

Systemes d'Exploitation:
La Gestion des Fichiers

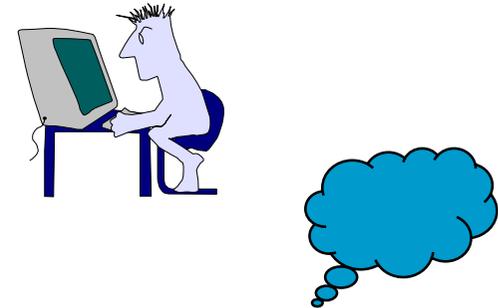
Hafid Bourzoufi
Université de Valenciennes

Didier Donsez
Université Grenoble 1
Didier.Donsez@imag.fr

Systeme de fichiers

■ De point de vue de l'utilisateur

- Interface du service de fichiers



■ De point de vue du concepteur

- Mise en œuvre d'un système de fichiers



Concept de fichier

■ Définition

- Un fichier est une unité de stockage logique de l'information.
 - Abstraction des propriétés physiques des dispositifs de stockage
- la correspondance est établie par le SE

■ Attributs des fichiers

- Nom, taille, type, protection, date, propriétaire, ...

■ Opérations sur les fichiers :

- Création, Ecriture/Lecture, Suppression , Concatenation (Append) ...

Type et structures des fichiers

■ Type de fichiers

- exécutable, commande, texte, ...

■ Certains SE supportent et reconnaissent le type de fichiers

- Windows utilise les suffixes des noms de fichiers : .exe, .bat, .txt, .com...
 - Un type est associé à une application
 - .htm, .html -> netscape.exe
 - .pdf -> ghostview.exe (au lieu d'acroread.exe)
 - MacOS utilise un typage présent dans le descripteur
 - Il est formé de 4 lettres pour l'application et de 4 lettres pour le type
 - Unix ne supporte pas le typage des fichiers
 - Cependant l'utilitaire file peut donner le type

■ Structure interne d'un fichier

- suite d'enregistrements (CP/M, VSAM)
- suite d'octets (Unix, MS-DOS)
 - Le SE n'effectue aucune interprétation

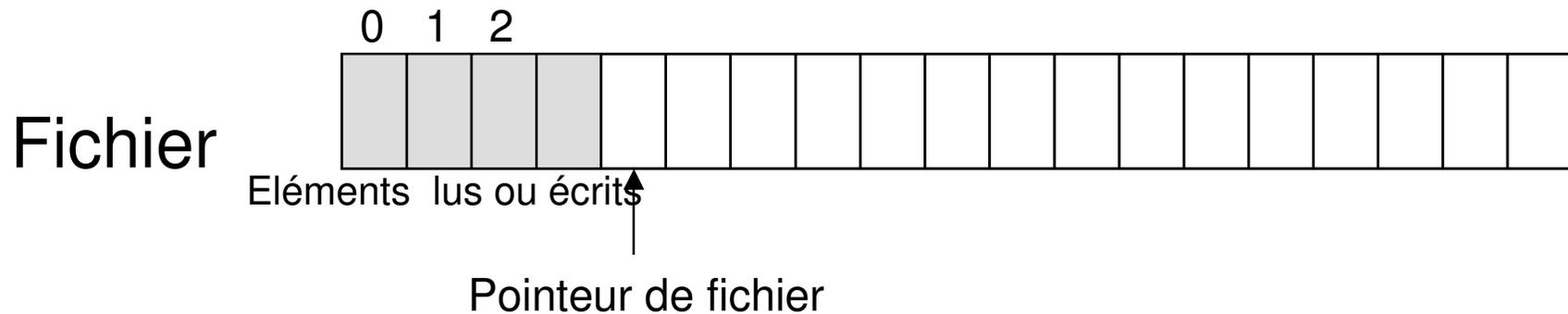
Méthodes d'accès

- Accès séquentiel
- Accès direct (dit aléatoire)
- Accès indexé

Méthodes d'accès (suite)

■ Accès séquentiel

- Les éléments sont lus ou écrits dans l'ordre

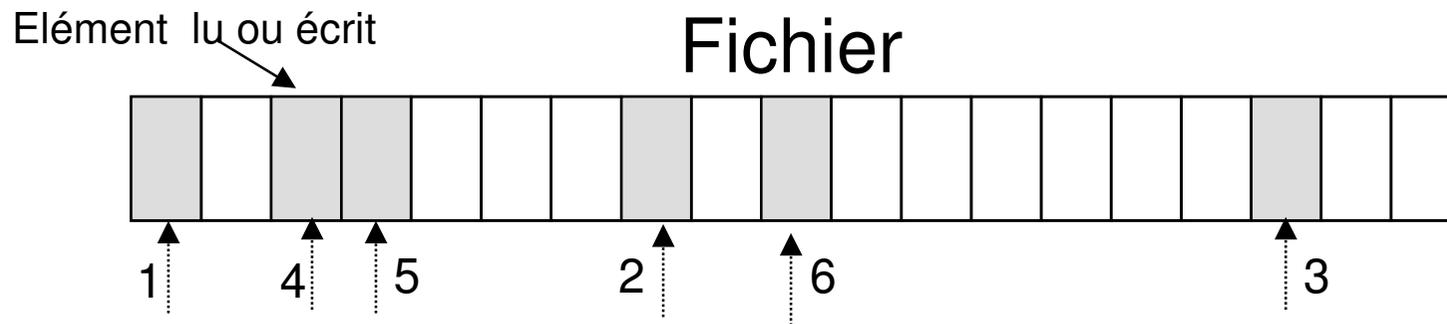


- Méthode adaptée aux supports de stockage séquentiels :
Bandes magnétiques

Méthodes d'accès (suite)

