

# TP 5 - Systèmes de fichiers, partitions et disques

## Inode

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 :

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

**Les liens physiques** partagent le même contenu. La création d'un lien physique incrémente un compteur de noms contenu dans les métadonnées dans l'inode

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

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

## Système de fichiers

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.

`df -T`

## /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 noyau.

On y trouve également un répertoire par processus actif, portant le numéro de ce processus,

## Exercice 1. Disques et partitions

1.

Créer un second disque dur

2.

Pour vérifier que le disque est bien détecté par le système

```
Vérifiez que ce nouveau disque dur est bien détecté par le système
Plusieurs solutions possibles : on constate la présence d'un fichier sdb dans /dev ; ou encore
lsblk et fdisk -l mentionnent un disque de 5 Go nommé sdb
```

3.

Partitionner le disque avec `fdisk` 2Go Linux et 3Go NTFS

```
fdisk /dev/sdb puis choisir n, p, +2G pour la première partition ; répéter pour la 2ème,
puis changer son type avec t et choisir le type 7. Prévisualiser le résultat avec p puis
écrire les modifications avec w. Vérifier ensuite avec fdisk -l que les partitions ont bien
été créées
```

4.

A l'aide de `mkfs` on formate les deux partitions

```
on utilise donc mkfs.ext4 /dev/sdb1 pour la première partition, et
mkfs.ntfs /dev/sdb2 pour la seconde
```

5.

Pour que la commande `df -T` fonctionne il faut monter le disque

6.

Pour que les deux partitions créées soient montées automatiquement au démarrage, respectivement dans les points de montage /data et /win

```
Créer les deux dossiers à la racine, puis éditer /etc/fstab :  
/dev/sdb1 /data ext4 defaults 0 0  
/dev/sdb2 /win ntfs defaults 0 0  
Ensuite, valider par mount -a puis mount
```

7

*mont* valide la configuration

## Exercice 2. Partitionnement LVM

---

2.

Créer partition LVM

```
Lancer fdisk, créer une nouvelle partition puis changer le type (commande t) et donner  
le type 8e (LVM)
```

3.

*pvcreate*, crée un volume physique LVM. On vérifie qu'il est bien créé avec la commande *pvdisplay*

```
sudo pvcreate /dev/sdb1
```

4.

*vgcreate*, crée un groupe de volumes. *vgdisplay* pour vérifier

```
sudo vgcreate vg00 /dev/sdb1
```

5.

Créez un volume logique appelé *lvData* occupant l'intégralité de l'espace disque disponible

```
sudo lvcreate -n lvData -l 100%FREE vg00
```

6.

créez une partition formater en *ext4*, et monter au demarrage dans /data.

```
fdisk /dev/mapper/vg00-lvData  
mkfs.ext4 /dev/mapper/vg00-lvData  
Ubuntu mappe les périphériques physiques en périphériques virtuels de plus haut niveau
```

7.

Création d'un second disque

```
fdisk /dev/sdc  
pvcreate /dev/sdc1
```

8.

Pour ajouter un nouveau disque au groupe de volumes

```
sudo vgextend vg00 /dev/sdc1
```

9.

lvresize (ou lvextend) pour agrandir le volume logique et redimensionner le système de fichier (resize2fs)

```
sudo lvresize -l 2046 /dev/mapper/vg00-lvData (2046 correspond à 8Gio)
sudo e2fsck -f /dev/mapper/vg00-lvData
sudo resize2fs /dev/mapper/vg00-lvData
```