

Administration Système sous Linux (Ubuntu Server)

Grégory Morel

2022-2023

CPE Lyon - 3IRC / 4ETI / 3ICS

Cours 5

Disques, partitionnement et systèmes de fichiers

Disques et partitions

Disque dur



HDD

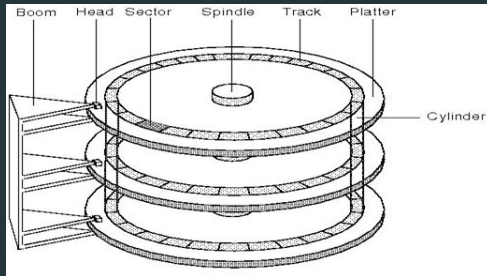


SSD

0. Illustration : <https://www.cleverfiles.com/howto/what-is-solid-state-drive.html>

Adressage les secteurs des disques durs

Adressage historique : **CHS** (Cylinder - Head - Sector) :



Max (théorique) : 1024 Cyl. / 256 Têtes / 63 secteurs soit $\approx 8 \text{ Go}^1$

⇒ Remplacé par ECHS puis **LBA** (Logical Block Addressing)

1. $1024 \times 256 \times 63 \times 5120$ (= taille usuelle d'un secteur de disque)

Partitions

Rappels

Le **partitionnement** consiste à diviser un support **physique** en espaces **logiques** (les **partitions**)

On peut ainsi découper un même disque physique en plusieurs disques virtuels, indépendants, et ayant chacun son propre système de fichiers.

Exemple

Création d'une partition pour le système et une partition pour les données, ce qui permet de réinstaller le système sans perdre de données

Partitions

Rappels

Le **partitionnement** consiste à diviser un support **physique** en espaces **logiques** (les **partitions**)

On peut ainsi découper un même disque physique en plusieurs disques virtuels, indépendants, et ayant chacun son propre système de fichiers.

Exemple

Création d'une partition pour le système et une partition pour les données, ce qui permet de réinstaller le système sans perdre de données

Table des partitions

Deux types de partitionnement d'un disque dur :

- **MBR** (*Master Boot Record*) ou **Intel** : introduit en 1983 (IBM PC DOS 2.0), "standard" sur les PC équipés d'un BIOS, jusqu'au début des années 2010.
Principales limitations :
 - ne gère pas les disques de plus de 2 To
 - limité à 4 partitions (dites *primaires*) par disque (mais il est possible de créer une *partition étendue* contenant des *partitions logiques*)
 - repose sur l'adressage CHS
- **GPT** (*GUID Partition Table*) : norme associée à **UEFI** (le remplaçant de BIOS)
 - chaque partition a un identifiant unique (Globally Unique ID)
 - jusqu'à 128 partitions
 - table des partitions répliquée à divers endroits du disque (récupération du système beaucoup plus simple)
 - utilise l'adressage LBA

Table des partitions

Deux types de partitionnement d'un disque dur :

- **MBR** (*Master Boot Record*) ou **Intel** : introduit en 1983 (IBM PC DOS 2.0), "standard" sur les PC équipés d'un BIOS, jusqu'au début des années 2010.
Principales limitations :
 - ne gère pas les disques de plus de 2 To
 - limité à 4 partitions (dites *primaires*) par disque (mais il est possible de créer une *partition étendue* contenant des *partitions logiques*)
 - repose sur l'adressage CHS
- **GPT** (*GUID Partition Table*) : norme associée à **UEFI** (le remplaçant de BIOS)
 - chaque partition a un identifiant unique (Globally Unique ID)
 - jusqu'à 128 partitions
 - table des partitions répliquée à divers endroits du disque (récupération du système beaucoup plus simple)
 - utilise l'adressage LBA

Lister les disques et les partitions

Rappel : sous Linux, un périphérique est un **fichier** dans le dossier **/dev**

- **disques durs** : *fichiers spéciaux* **/dev/sda**, **/dev/sdb...**¹

- **partitions** : nom du périphérique + index (**sda1**, **sda2...**)

```
$ ll /dev/sd*  
brw-rw---- 1 root disk 8, 0 mars  7 09:45 /dev/sda  
brw-rw---- 1 root disk 8, 1 mars  7 09:45 /dev/sda1  
brw-rw---- 1 root disk 8, 2 mars  7 09:45 /dev/sda2
```

💡 Le **b** qui précède les permissions indique un périphérique en **mode bloc** : les données sont lues et écrites sous forme de blocs²

1. Les anciens disques, qui utilisaient l'interface IDE, peuvent être nommés **hda**, **hdb...**

2. Contrairement aux périphériques en **mode caractère** (écran, clavier...) où les données sont lues / écrites caractère par caractère.