■ Accès direct (aléatoire)

- L'ordre des accès aux éléments est quelconque



- Méthode adaptée aux supports de stockage à accès direct : disques

■ Accès indexé : (généralisation)

- Accès à partir d'une clé

Les Répertoires

■ Objectif

- organiser les fichiers

■ Plusieurs organisations

- répertoire à un niveau
- structure de répertoire à deux niveaux
- organisation arborescente (Unix, Ms-dos)

■ Opérations

- parcours, listage, renommage, ...

Conception d'un système de fichier

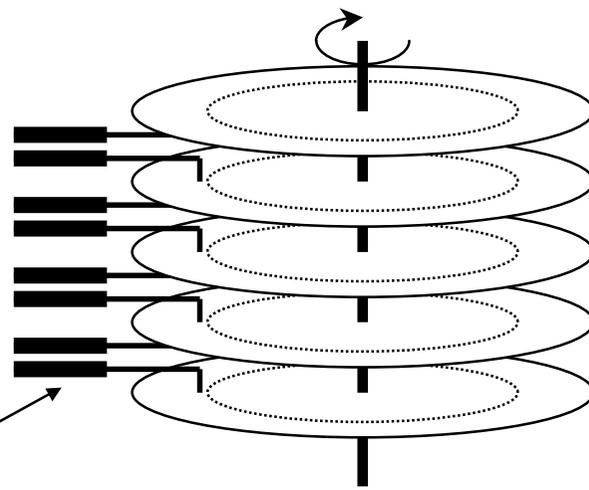
- Organisation des disques
- Méthode d'allocation
- Gestion de l'espace libre
- Performance
- Protection et Sécurité

Disques Mécaniques



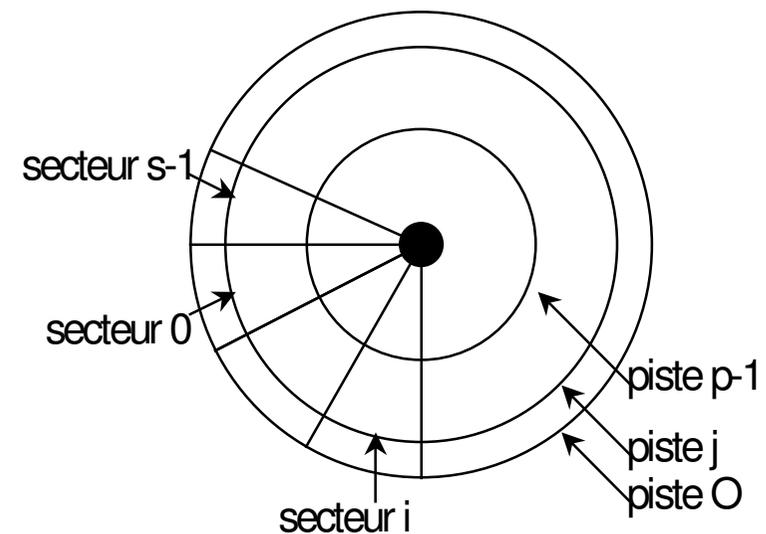
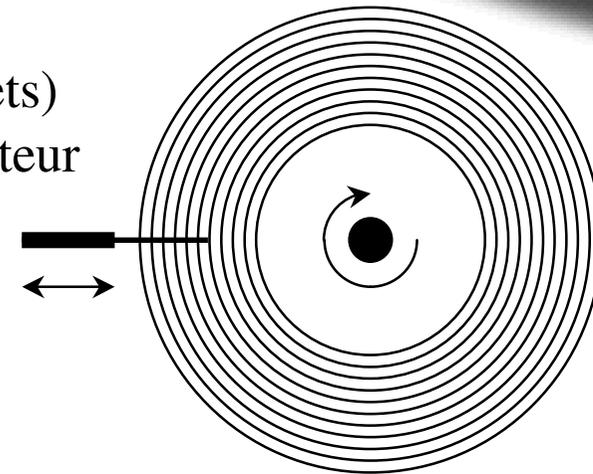
■ Organisation

- blocs de mémoire de taille fixe (N octets)
- disque organisé en Tête, Cylindre, Secteur



Têtes de lecture/écriture
montés sur des bras en mouvement

Axe de rotation



Linéarisation de l'espace disque

■ Linéarisation :

