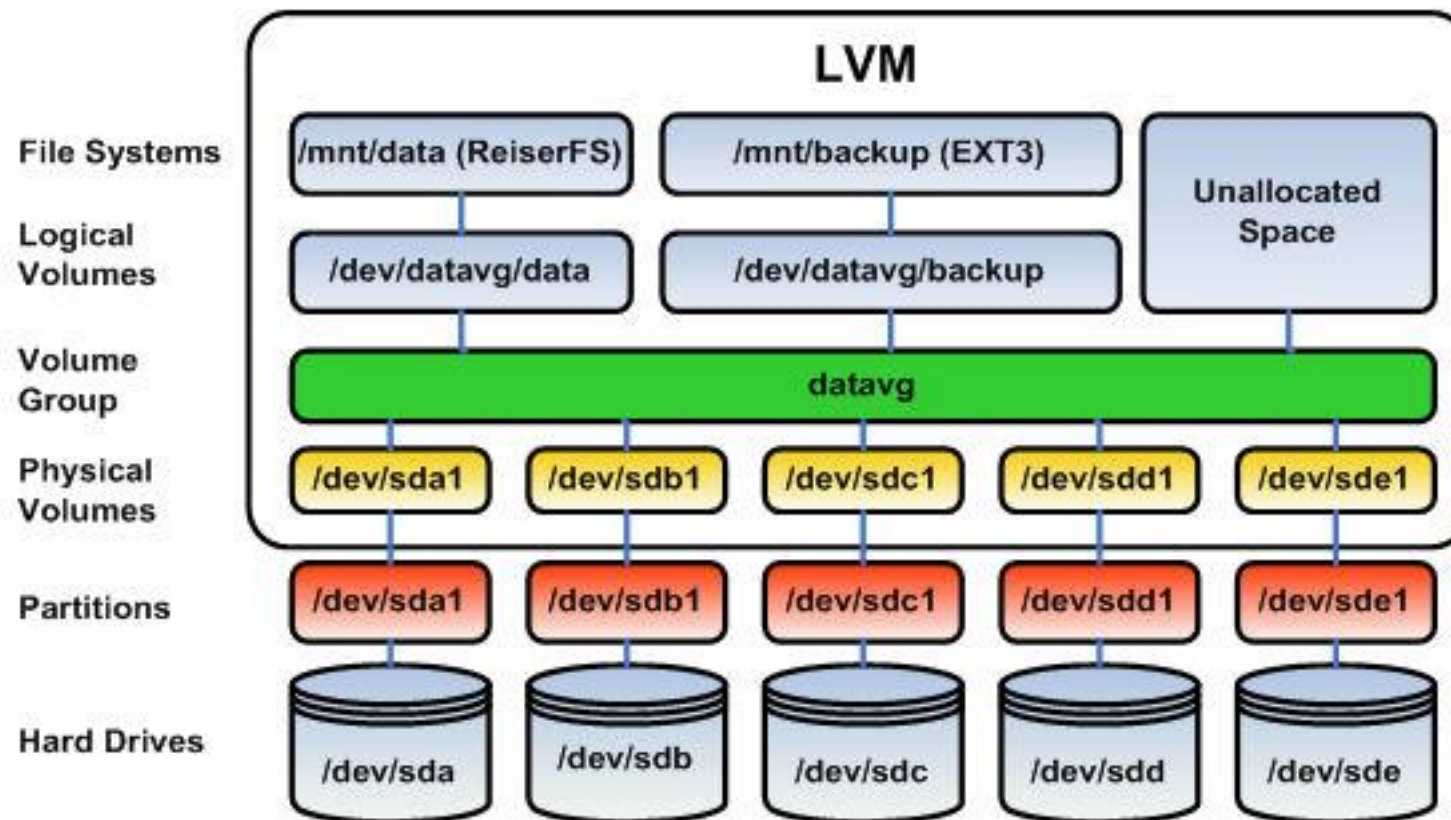
- Chaque stratégie doit répondre de la meilleure façon aux problématiques adressées :
 - ✓ Evolutivité ?
 - ✓ OS (màj, paquets) ?
 - ✓ SWAP ?