Lister les disques et les partitions

Rappel : sous Linux, un périphérique est un **fichier** dans le dossier **/dev**

- **disques durs** : *fichiers spéciaux* **/dev/sda**, **/dev/sdb...**¹

- **partitions** : nom du périphérique + index (**sda1**, **sda2...**)

```
$ ll /dev/sd*  
brw-rw---- 1 root disk 8, 0 mars  7 09:45 /dev/sda  
brw-rw---- 1 root disk 8, 1 mars  7 09:45 /dev/sda1  
brw-rw---- 1 root disk 8, 2 mars  7 09:45 /dev/sda2
```

💡 Le **b** qui précède les permissions indique un périphérique en **mode bloc** : les données sont lues et écrites sous forme de blocs²

1. Les anciens disques, qui utilisaient l'interface IDE, peuvent être nommés **hda**, **hdb...**

2. Contrairement aux périphériques en **mode caractère** (écran, clavier...) où les données sont lues / écrites caractère par caractère.

Lister les disques et les partitions

Rappel : sous Linux, un périphérique est un **fichier** dans le dossier **/dev**

- **disques durs** : *fichiers spéciaux* **/dev/sda**, **/dev/sdb...**¹

- **partitions** : nom du périphérique + index (**sda1**, **sda2...**)

```
$ ll /dev/sd*  
brw-rw---- 1 root disk 8, 0 mars  7 09:45 /dev/sda  
brw-rw---- 1 root disk 8, 1 mars  7 09:45 /dev/sda1  
brw-rw---- 1 root disk 8, 2 mars  7 09:45 /dev/sda2
```

💡 Le **b** qui précède les permissions indique un périphérique en **mode bloc** : les données sont lues et écrites sous forme de blocs²

1. Les anciens disques, qui utilisaient l'interface IDE, peuvent être nommés **hda**, **hdb...**

2. Contrairement aux périphériques en **mode caractère** (écran, clavier...) où les données sont lues / écrites caractère par caractère.

Lister les disques et les partitions

Rappel : sous Linux, un périphérique est un **fichier** dans le dossier **/dev**

- **disques durs** : *fichiers spéciaux* **/dev/sda**, **/dev/sdb...**¹

- **partitions** : nom du périphérique + index (**sda1**, **sda2...**)

```
$ ll /dev/sd*  
brw-rw---- 1 root disk 8, 0 mars  7 09:45 /dev/sda  
brw-rw---- 1 root disk 8, 1 mars  7 09:45 /dev/sda1  
brw-rw---- 1 root disk 8, 2 mars  7 09:45 /dev/sda2
```

💡 Le **b** qui précède les permissions indique un périphérique en **mode bloc** : les données sont lues et écrites sous forme de blocs²

1. Les anciens disques, qui utilisaient l'interface IDE, peuvent être nommés **hda**, **hdb...**

2. Contrairement aux périphériques en **mode caractère** (écran, clavier...) où les données sont lues / écrites caractère par caractère.

Lister les disques et les partitions

lsblk : liste les périphériques en mode bloc, et donc les disques et partitions :

```
$ lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
...
sda      8:0    0   10G  0 disk
|_sda1   8:1    0    1M  0 part
|_sda2   8:2    0   10G  0 part /
```

💡 **RM** : 1 si média amovible; **RO** : 1 si en lecture seule

Informations très basiques!