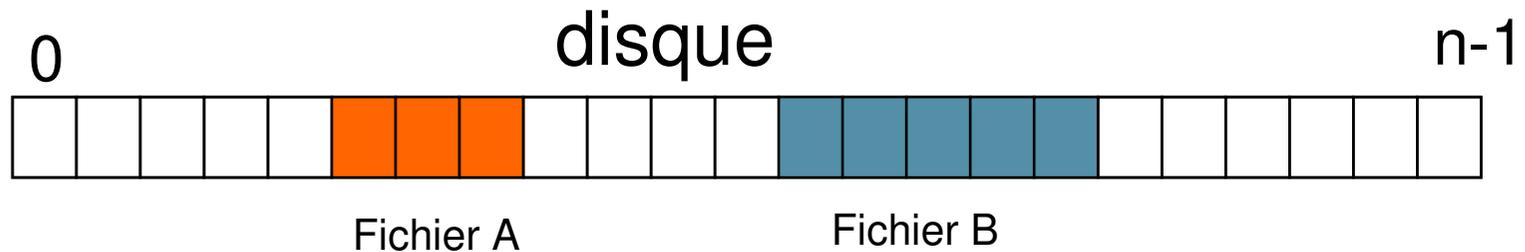
- $(nc, nf, ns) \rightarrow nv = \text{numéro unique (n}^\circ \text{ virtuelle)}$
- $nc : \text{n}^\circ \text{ cylindre} \quad 0 \leq nc < \text{nombre total de cylindres}$
- $nf : \text{n}^\circ \text{ disque} \quad 0 \leq nf < \text{nombre total de disques}$
- $ns : \text{n}^\circ \text{ secteur} \quad 0 \leq ns < \text{nombre de secteurs / piste}$

■ Le choix de la fonction de linéarisation influe sur les performances du SF

- Réduction des déplacements de la tête

Méthodes d'Allocation

- Allocation contiguë : chaque fichier occupe un nombre de blocs contigus sur le disque



- Accès à un bloc :

- $nv = n^{\circ} \text{ bloc logique} + n^{\circ} \text{ du premier bloc}$
 - accès au I^{ème} bloc : $nv = n^{\circ} \text{ du premier bloc} + I$

Allocation Contiguë

■ Avantages :

- Simple à implémenter
- Accès direct aux blocs en temps constant
- Adapté aux supports « Write Once » (CDRom,CDRW)

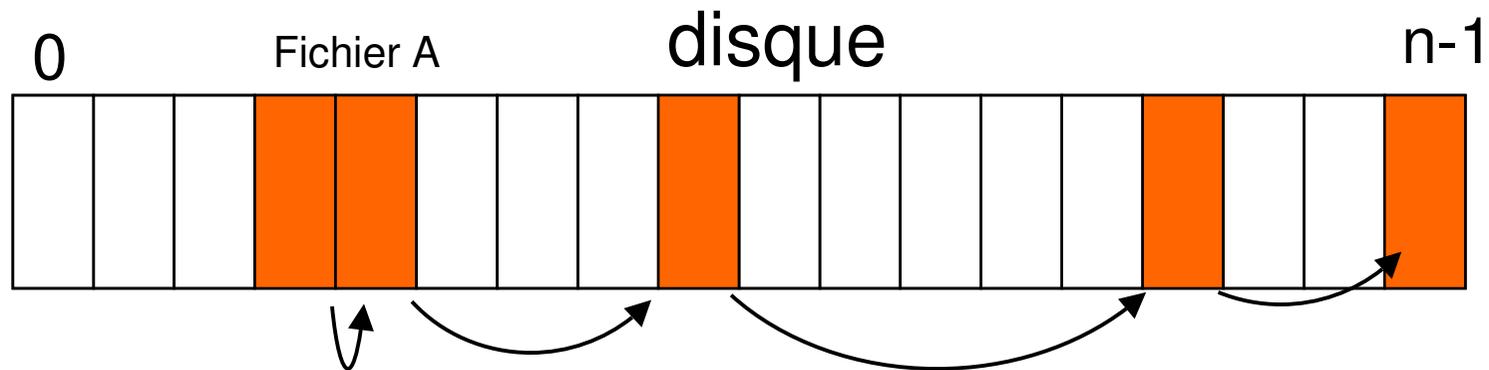
■ Inconvénients :

- Problème : Extension d 'un fichier
 - Nécessite de connaître à l 'avance de la taille du fichier
 - Déplacement possible du fichier lors de son extension
- Problème : Fragmentation
 - L 'espace libre peut être fragmenté en plusieurs trous et aucun n 'est suffisant pour stocker un fichierdéfragmentation

Méthodes d'Allocation

■ Allocation chaînée :

- Un fichier occupe une liste chaînée de blocs sur le disque
- Chaque bloc contient une partie des données et un pointeur sur le bloc suivant



Allocation chaînée

■ Avantages

- Possibilité d 'étendre un fichier
- Allocation par bloc individuel : Tout bloc libre peut être utilisé pour satisfaire une requête d 'allocation

■ Inconvénients

- Solution non adaptée à l 'accès direct
 - L 'accès à un bloc quelconque nécessite l 'accès à tous les blocs qui le précèdent
- Les pointeurs sont stockés sur disque