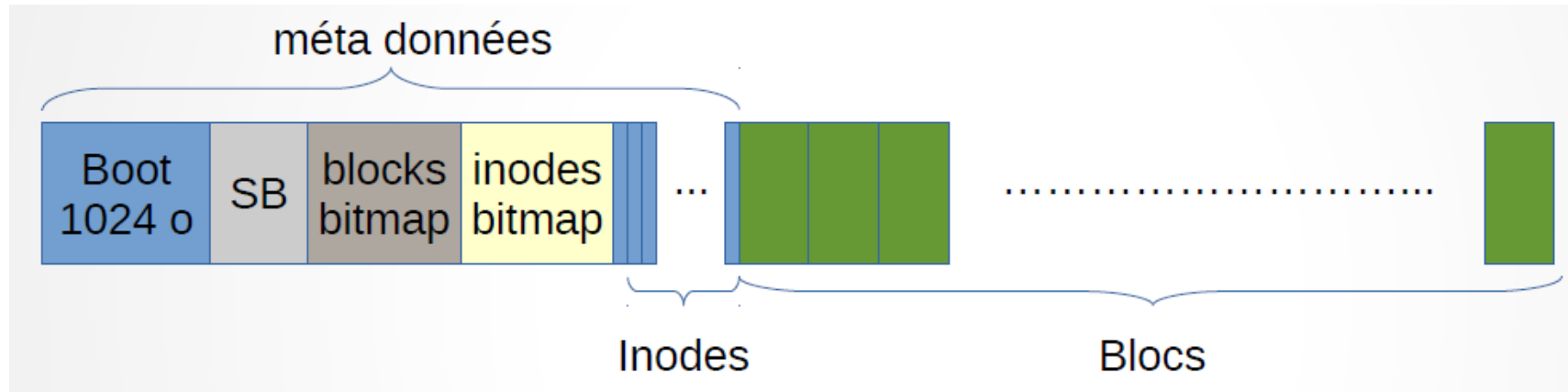
- ✓ Màj OS facilité
- ✓ Peu d'espace pour les logs
- ✓ SWAP idéal = taille de la RAM*

Stratégie de partitionnement : LVM ?

- Une alternative pour contrer une stratégie de partitionnement obsolète ou inadapté



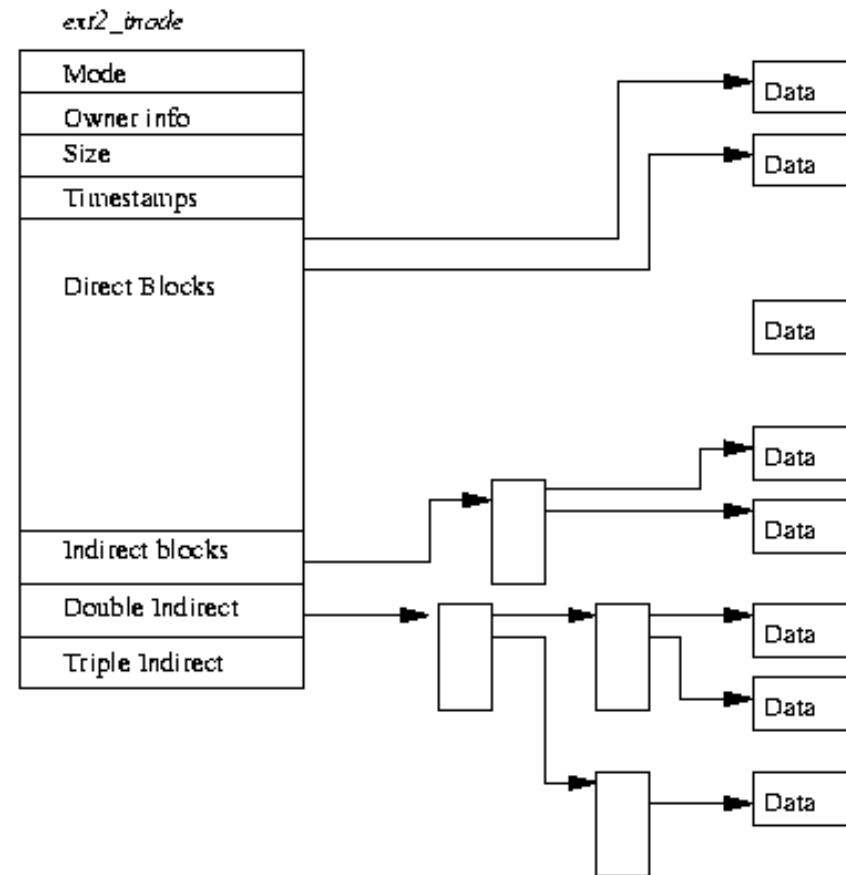
Exemple simplifié: le système ext2 - 2/2