Lister les disques et les partitions

Pour avoir un peu plus d'infos : `fdisk -l` (indique le début et la fin de chaque partition, ainsi que le type de chaque partition)¹

```
$ fdisk -l /dev/sda
Disque /dev/sda : 10 GiB, 10737418240 octets, 20971520 secteurs
Unités : secteurs de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
Type d'étiquette de disque : gpt
Identifiant de disque : 9B945FAE-E61B-BAC7-8CC3-54287DFA8B6E

Périphérique Début          Fin Secteurs Taille Type
/dev/sda1      2048          4095      2048      1M BIOS boot
/dev/sda2      4096 20969471 20965376    10G Linux filesystem
```

1. Citons également les commandes `parted`, `gdisk`, `sfdisk`...

UUID

Un disque ou une partition peut aussi être désigné par un **UUID**¹. On l'obtient :

- soit avec **blkid** :

```
$ sudo blkid  
/dev/sda2: UUID="6d65e8fd-e1fa-423d-9654-a1300e5ef89d" TYPE="ext4"
```

- soit en examinant le contenu des dossiers **/dev/disk/by-***** :

```
$ ls -l /dev/disk/by-uuid/  
total 0  
lrwxrwxrwx 1 root root 10 mars 7 09:45 6d65e8fd-e1fa-423d-9654-  
a1300e5ef89d -> ../../sda2
```

1. Universal Unique Identifier

UUID

Un disque ou une partition peut aussi être désigné par un **UUID**¹. On l'obtient :

- soit avec **blkid** :

```
$ sudo blkid  
/dev/sda2: UUID="6d65e8fd-e1fa-423d-9654-a1300e5ef89d" TYPE="ext4"
```

- soit en examinant le contenu des dossiers **/dev/disk/by-***** :

```
$ ls -l /dev/disk/by-uuid/  
total 0  
lrwxrwxrwx 1 root root 10 mars 7 09:45 6d65e8fd-e1fa-423d-9654-  
a1300e5ef89d -> ../../sda2
```

1. Universal Unique Identifier

Editer les partitions

fdisk permet aussi d'**éditer** la table des partitions ⇒ A utiliser avec précaution !

Exemple : **créer une partition**

```
Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1):
First sector (2048-1953525167, default 2048):
Last sector, +sectors or +sizeK,M,G,T,P (2048-1953525167, default
1953525167):

Created a new partition 1 of type 'Linux' and of size 931,5 GiB.
```

Partition particulière : le **swap** ou **espace d'échange**

Le **swap** :

- permet de stocker temporairement une partie de la RAM quand celle-ci est saturée;
- également utilisé pour l'**hibernation** (veille prolongée) : dans ce cas, l'intégralité de la RAM est transférée sur disque.

💡 Depuis Ubuntu 17.04, la partition d'échange est remplacée par un simple fichier d'échange ¹

Attention!

Les disques SSD ont un nombre de cycles d'écriture limité, il est donc déconseillé d'y logger une partition swap ou un fichier d'échange!

1. Existe aussi sous Windows : ce sont les fichiers **pagefile.sys** et **hiberfil.sys**

Partition particulière : le **swap** ou **espace d'échange**

Le **swap** :

- permet de stocker temporairement une partie de la RAM quand celle-ci est saturée;
- également utilisé pour l'**hibernation** (veille prolongée) : dans ce cas, l'intégralité de la RAM est transférée sur disque.

💡 Depuis Ubuntu 17.04, la partition d'échange est remplacée par un simple fichier d'échange¹

Attention!

Les disques SSD ont un nombre de cycles d'écriture limité, il est donc déconseillé d'y logger une partition swap ou un fichier d'échange!

1. Existe aussi sous Windows : ce sont les fichiers **pagefile.sys** et **hiberfil.sys**

Partition particulière : le **swap** ou **espace d'échange**

Le **swap** :

- permet de stocker temporairement une partie de la RAM quand celle-ci est saturée;
- également utilisé pour l'**hibernation** (veille prolongée) : dans ce cas, l'intégralité de la RAM est transférée sur disque.

💡 Depuis Ubuntu 17.04, la partition d'échange est remplacée par un simple fichier d'échange¹

Attention!

Les disques SSD ont un nombre de cycles d'écriture limité, il est donc déconseillé d'y logger une partition swap ou un fichier d'échange!