Allocation chaînée et indexée

■ Idée : Séparer les pointeurs et les données

■ Technique

- Utilisation d'une table d'allocation de fichier
 - (FAT : File Allocation Table)
- A chaque bloc est associée une entrée dans la FAT qui contient le n° du bloc suivant
 - Méthode Utilisée dans MS-DOS et OS/2

0	3
1	Fin de fichier
2	4
3	6
4	1
5	
6	Fin de fichier
	⋮
	-

Fichiers	blocs occupés
A	0 3 6
B	2 4 1

Allocation chaînée et indexée

■ Avantages

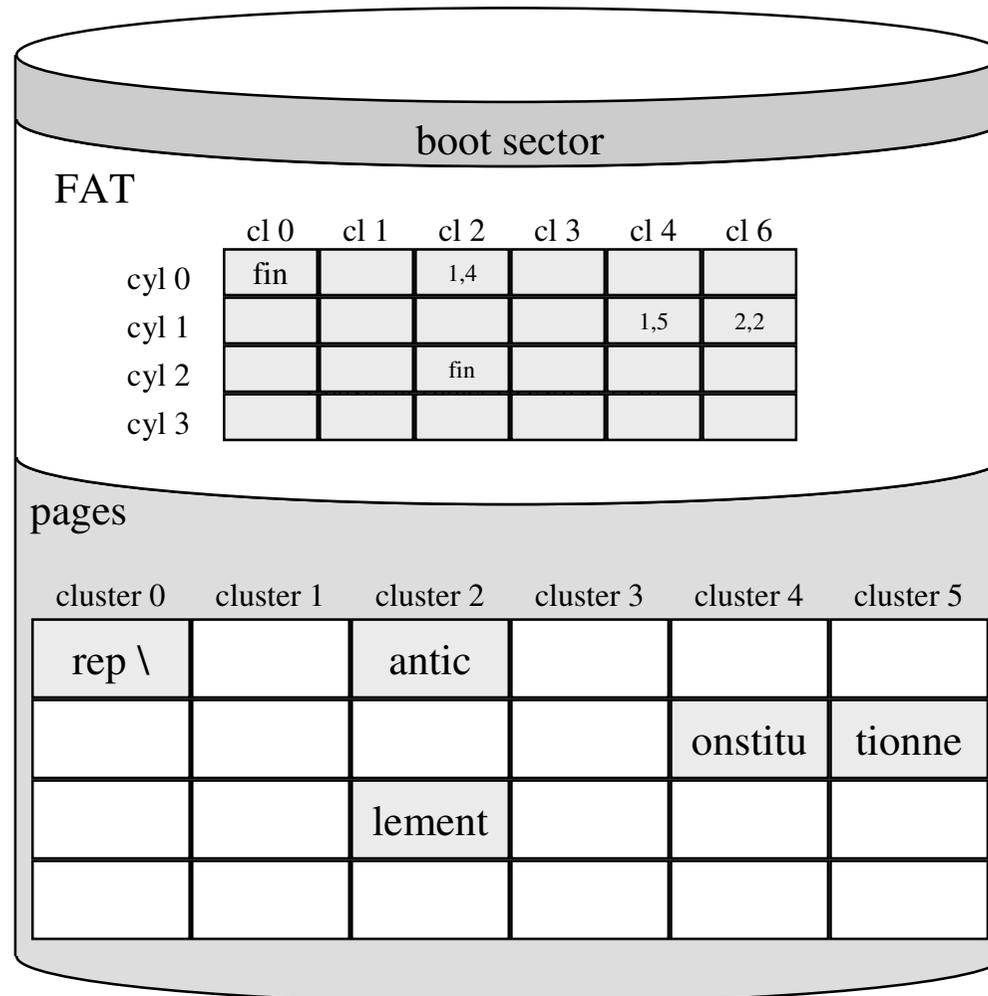
- Extension des fichiers
- les blocs de données ne contiennent pas les pointeurs
- accès direct facile
 - utilise un mécanisme d'ombre pour protéger la FAT

■ Inconvénients

- Occupation de la mémoire centrale par la FAT
- Problème des disques de grande capacité
- Une table pour un disque de 1Go en blocs de 1Ko occuperait 4 Mo (+ 4Mo pour la FAT « ombre »)

■ Voir les solutions de Windows NT : CDFS et NTFS

SGF FAT (MSDOS & Windows)



