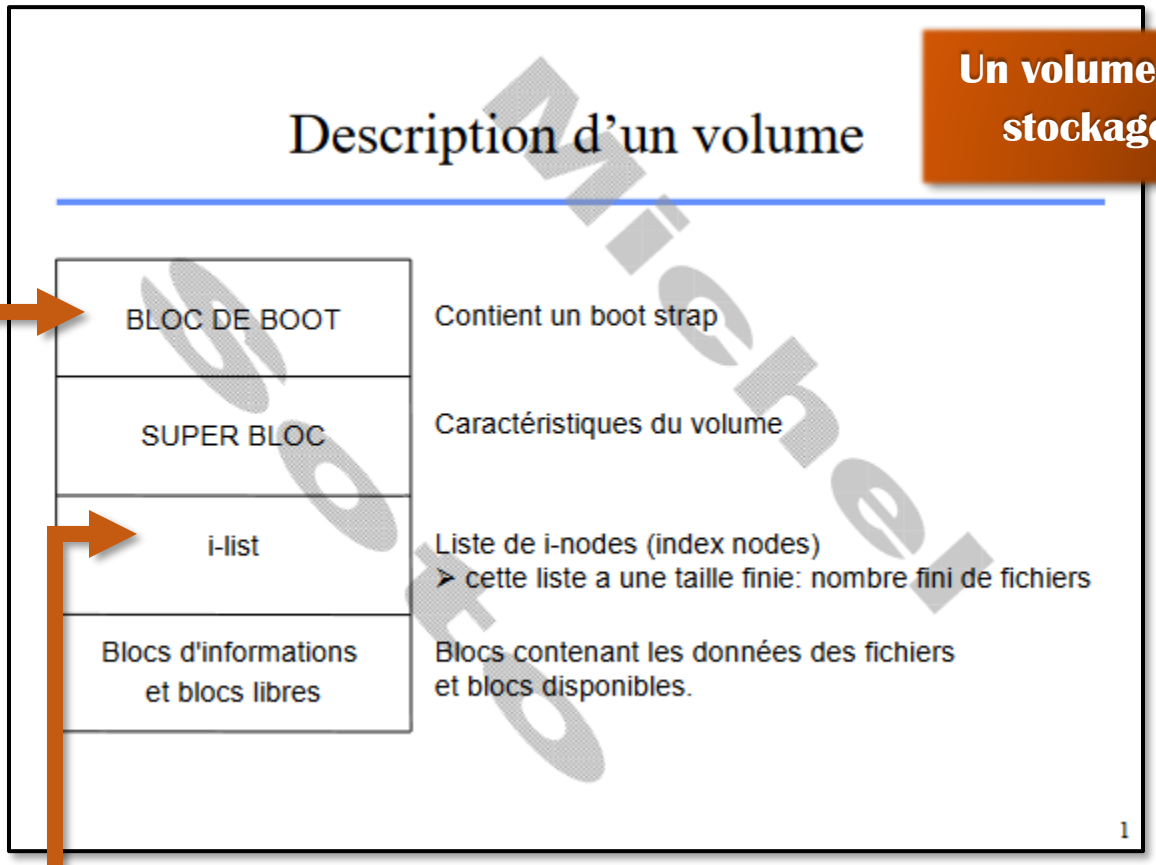
UE Programmation Unix • CM3 5/10/20

Appels Système Gestion des fichiers



Un volume de stockage

BOOT – démarrage de la machine. Contiens un code qu'on appelle le boot strap. Un des premier code qui s'exécute au démarrage de la machine.

Ces neuds se retrouve dans index qui va nous permettre de savoir ou se trouve les info sur le disk.