- **Blocs:** données des fichiers
- **Inodes:** attributs des fichiers (1 inode = 1 fichier)
- **SB:** super bloc, contient: # inodes libres, offset 1er inode, idem pour les blocs, etc.
- **blocks bitmap:** carte des blocs libres et occupés
- **inode bitmap:** carte des inodes

Exemple simplifié: le système ext2

- Inode = métadata + @bloc data
- Mécanisme d'indirection :
 - ✓ Implémentation de liste chaînée
- Distribution EXT2:
 - 1@ = 4octets
 - 10blocs direct
 - 1bloc indirect simple
 - 1bloc indirect double
 - 1bloc indirect triple



SWAP

- Zone disque utilisée comme extension de la RAM, contient des pages des processus
 - ✓ N'est pas un système de fichier en soit
- 2 modes de fabrication :
 - ✓ Partition dédiée
 - ✓ Au sein d'un fichier
 - ✓ **mkswap** /path/to/my/swap_file
 - ✓ **swapon** /path/to/my/swap_file

Mise en œuvre

- 2 modes pour monter un FS au sein d'un OS Linux :
 - ✓ « Automatisé » utilisation d'un fichier spécial :

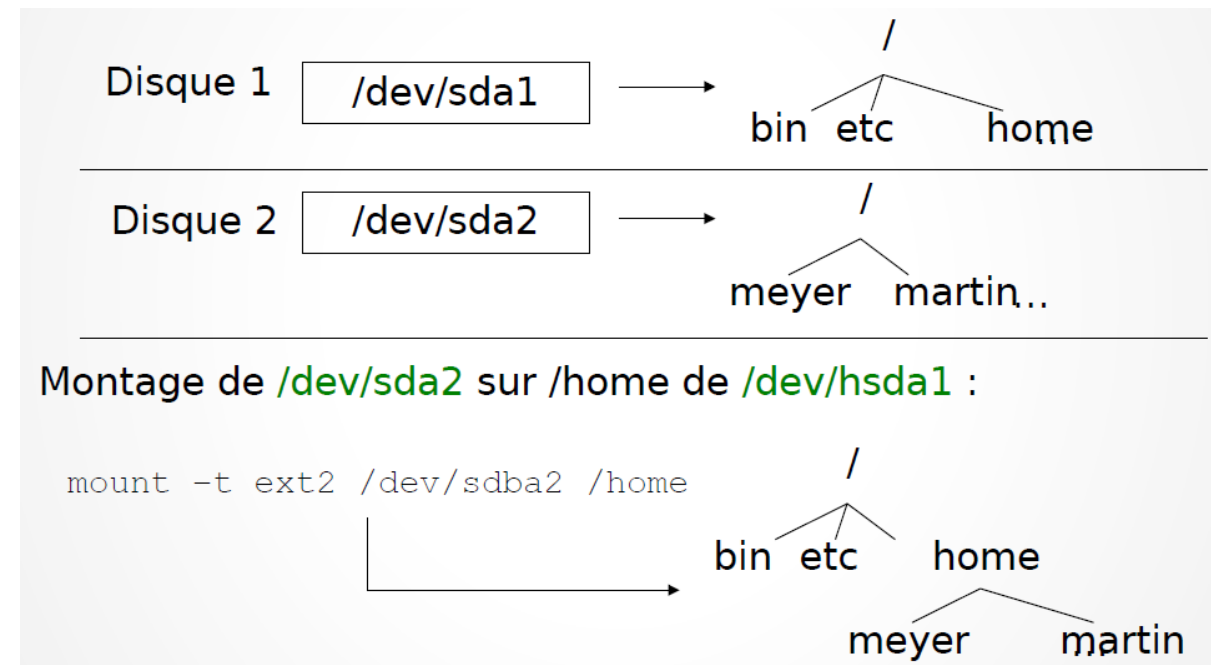
✓ /etc/fstab

emplacement					
/dev/sda1	/	ext3	defaults	0	1
/dev/sda2	/usr	ext3	defaults	0	1
/dev/scd0	/media/cdrom0	iso9660	ro,user,noauto	0	0
/dev/sdb1	/mnt/disque-usb1	auto	users	0	0

Diagram illustrating the /etc/fstab file structure with labels:

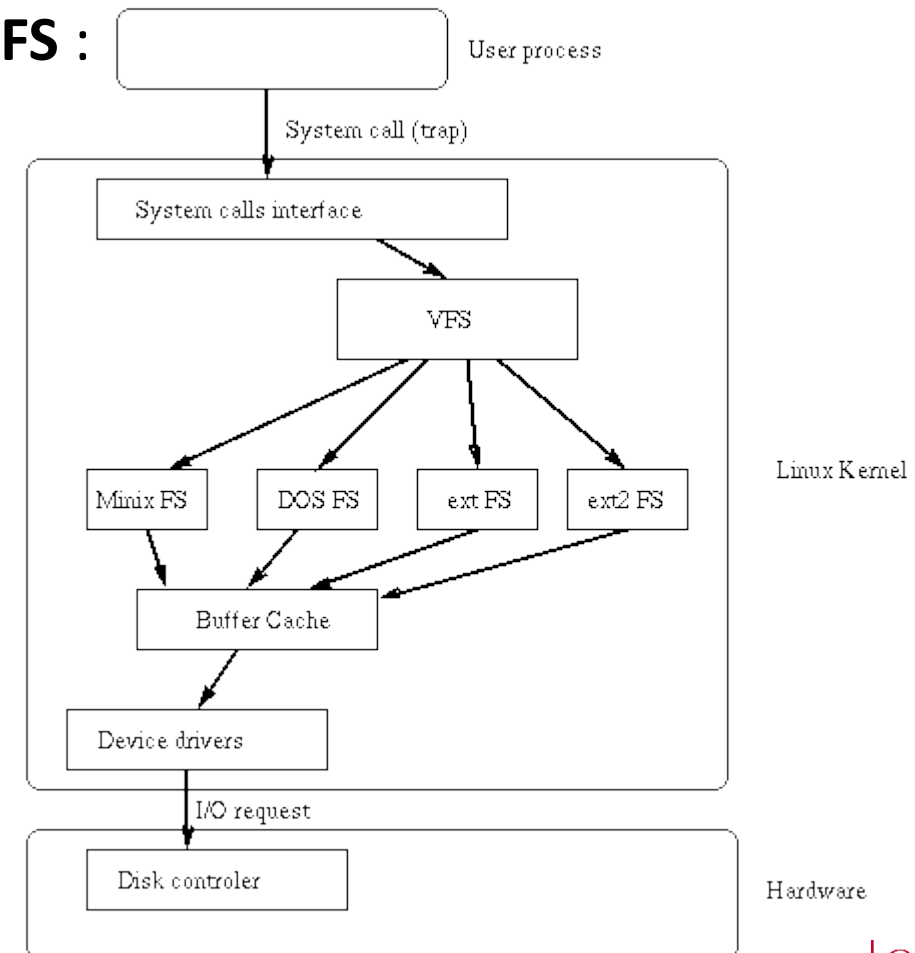
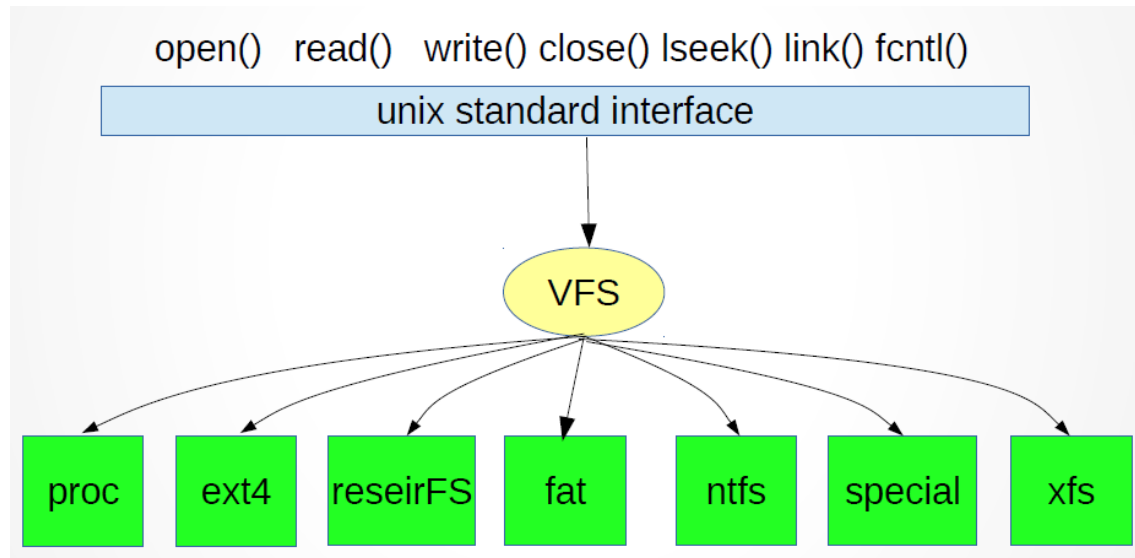
- emplacement**: Points to the device path (e.g., /dev/sda1).
- périphérique**: Points to the mount point (e.g., /).
- système de fichiers**: Points to the filesystem type (e.g., ext3).
- options**: Points to the mount options (e.g., defaults).

- ✓ « Manuel » utilisation de la commande **mount** (man mount)



Virtual File System - VFS

- API File System appelé **Virtual File System - VFS** :



Sommaire

- I. Linux & généralités
 - II. Système de gestion de fichier
 - III. Essentiel
 - IV. Conclusion
-

Définitions

- Un fichier se définit de la sorte :

ENSEMBLE

METADONNEES

- Une suite d'octets
- Un ensemble d'attributs dépendant du (*FS*)
 - ✓ Nom
 - ✓ Taille
 - ✓ Type
 - ✓ Propriétaire
 - ✓ Permissions
 - ✓ ...
- Présent et persitable sur un support physique selon une méthodologie définie (*SGF*)
- Le stockage d'un fichier est réalisé par bloc