Allocation par nœud d 'information

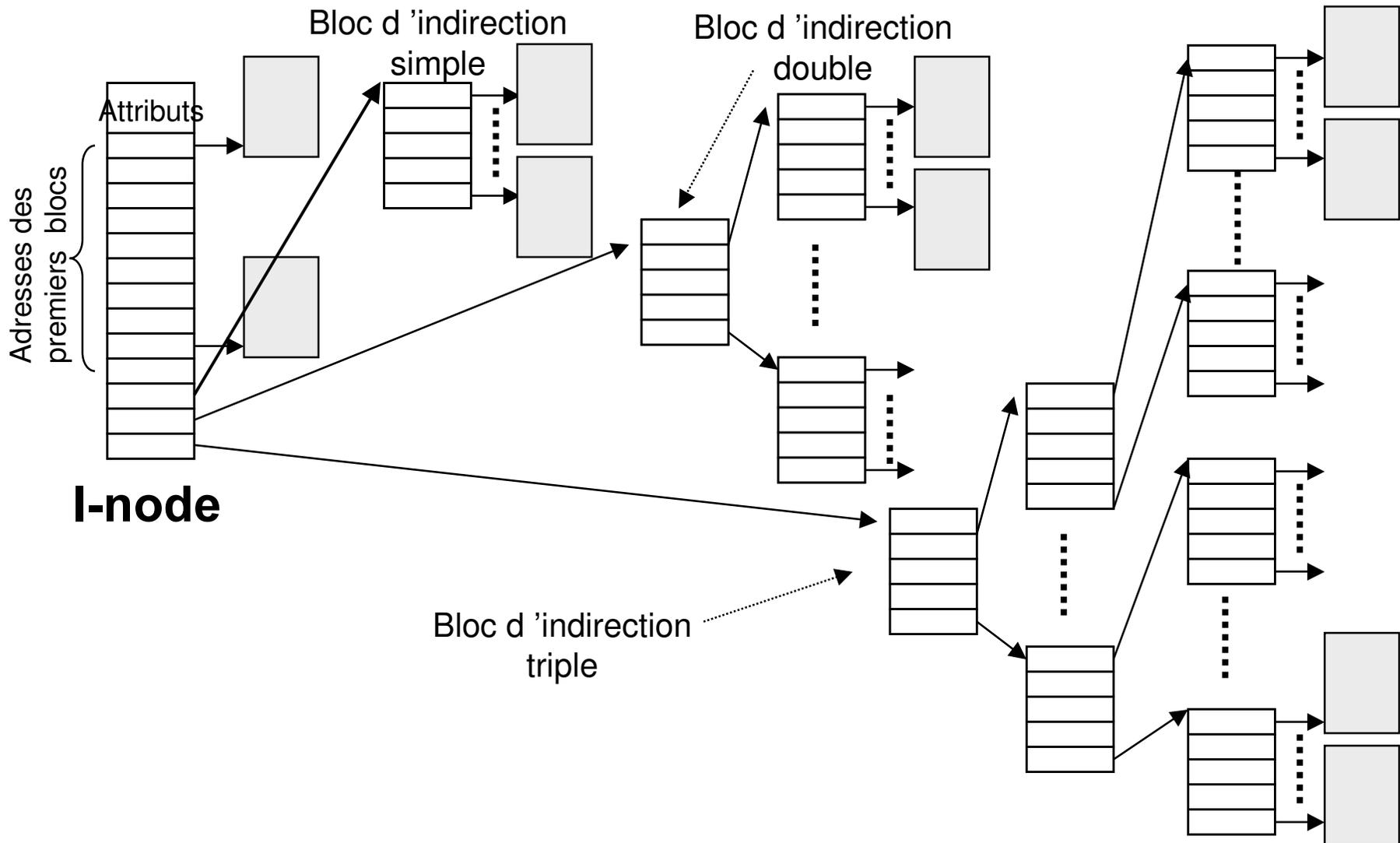
■ Idée

- Eclater la FAT en plusieurs petites tables appelées nœuds d 'informations (i-node)
- A chaque fichier est associé un nœud d 'information
- Chaque table contient les attributs et les adresses sur le disque des blocs du fichier