Description d'un volume

BLOC DE BOOT	Contient un boot strap
SUPER BLOC	Caractéristiques du volume

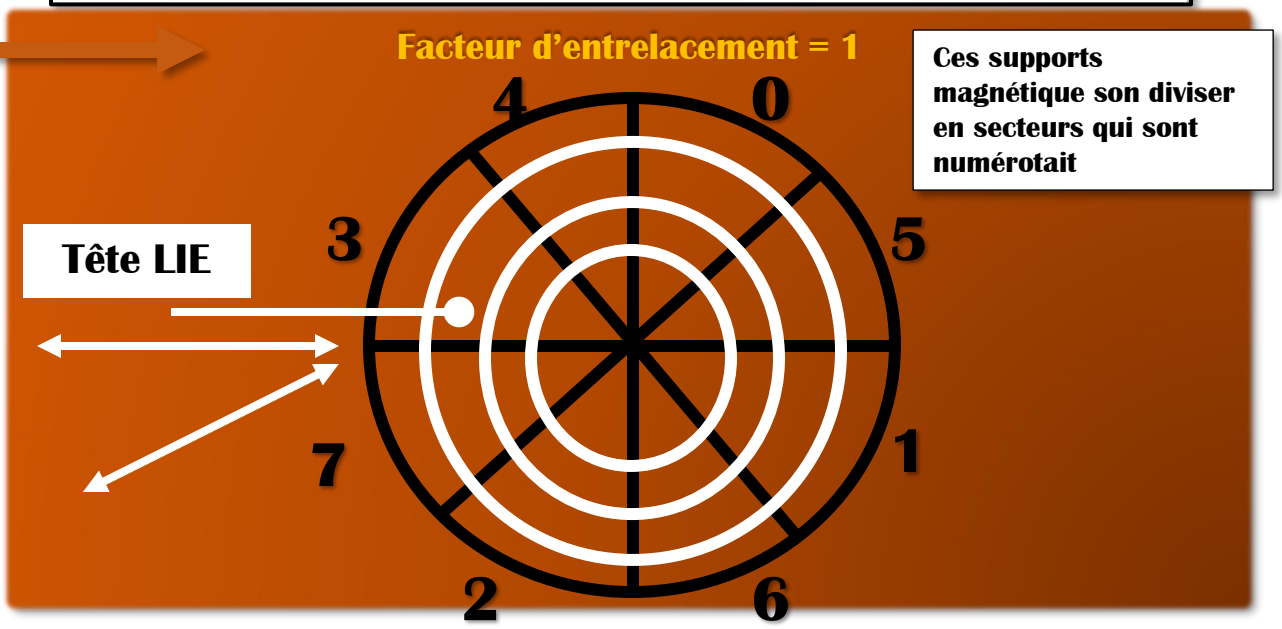
Structure du super bloc

taille de la i-list
taille du volume
taille liste des blocs libres
liste partielle des blocs libres
nombre d'inodes libres
liste partielle des i-nodes libres
verrou liste des blocs libres
verrou liste des i-nodes libres
Date
nombre total des blocs libres
nombre total des i-nodes libres
1er facteur d'entrelacement

Verrous: éviter que plusieurs processus ne manipulent simultanément ces listes

Indique comment sont numérotés les blocs afin d'optimiser le déplacement des têtes de lecture-écriture

2



Ces supports magnétique son diviser en secteurs qui sont numérotait

Répertoire / fichier régulier

Structure d'une i-node

Type de fichier
Protection
Identité du propriétaire (uid)
Groupe du propriétaire (gid)
Nombre de liens
Taille du fichier
Date de dernière modification
Date de la dernière lecture
Zones des adresses des blocs de données