1. Existe aussi sous Windows : ce sont les fichiers **pagefile.sys** et **hiberfil.sys**

Montage / démontage d'une partition

Monter / démonter un périphérique ou une partition

Définition

Monter un périphérique (ou une partition) = relier son système de fichiers à celui de la machine, en un **point de montage**

💡 Inversement, *démonter un périphérique*, c'est retirer ce point de montage (≈ “retirer en toute sécurité” sous Windows)

Pour monter / démonter un SF, on utilise les commandes **mount**/**umount** :

```
$ sudo mount /dev/cdrom /media/cdrom  
$ sudo umount /media/cdrom
```

💡 Le système de fichiers est normalement détecté automatiquement

💡 Si le démontage ne fonctionne pas, **fuser** permet de savoir quel processus utilise la partition

Monter / démonter un périphérique ou une partition

Définition

Monter un périphérique (ou une partition) = relier son système de fichiers à celui de la machine, en un **point de montage**

💡 Inversement, *démonter un périphérique*, c'est retirer ce point de montage (≈ “retirer en toute sécurité” sous Windows)

Pour monter / démonter un SF, on utilise les commandes **mount**/**umount** :

```
$ sudo mount /dev/cdrom /media/cdrom
$ sudo umount /media/cdrom
```

💡 Le système de fichiers est normalement détecté automatiquement

💡 Si le démontage ne fonctionne pas, **fuser** permet de savoir quel processus utilise la partition

Le fichier de configuration **/etc/fstab** (filesystems table)

💡 Contient les informations nécessaires pour automatiser le montage des partitions (au démarrage ou au branchement du périphérique)

```
$ cat /etc/fstab
UUID=6d65e8fd-e1fa-423d-9654-a1300e5ef89d /      ext4 defaults 0 0
/swap.img                             none  swap sw      0 0
```

6 colonnes :

1. UUID
2. Point de montage
3. Système de fichiers (ou **auto** pour les médias amovibles)
4. Options de montage (cf. **man mount**)
5. Active / désactive la sauvegarde (commande *dump*) (généralement à 0)
6. Vérification de la partition au démarrage (0 : pas de vérification)

Seul **root** peut monter un périphérique ou une partition qui n'apparaît pas dans **fstab**

Le fichier de configuration `/etc/fstab`

Avantage des UUID : un disque `sda` peut très bien s'appeler `sdb` après insertion d'un nouveau disque et redémarrage !

L'UUID est calculé à partir de données de la machine, de la date de formatage, etc.

⇒ quand on formate un disque, son UUID change !

⇒ il faut penser à modifier le fichier `/etc/fstab` !

De plus, le dossier `/dev/disk/by-uuid` n'est mis à jour qu'au démarrage ; on peut le mettre à jour manuellement avec `sudo udevadm trigger`

En revanche, l'UUID reste le même quand on clone une partition ; on peut générer un nouvel UUID avec `uuidgen`

Le fichier de configuration `/etc/fstab`

Forcer la prise en compte des modification de la *fstab* : `mount -a`

Le fichier `/etc/mtab` contient la liste des montages effectués (avec `/etc/fstab` ou avec `mount`).

Systèmes de fichiers

Qu'est-ce qu'un système de fichiers ?

Définition

Un **système de fichiers** (SF) décrit la manière dont les données sont organisées à l'intérieur d'une partition.

Il fait l'**interface** entre l'utilisateur (applications) et le pilote d'E/S du matériel.

💡 Sans SF, ce serait à l'utilisateur d'indiquer une adresse sur le disque et le nombre d'octets à lire à chaque ouverture de fichier !

Avec un SF, les données sont découpées en **fichiers**, organisés hiérarchiquement, et c'est l'OS qui se charge de retrouver les données associées.

💡 On peut voir un SF comme l'**index** de la partition : il donne l'adresse de chaque fichier

Formater une partition, c'est y créer un système de fichiers

Qu'est-ce qu'un système de fichiers ?

Définition

Un **système de fichiers** (SF) décrit la manière dont les données sont organisées à l'intérieur d'une partition.

Il fait l'**interface** entre l'utilisateur (applications) et le pilote d'E/S du matériel.

💡 Sans SF, ce serait à l'utilisateur d'indiquer une adresse sur le disque et le nombre d'octets à lire à chaque ouverture de fichier !

Avec un SF, les données sont découpées en **fichiers**, organisés hiérarchiquement, et c'est l'OS qui se charge de retrouver les données associées.

💡 On peut voir un SF comme l'**index** de la partition : il donne l'adresse de chaque fichier

Formater une partition, c'est y créer un système de fichiers

Qu'est-ce qu'un système de fichiers ?

Définition

Un **système de fichiers** (SF) décrit la manière dont les données sont organisées à l'intérieur d'une partition.

Il fait l'**interface** entre l'utilisateur (applications) et le pilote d'E/S du matériel.