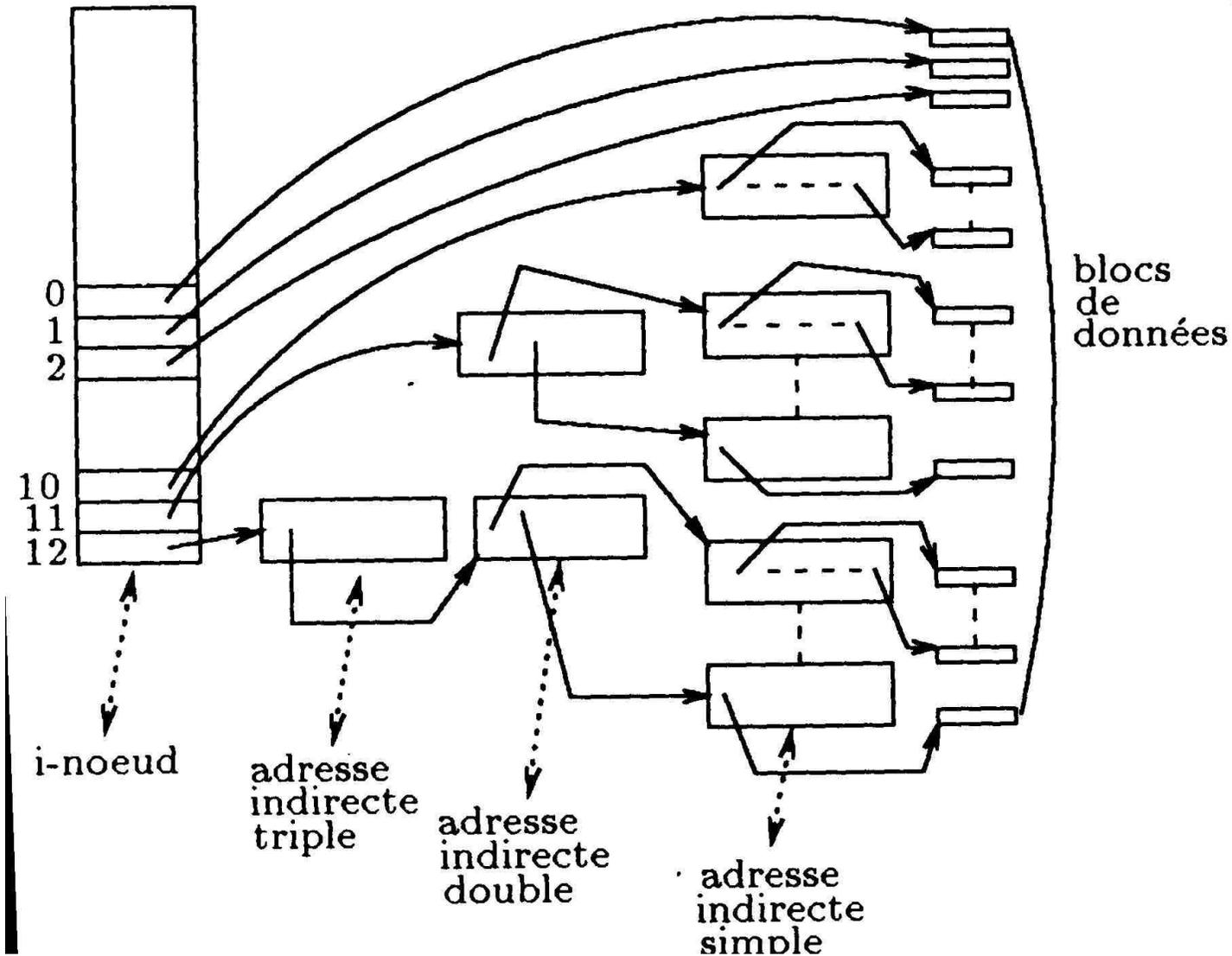
■ Unix

- La table est hiérarchisé sur Unix
- FS System V
 - 10 direct, 1 simple indirection, 1double indirection, 1 triple indirection
- BSD Fast FS / UFS
 - 12 direct, 1 simple indirection, 2 double indirection

Structure d'un nœud d'information (FS SYSV)