L'i-node décrit le fichier et indique sa localisation physique

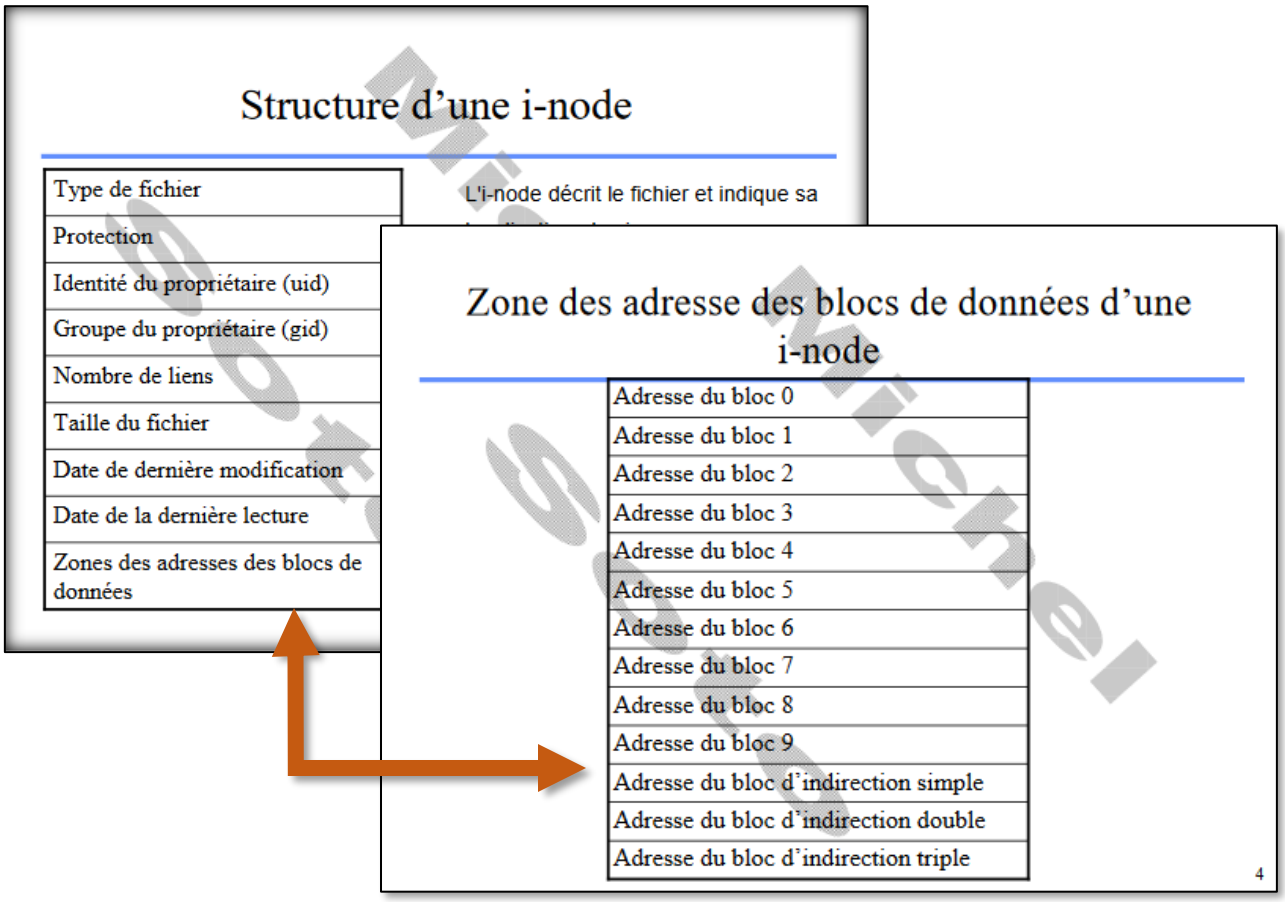
- Il est chargé en mémoire dès qu'on utilise le fichier.
- **Le nom du fichier ne figure pas !!**
- Chaque i-node est identifiée dans la i-liste du volume par un N°

Un lien : comme un raccourci sur Windows

3

Lorsque on ouvre un fichier le système cherche ces info, surtout – ou son les donnes.

Ces info vont être charger en mémoire centrale qu'on a besoin dun fichier.