💡 Sans SF, ce serait à l'utilisateur d'indiquer une adresse sur le disque et le nombre d'octets à lire à chaque ouverture de fichier !

Avec un SF, les données sont découpées en **fichiers**, organisés hiérarchiquement, et c'est l'OS qui se charge de retrouver les données associées.

💡 On peut voir un SF comme l'**index** de la partition : il donne l'adresse de chaque fichier

Formater une partition, c'est y créer un système de fichiers

Fonctionnalités d'un système de fichiers

Il existe de nombreux systèmes de fichiers (**NTFS** sous Windows, **extX** sous Linux, souvent **FAT** sur les clés USB...), qui diffèrent sur les fonctionnalités qu'ils offrent :

- gestion de l'espace libre et de la fragmentation,
- journalisation (traces des opérations d'écritures non terminées)
- gestion des droits
- maintien de l'intégrité (que se passe-t-il si on arrache une clé USB avant la fin du déplacement d'un fichier?)

Sous Linux, la commande **df -T** indique le système de fichiers utilisé sur chaque partition

Fonctionnalités d'un système de fichiers

Il existe de nombreux systèmes de fichiers (**NTFS** sous Windows, **extX** sous Linux, souvent **FAT** sur les clés USB...), qui diffèrent sur les fonctionnalités qu'ils offrent :

- gestion de l'espace libre et de la fragmentation,
- journalisation (traces des opérations d'écritures non terminées)
- gestion des droits
- maintien de l'intégrité (que se passe-t-il si on arrache une clé USB avant la fin du déplacement d'un fichier?)

Sous Linux, la commande **df -T** indique le système de fichiers utilisé sur chaque partition

Fonctionnalités d'un système de fichiers

Il existe de nombreux systèmes de fichiers (**NTFS** sous Windows, **extX** sous Linux, souvent **FAT** sur les clés USB...), qui diffèrent sur les fonctionnalités qu'ils offrent :

- gestion de l'espace libre et de la fragmentation,
- journalisation (traces des opérations d'écritures non terminées)
- gestion des droits
- maintien de l'intégrité (que se passe-t-il si on arrache une clé USB avant la fin du déplacement d'un fichier?)

Sous Linux, la commande **df -T** indique le système de fichiers utilisé sur chaque partition

Fonctionnalités d'un système de fichiers

Il existe de nombreux systèmes de fichiers (**NTFS** sous Windows, **extX** sous Linux, souvent **FAT** sur les clés USB...), qui diffèrent sur les fonctionnalités qu'ils offrent :

- gestion de l'espace libre et de la fragmentation,
- journalisation (traces des opérations d'écritures non terminées)
- gestion des droits
- maintien de l'intégrité (que se passe-t-il si on arrache une clé USB avant la fin du déplacement d'un fichier?)

Sous Linux, la commande **df -T** indique le système de fichiers utilisé sur chaque partition

Fonctionnalités d'un système de fichiers

Il existe de nombreux systèmes de fichiers (**NTFS** sous Windows, **extX** sous Linux, souvent **FAT** sur les clés USB...), qui diffèrent sur les fonctionnalités qu'ils offrent :

- gestion de l'espace libre et de la fragmentation,
- journalisation (traces des opérations d'écritures non terminées)
- gestion des droits
- maintien de l'intégrité (que se passe-t-il si on arrache une clé USB avant la fin du déplacement d'un fichier?)

Sous Linux, la commande **df -T** indique le système de fichiers utilisé sur chaque partition

Quelques systèmes de fichiers Linux

- **ext2** (1993) : SF historique de Linux, non journalisé
- **ext3** (1999) : successeur de ext2, apporte la journalisation
- **ext4** (2008) : SF par défaut sur de nombreuses distributions; apporte des fonctionnalités plus avancées contre la fragmentation (*extents*); vu comme une étape de transition vers BTRFS
- **BTRFS**¹ (2014) : développé par Oracle, Intel, Facebook...; encore expérimental, il permet d'ajouter ou supprimer des volumes "à chaud"; il permet aussi la création de *snapshots* autorisant à faire des tests et revenir à l'état antérieur (comme avec LVM)
- d'autres comme **ReiserFS**², **XFS**...