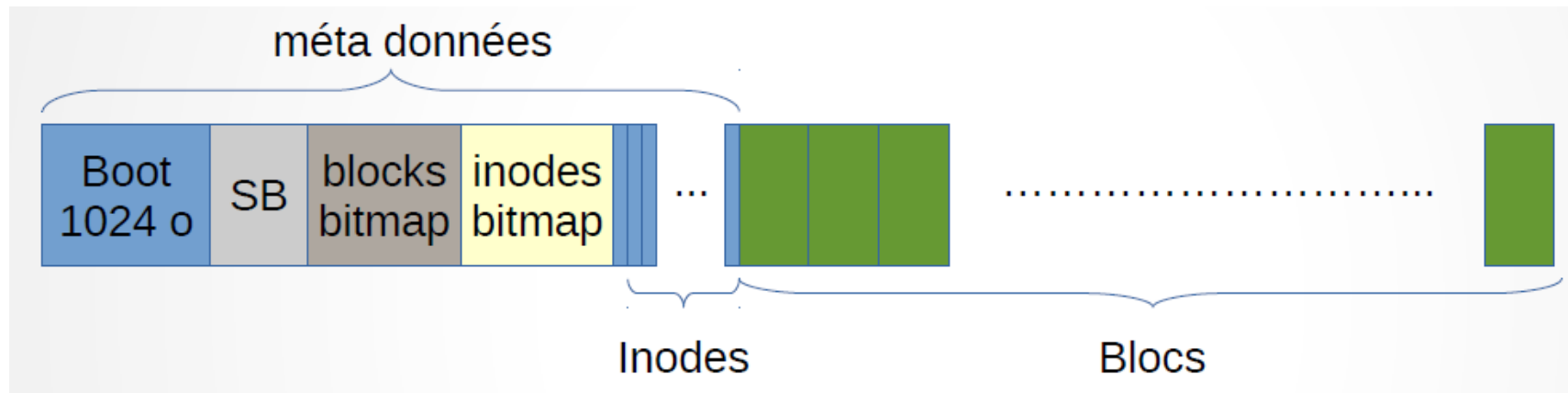
Définitions

- Le contenu d'un fichier est donc un sous-ensemble de blocs
 - <> Un sous-ensemble *précis* de blocs compose le contenu d'un fichier
- Le découpage en blocs sur un périphérique de stockage (disque dur, ram) implique:
 - ✓ La fragmentation interne du support : **non contiguïté des blocs de données**
 - ✓ La gestion de l'indexation des blocs
- L'ensemble des fichiers est structuré en arbre
 - ✓ L'élément racine de toute structuration étant : / (Linux) ou C:\ (Windows)

Définitions

- Un OS gère une forêt de fichier
 - ✓ La structuration de l'espace de stockage est appelé **File System (FS)**
 - ✓ Multiples espaces de stockage implique potentiellement plusieurs FS différents
 - L'OS doit être capable de gérer plusieurs types répartis sur plusieurs support
- Un système de gestion de fichier gère
 - ✓ La **structuration** de l'espace de stockage appelé **Formatage**
 - ✓ La **gestion** et l'**organisation** de l'espace de stockage : *Quid taille des blocs ?*
 - ✓ La **manipulation** des fichiers
 - ✓ La gestion des **métadonnées** des items
 - ✓ La **sécurité** des fichiers
- Chaque système adresse des problématiques différentes :
 - ✓ **Journalisation / Snapshot / Réseau / Virtualisation**