Structure d'un nœud d'information

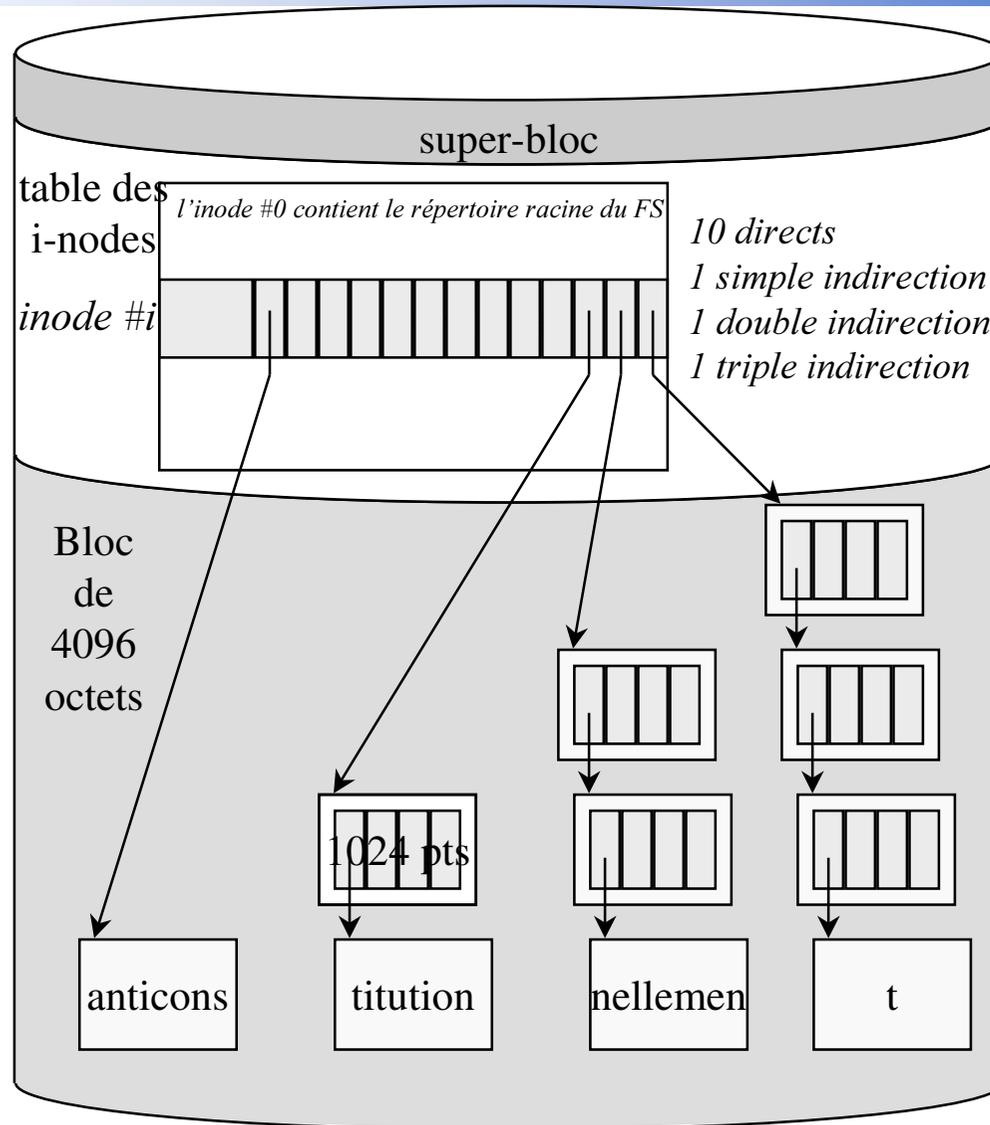


Allocation par nœud d'information

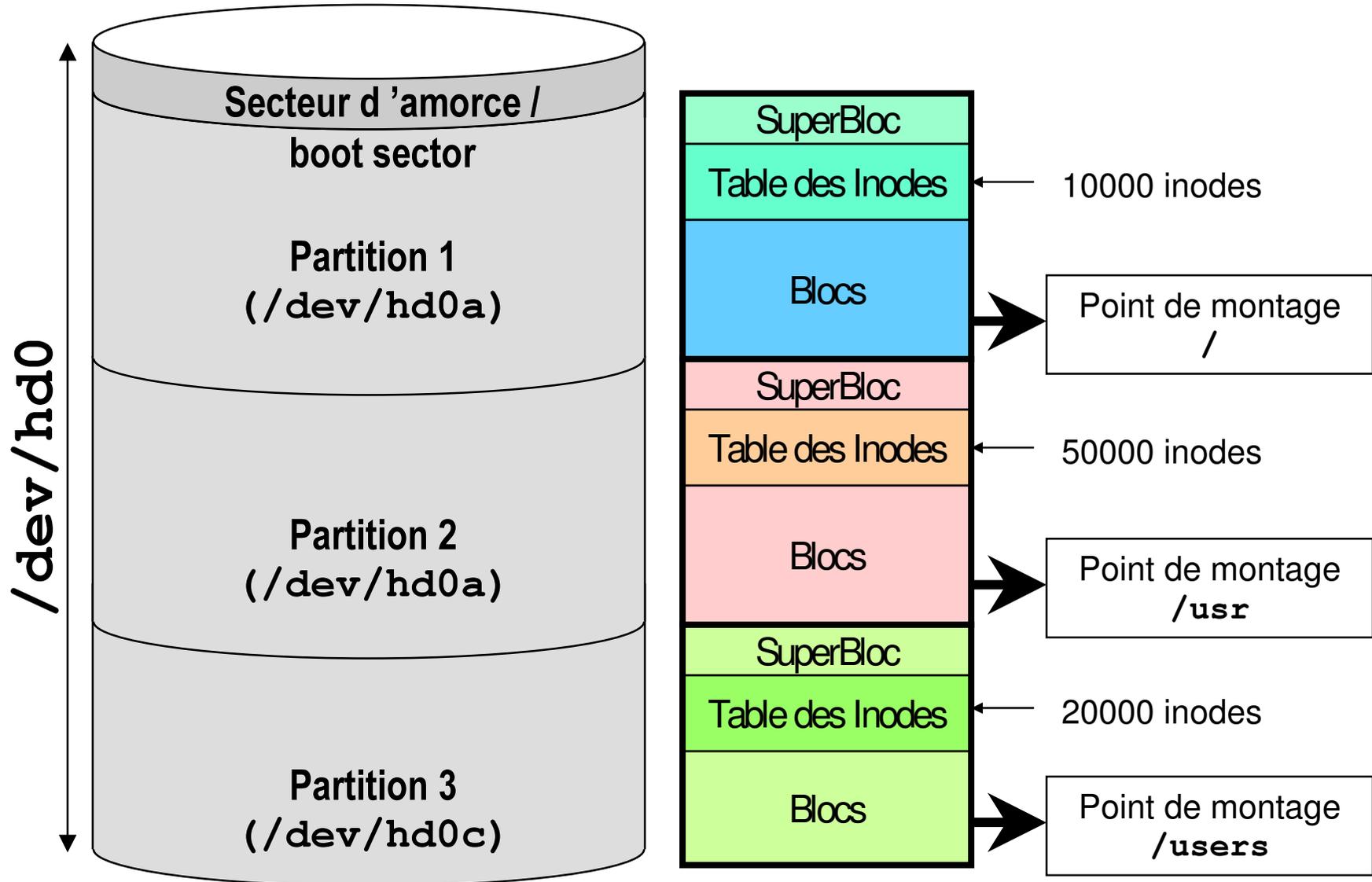
- Seuls les nœuds d'information des fichiers ouverts sont chargés en mémoire centrale
- Allocation par bloc individuel
- Accès direct facile
 - nécessite au maximum 4 accès disque
- Adaptée aux disques de très grande capacité