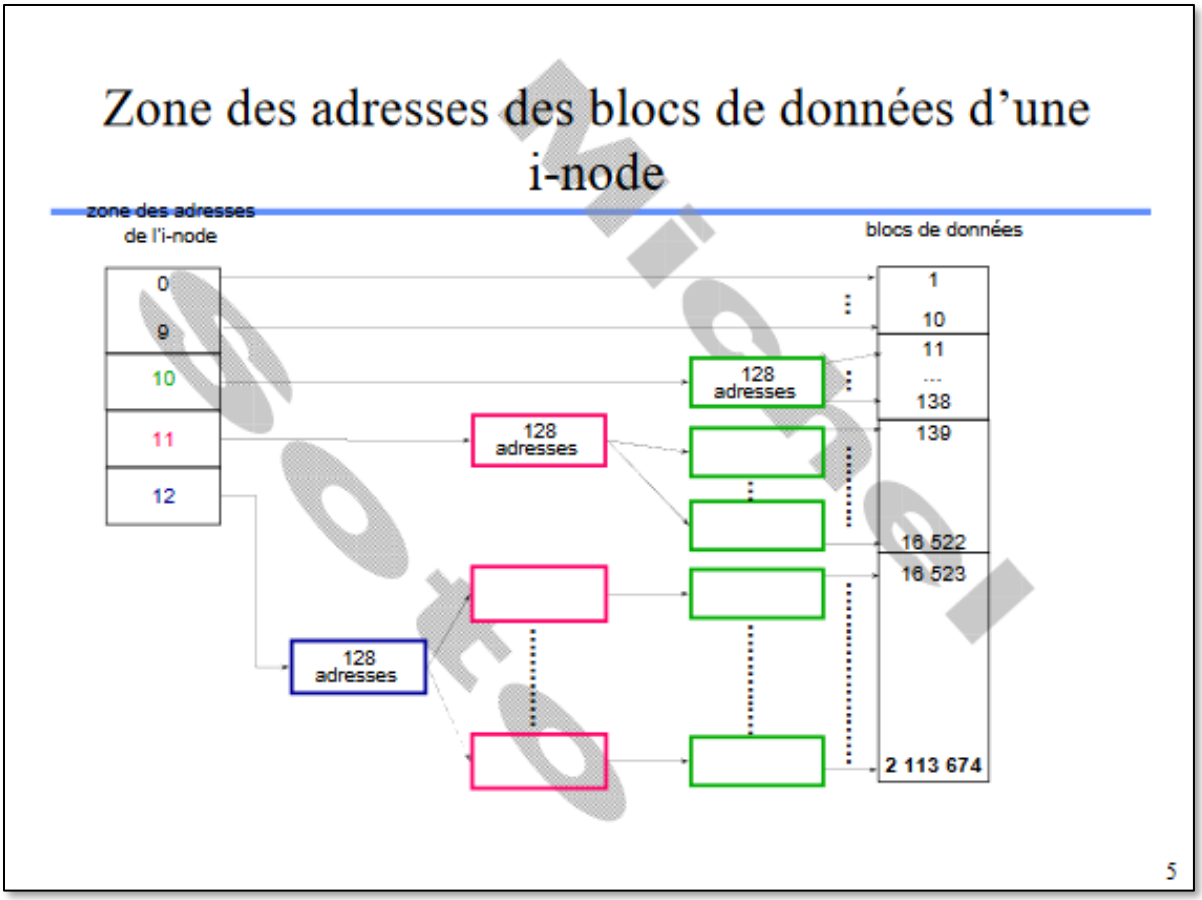


Le bloc sur le disk ca corespont a 1 secteur

Ce sont des adress direct – on tombe sur un block qui contient les donnes du fichier.

Adress du bloc dindirection simple – on trouve une autre adress la ou se trouve les donnes.

Q'on on peut plus faire ca on va vaire un bloc dindirection triple.



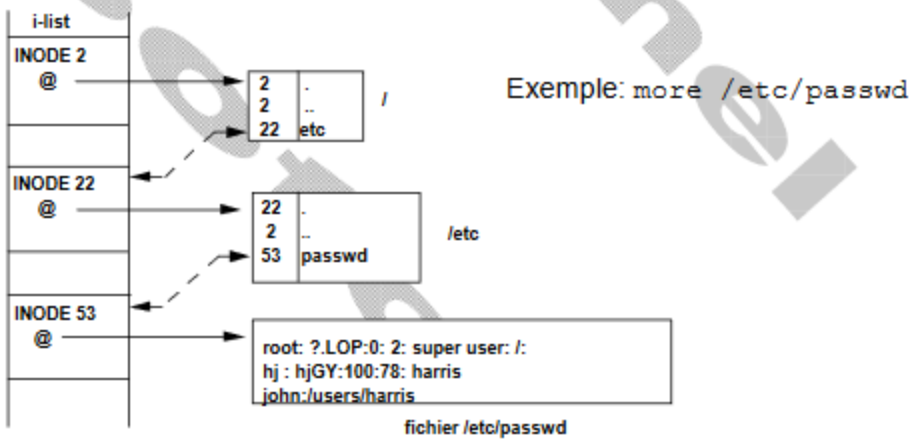
Ca resume a quoi correspond un fichier sur un disk.

Quant jai consommer mes 128 adress jai plus de bloc dinteraction simple
je passe au bloc dinteraction double (128 adress)
et après les 128 adress je passe au bloc dinterdiction triple.

On voit que dans les i-nodes ya pas les non de fichier. Ces la qu'intervienne les répertoires

Utilisation d'un fichier

- Les répertoires servent à maintenir la correspondance N°i-node←→Nom de fichier
- Les répertoires sont des fichiers et possèdent donc une i-node
- Le répertoire racine « / » possède toujours l'i-node n°2



6

Nous quant on fait un ls on voit que le noms du fichier mes les i-nodes ison on les voit en écrivant
ls -i

imaginent que qulquin veut voire le contenu du fichier des mot de passe.