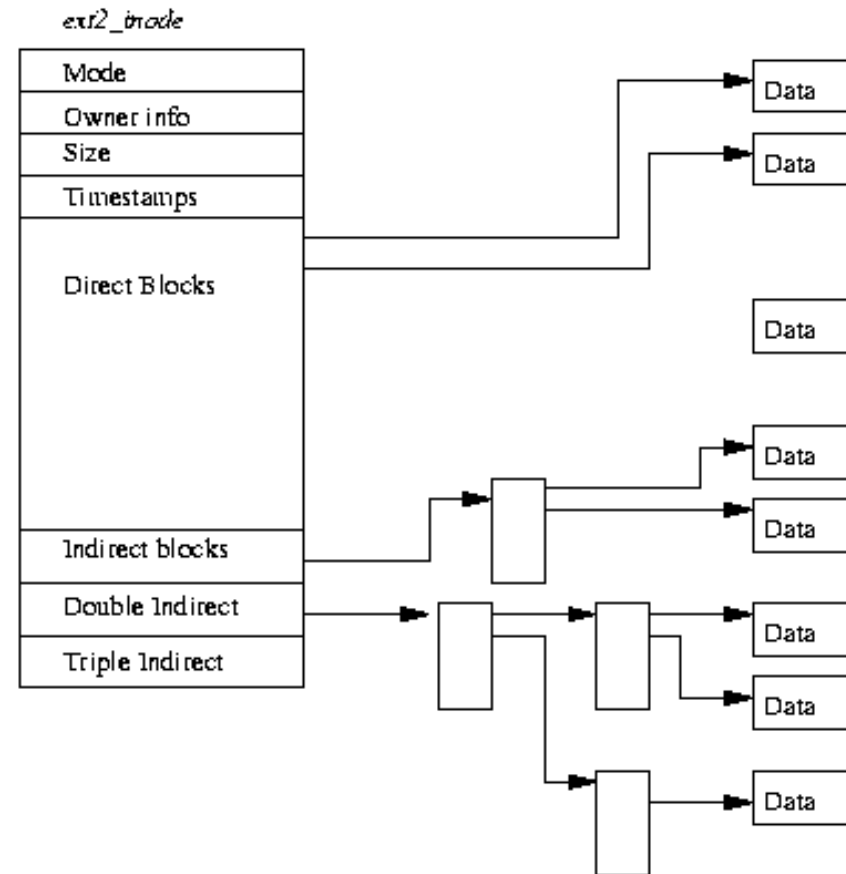
Exemple simplifié: le système ext2 - 2/2



- **Blocs:** données des fichiers
- **Inodes:** attributs des fichiers (1 inode = 1 fichier)
- **SB:** super bloc, contient: # inodes libres, offset 1er inode, idem pour les blocs, etc.
- **blocks bitmap:** carte des blocs libres et occupés
- **inode bitmap:** carte des inodes