Exemple

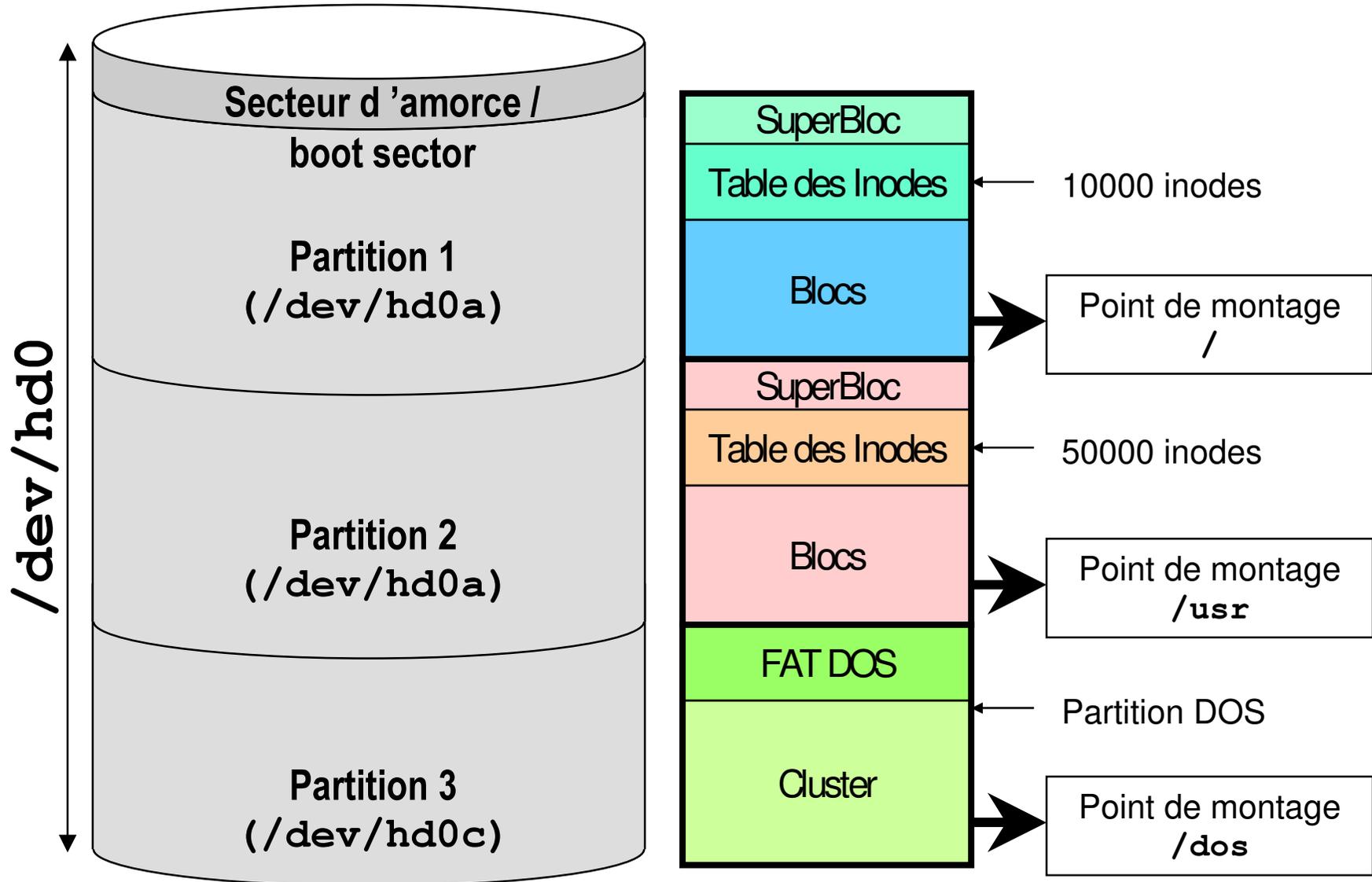
le Système de Fichier d 'Unix System V



Organisation Unix d'un disque en plusieurs File System



Organisation Unix d'un disque en plusieurs File System



Allocation par journalisation (Log File System)

■ SGF performant pour les stations

- Le nombre d'écritures domine le nombre de lecture

■ Principe

- Constat : Les disques sont performants
pour les lectures et les écritures séquentielles
- Les écritures de synchronisation du cache
se font séquentiellement
- Le disque est organisé en segments de bloc contigus

Journaling Flash File System (JFFS)

- SGF destiné aux mémoires Flash
- remarque sur les Flash
 - Temps d'accès uniforme
 - Nombre limité des réécritures de points mémoires (~100000).
- Technique :
 - la réécriture d'une "page" se fait toujours à un autre endroit (journal à gestion cyclique) pour éviter l' »usure des points mémoires.
- Implantations
 - jffs, jffs2 pour Embedded Linux
 - Windows CE, PalmOS, ...
- Drivers Flash
 - DiskOnChip®, MTD (Memory Technology Device), ...

Global File System

- Partage transparent de plusieurs devices raw entre plusieurs nœuds
 - Permet d'émuler un SAN
 - Configurable en mirroring
- Implémentations
 - *RedHat GFS*