1. Prononcé "Better FS" ou "Butter FS", ou encore parfois "B-tree" FS

2. Un peu moins maintenu depuis que son créateur purge une peine de prison pour le meurtre de sa femme...

Quelques systèmes de fichiers Linux

- **ext2** (1993) : SF historique de Linux, non journalisé
- **ext3** (1999) : successeur de ext2, apporte la journalisation
- **ext4** (2008) : SF par défaut sur de nombreuses distributions; apporte des fonctionnalités plus avancées contre la fragmentation (*extents*); vu comme une étape de transition vers BTRFS
- **BTRFS**¹ (2014) : développé par Oracle, Intel, Facebook...; encore expérimental, il permet d'ajouter ou supprimer des volumes "à chaud"; il permet aussi la création de *snapshots* autorisant à faire des tests et revenir à l'état antérieur (comme avec LVM)
- d'autres comme **ReiserFS**², **XFS**...

1. Prononcé "Better FS" ou "Butter FS", ou encore parfois "B-tree" FS

2. Un peu moins maintenu depuis que son créateur purge une peine de prison pour le meurtre de sa femme...

Quelques systèmes de fichiers Linux

- **ext2** (1993) : SF historique de Linux, non journalisé
- **ext3** (1999) : successeur de ext2, apporte la journalisation
- **ext4** (2008) : SF par défaut sur de nombreuses distributions; apporte des fonctionnalités plus avancées contre la fragmentation (*extents*); vu comme une étape de transition vers BTRFS
- **BTRFS**¹ (2014) : développé par Oracle, Intel, Facebook...; encore expérimental, il permet d'ajouter ou supprimer des volumes "à chaud"; il permet aussi la création de *snapshots* autorisant à faire des tests et revenir à l'état antérieur (comme avec LVM)
- d'autres comme **ReiserFS**², **XFS**...

1. Prononcé "Better FS" ou "Butter FS", ou encore parfois "B-tree" FS

2. Un peu moins maintenu depuis que son créateur purge une peine de prison pour le meurtre de sa femme...

Quelques systèmes de fichiers Linux

- **ext2** (1993) : SF historique de Linux, non journalisé
- **ext3** (1999) : successeur de ext2, apporte la journalisation
- **ext4** (2008) : SF par défaut sur de nombreuses distributions; apporte des fonctionnalités plus avancées contre la fragmentation (*extents*); vu comme une étape de transition vers BTRFS
- **BTRFS**¹ (2014) : développé par Oracle, Intel, Facebook...; encore expérimental, il permet d'ajouter ou supprimer des volumes “à chaud”; il permet aussi la création de *snapshots* autorisant à faire des tests et revenir à l'état antérieur (comme avec LVM)
- d'autres comme **ReiserFS**², **XFS**...

1. Prononcé “Better FS” ou “Butter FS”, ou encore parfois “B-tree” FS

2. Un peu moins maintenu depuis que son créateur purge une peine de prison pour le meurtre de sa femme...

Quelques systèmes de fichiers Linux

- **ext2** (1993) : SF historique de Linux, non journalisé
- **ext3** (1999) : successeur de ext2, apporte la journalisation
- **ext4** (2008) : SF par défaut sur de nombreuses distributions; apporte des fonctionnalités plus avancées contre la fragmentation (*extents*); vu comme une étape de transition vers BTRFS
- **BTRFS**¹ (2014) : développé par Oracle, Intel, Facebook...; encore expérimental, il permet d'ajouter ou supprimer des volumes "à chaud"; il permet aussi la création de *snapshots* autorisant à faire des tests et revenir à l'état antérieur (comme avec LVM)
- d'autres comme **ReiserFS**², **XFS**...

1. Prononcé "Better FS" ou "Butter FS", ou encore parfois "B-tree" FS

2. Un peu moins maintenu depuis que son créateur purge une peine de prison pour le meurtre de sa femme...

Système de fichiers virtuel proc

proc : système de fichiers virtuel / pseudo système de fichiers, monté généralement sur **/proc**

⇒ fournit une interface avec les structures de données du noyau. La plupart des fichiers sont en lecture seule (ex. : **cpuinfo**), mais quelques uns permettent la modification à chaud du noyau¹.