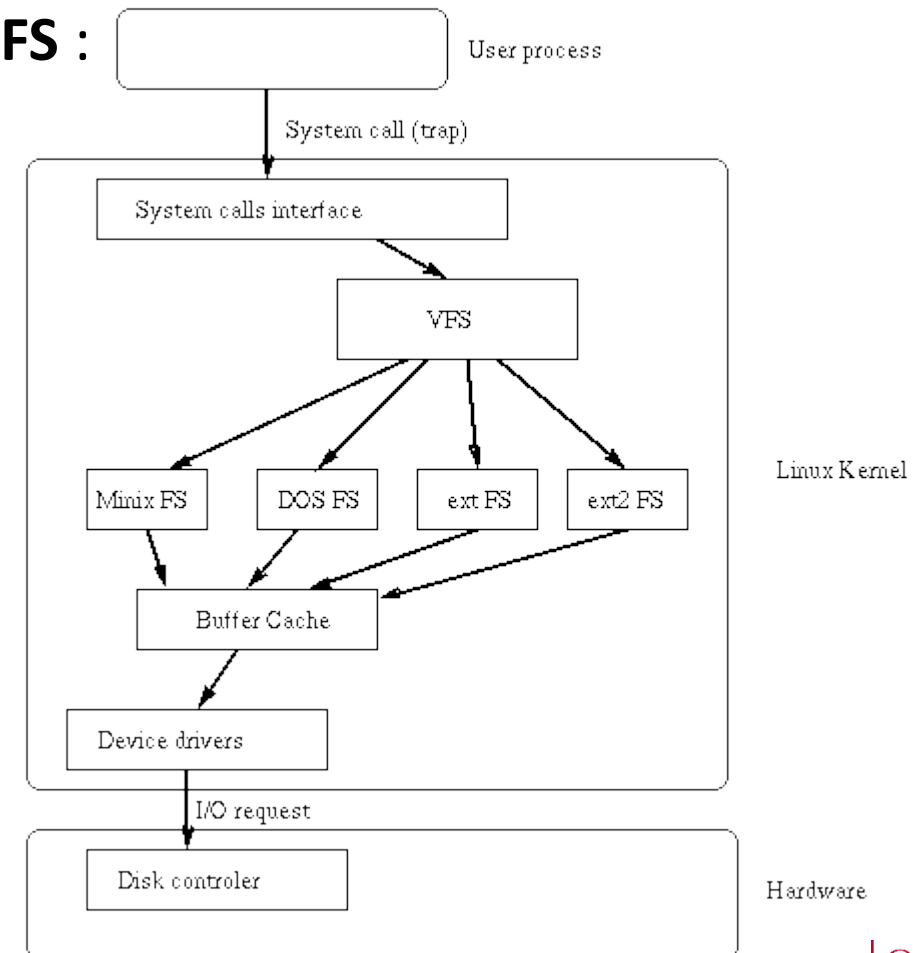
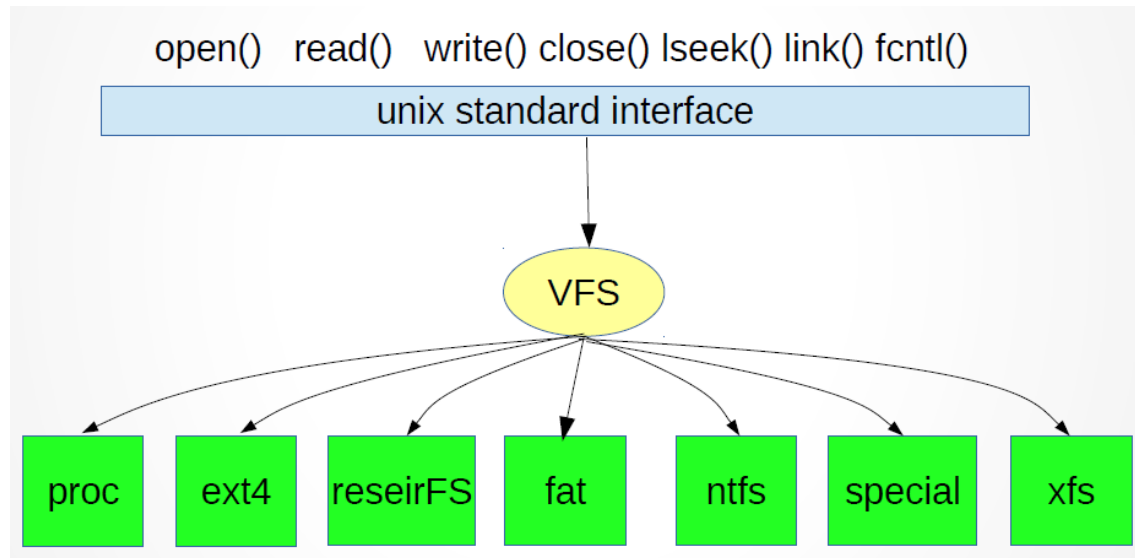
Exemple simplifié: le système ext2

- Inode = métadata + @bloc data
- Mécanisme d'indirection :
 - ✓ Implémentation de liste chaînée
- Distribution EXT2:
 - 1@ = 4octets
 - 10blocs direct
 - 1bloc indirect simple
 - 1bloc indirect double
 - 1bloc indirect triple



Virtual File System - VFS

- API File System appelé **Virtual File System - VFS** :





Conclusion



Annexes

Annexes

- Liens annexes :
 - ✓ *Filesystem Hierarchy Standard* : <http://www.pathname.com/fhs/>