- **La racine cest un repertoire.**
- **Fichier au sense large : soit des fichier regulier soit des répertoires**
- **En face du node deux ya un point (repoirtoire courent)**
- **Pour la racine . = ..**
- **A un moment donnes le system cherche la chaine de caractere etc**
- **Le system va charcher dans ca i-liste le fichier repertoire etc**
- **Le repertoire au dessu de etc cest 2**
- **...**
- **Il cherche le fichier passwd**
- **Cest ecrit que cest dans l'i-node 53**
- **Il va chercher l'i-node 53.**
- **Si on modifie le nom dun fichier**
- **Quelle est la conssequence de changement du non de fichier ?**
- **L'i-node resste le même !**
- **C just un changement de node**
- **Cp cest pas pareill ca cree une copie.**

Tables en mémoire exploitées par le noyau
u_ofile

- **u_ofile:** table des descripteurs de fichiers d'un processus
  - ➢ Chaque processus possède la sienne
    - ➢ Vision PAR PROCESSUS des fichiers ouverts
- Lors de l'ouverture d'un fichier, le noyau lui associe une entrée dans cette table
- Se trouve dans le structure U associée à chaque processus

```
struct user{
        struct inode *u_cdir; <-- pointe sur le répertoire courant
        struct inode *u_rdir; <-- pointe sur le répertoire racine
                                  du volume
        short u_cmask <-- protection
        struct *u_ofile[NOFILE] <-- NOFILE: nombre de fichiers
                                    maximun ouverts par un processus
        ....
       }
```

7

Chaque prossesus se voit associer une structure u (comme utilisateur)

Il y a les fichier dans le processus…

Il y a tjr une entree user open file (u_ofile) une table dans laquelle se trouve tt les ouvertures de fichier(pas pareill que fichier ouvert !)

Un même fichier peut être ouvert plusieurs fois

Je peut dire que jai 3 fichier ouvert mais ils on fait …. Plusieurs ouverture (une fois lecture, une fois ecriture)

Tables en mémoire exploitées par le noyau
u_ofile

- Descripteurs standards dans la u_ofile
  - 0 (stdin) : flux correspondant à l'entrée standard
  - 1 (stdout) : flux correspondant à la sortie standard
  - 2 (stderr) : flux correspondant à la sortie d'erreur standard
- Exemple

```
...
fprintf(stderr,"erreur numero %d", errno);
…
```

- Chaque entrée associée à un fichier ouvert pointe vers une entrée de la table des ouvertures de fichiers: file table

8

Stderr : flux different

Par exemple si on reffere stdout on a tjr les ereurs a lecrants.

File table – la table des fichier

Chaque processus a sa u_ofile

Ouverture de fichiers

Et par concequance les fichies ouvert.

Tables en mémoire exploitées par le noyau
file table

- Contient les informations sur tous les fichiers ouverts dans le système par l'ensemble des processus à un instant donné.
  - ➢ Une entrée utilisée par ouverture de fichier
- Permet à plusieurs processus de même filiation de partager un fichier ouvert

```
struct file{
  char f_flag   <-- mode d'ouverture écriture, lecture, pipe
  cnt_t f_count <-- nombre de processus qui accèdent au fichier
  struct inode *f_inode <-- pointeur vers la table des i-nodes
    .........
}
```

9

Chaque entree.. ouverture dun fichier et pas un fichier ouvert.

Si yavait pas cette file table le fork.... Pour lheritage des fichies uvert par un processus père.

Un fichier peut être utiliser par plusieurs processus (de la même famille ou pas)
et enfin on a un pointeur vers une 3ieme table

Tables en mémoire exploitées par le noyau
i-node table

- Permet la localisation physique du fichier.
- L'i-node de la i-list du fichier ouvert est chargée dans une entrée de cette table lors de sa première ouverture.
  - Vision GLOBALE des fichiers ouverts
  - Une entrée utilisée par fichier ouvert

```
struct inode {
   flag   <-- indique si l'i-node est
              vérouillée, modifiée,...
    count <-- nombre de références
    dev   <-- device de résidence
    number <-- son numéro (sa place dans la i-list)
   }
```

10

Chaque fois qun fichier est utiliser ...

Plus simple dacedder a la mémoire centrale que ....

Flag : essque l'i-node était modifier.

Après ya un pointer vers le device de ...


Emission d'un signal
kill