💡 On y trouve également un répertoire par processus actif, portant le numéro de ce processus, et contenant notamment les fichiers :

- **cmdline** : ligne de commande qui a créé le processus
- **status** : état du processus (en attente, en exécution...)
- **exe** : lien vers l'exécutable utilisé par le processus

1. Ex. : **/proc/sys/net/ipv4/ip_forward** pour activer l'IP Forwarding

Système de fichiers virtuel proc

proc : système de fichiers virtuel / pseudo système de fichiers, monté généralement sur **/proc**

⇒ fournit une interface avec les structures de données du noyau. La plupart des fichiers sont en lecture seule (ex. : **cpuinfo**), mais quelques uns permettent la modification à chaud du noyau¹.

💡 On y trouve également un répertoire par processus actif, portant le numéro de ce processus, et contenant notamment les fichiers :

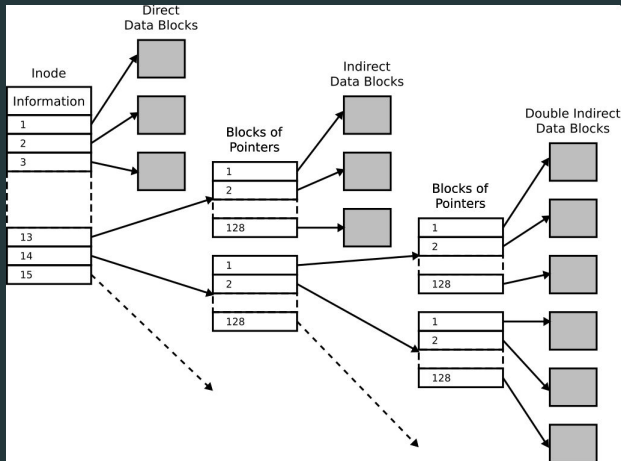
- **cmdline** : ligne de commande qui a créé le processus
- **status** : état du processus (en attente, en exécution...)
- **exe** : lien vers l'exécutable utilisé par le processus

1. Ex. : **/proc/sys/net/ipv4/ip_forward** pour activer l'IP Forwarding

Inodes

Inodes

Dans un système Unix, un **inode** (*index node*) est une structure de données qui contient les **métadonnées** associées à un fichier / dossier (permissions, propriétaire, groupe, date de dernier accès...), ainsi que des **pointeurs** sur les données :



Structure d'un
inode dans ext3

Inodes

Rem. : le **nom** d'un fichier n'est pas stocké dans l'inode (on verra plus loin pourquoi)

Chaque inode est identifié par un nombre (*i-number*) :

```
$ ls -li
264527 Bureau
264579 Documents
276656 examples.desktop
264581 Images
264577 Modèles
264580 Musique
264578 Public
264576 Téléchargements
264582 Vidéos
```

Le système de fichiers maintient une **table des inodes** qui lie n° d'inodes et inodes de tous les fichiers présents sur le disque

Important

Sous Linux, on peut associer **plusieurs noms** à un même inode; autrement dit, un fichier peut avoir **plusieurs noms**!

Pour rajouter un nom à un fichier, on crée un **lien physique** (*hard link*), avec la commande **ln** :

```
$ touch toto ; ls -li
270127 toto
$ ln toto titi ; ls -li
270127 titi 270127 toto
```

💡 La création d'un lien physique incrémente un **compteur de noms** contenu dans les métadonnées dans l'inode; la suppression d'un fichier ne fait que décrémenter ce compteur. Le contenu du fichier devient inaccessible lorsque le compteur passe à 0.

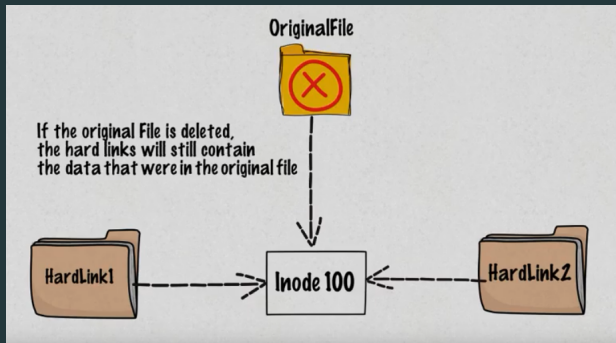
On peut visualiser le nombre de noms d'un fichier avec la commande **ls -l** :

```
$ ls -l toto  
-rw-r--r-- 2 greg greg 0 avr. 12 00:05 toto
```

💡 Pour un dossier, le nombre de noms est égal au nombre de sous-dossiers directs + 2 (lui-même et son dossier parent)

Inodes

Les liens physiques partagent le même contenu. Par conséquent, même si on supprime le « fichier » (en réalité, le *nom*) original, les données sont toujours accessibles à travers le lien :



(Source : chaîne Youtube *Linux Recipe*)

On peut créer des liens physiques sous Windows avec la commande `fsutil hardlink`

💡 Les noms d'un fichier ne sont pas stockés dans les métadonnées contenues dans l'inode

A retenir

Un dossier (= répertoire) est un fichier contenant une table de correspondance *noms* \leftrightarrow *n° d'inodes*

💡 C'est pourquoi un dossier ne peut pas contenir deux fichiers portant le même nom!

❓ Si un fichier est associé à son inode *via* son dossier parent, comment le dossier / est-il associé à son inode?

⇒ L'inode de / est fixé et vaut toujours 2

Inodes

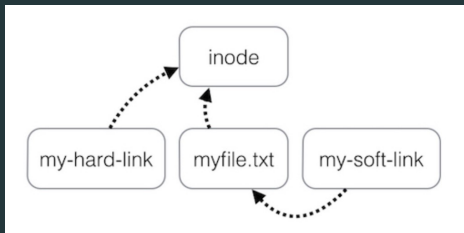
On peut également créer un lien **symbolique** (*symlink* ou *soft link*) sur un fichier

Important

Un lien symbolique se comporte comme un « raccourci » sous Windows : si on supprime le fichier original, le lien ne permet plus d'accéder aux données

```
$ touch toto ; ln -s toto tutu ; ls -li  
270127 titi 270127 toto 270465 tutu
```

💡 Un lien symbolique pointe sur un **nom de fichier** (il possède son propre inode, dont les données contiennent le nom du fichier pointé)



Inodes

Différences entre liens physiques et liens symboliques :

- impossible de créer un lien physique vers un dossier¹
- impossible de créer un lien physique vers un fichier sur un autre système de fichiers ou une autre partition (les inodes sont propres à une partition)

Le nombre d'inodes est limité et déterminé lors de la création du système de fichiers sur la partition. Lorsque les inodes sont épuisés, il est impossible de créer un nouveau fichier même s'il reste de l'espace sur le disque². On peut connaître le nombre d'inodes restants avec `df -i` :

```
$ df -i
```

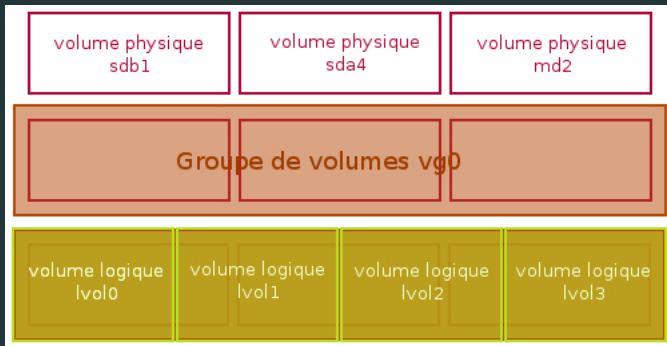
Filesystem	Inodes	IUsed	IFree	IUse%	Mounted on
/dev/sda2	655360	115130	540230	18%	/

-
2. Sinon on pourrait créer des boucles ou faire d'un dossier son propre parent
2. Peut arriver quand on a beaucoup de petits fichiers, par exemple sur un serveur mail

LVM

LVM : Logical Volume Manager

Consiste à regrouper des disques ou volumes (= partitions) *physiques* en un seul grand volume dans lequel on peut découper des volumes *logiques*, et les agrandir, les réduire, etc. à volonté.



Chaque volume logique correspond ensuite à une partition.

Intérêt

- partitionnement simplifié
- possibilité de réaliser des clichés (*snapshots*), ce qui permet de revenir à un état antérieur

LVM vs. RAID

- pas de redondance de données
- pas d'entrelacement de données
- si un des volumes physiques devient HS, l'ensemble des volumes logiques qui utilisent ce volume physique sont perdus

💡 Solution : utiliser LVM sur des disques RAID par exemple

💡 LVM est à privilégier dans le cadre d'un serveur d'entreprise, ou quand on pense rajouter des disques physiques plus tard.