ASR2-Système : systèmes de fichiers

Semestre 2, année 2009-2019

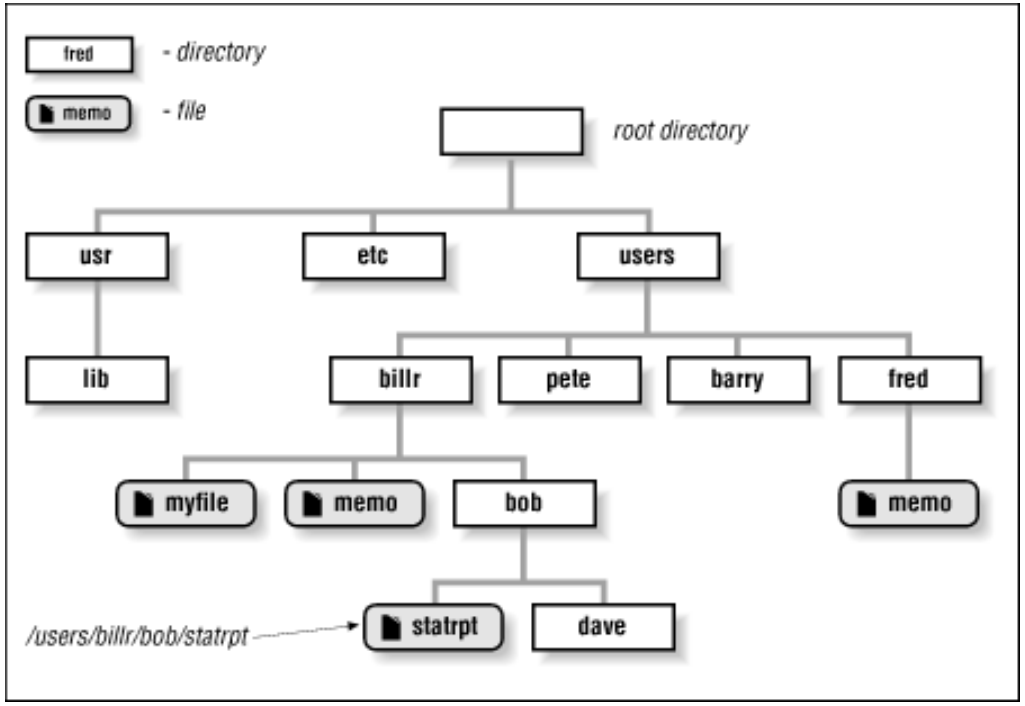
Département d'informatique
IUT Bordeaux 1

Mai 2009

Système de fichiers
= structure de données

représentation de
fichiers et répertoires
sous forme de 0 et de 1

stockés en
mémoire secondaire



pour l'utilisateur : arborescence

fichiers/répertoires
accessibles par leur nom
(chemin d'accès)

Un fichier a
un contenu
des méta-données

Méta-données d'un fichier

Informations

- taille
- propriétaire
- droits d'accès
- date de création
- date de dernier accès
- ...

Système de Gestion de Fichiers : fonctions

- Manipulation des fichiers : créer/détruire des fichiers, ...
- Allocation de la place sur mémoires secondaires
- Localisation des fichiers : accès au contenu
- Sécurité et contrôle des fichiers
- Fiabilité en cas de panne
- ...

ici : quelques idées sur la représentation

1 - Catalogue de fichiers

Catalogue de fichiers

VTOC = Volume Table of Contents (IBM)

- pas de répertoires,
- fichiers contigus

Catalogue de fichiers

Table des fichiers

nom du fichier	position du premier bloc	taille
CLIENTS	10	50
PRODUITS	60	500
FACTURES	560	2000
...	...	...

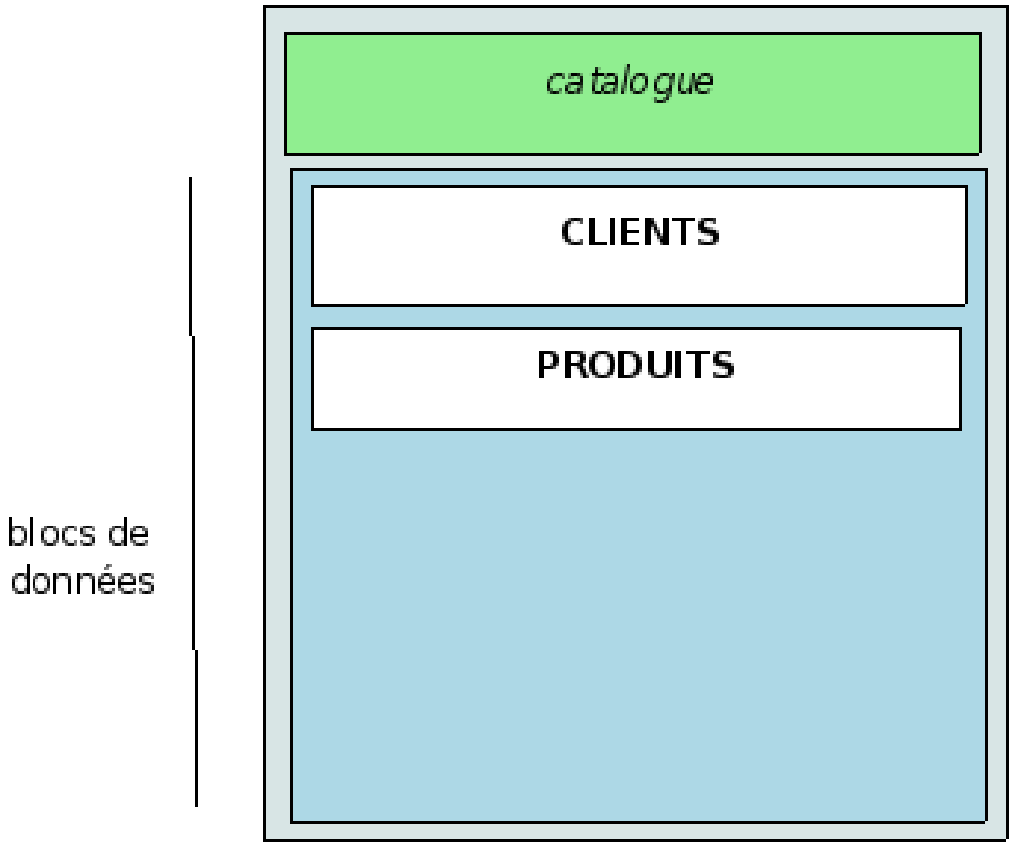
située au début du disque.

Dans le catalogue :
liste des
espaces libres

Reste du disque :

blocs de données

(contenu des fichiers)



VTOC : occupation du disque

Gestion de l'espace

- Espaces contigus
- Réservation d'espace à la création d'un fichier
- Restitution quand le fichier est supprimé
- Extension des fichiers ?

Avantages/Inconvénients

Avantages

- simplicité
- performances

Avantages/Inconvénients

Inconvénients

Perte de place causée par

- réservations non utilisées
- espaces contigus de taille variable

Solutions

- 1 fichier = plusieurs zones, allouées au besoin (dynamiquement)

Exemple

Réservation d'un fichier de 20 Ko + 5 extensions de 10 Ko

- utilitaire de réorganisation du disque

2 - Table d'allocation

fichiers non contigus :
allocation plus facile à gérer

table supplémentaire :
index du bloc suivant

Catalogue de fichiers

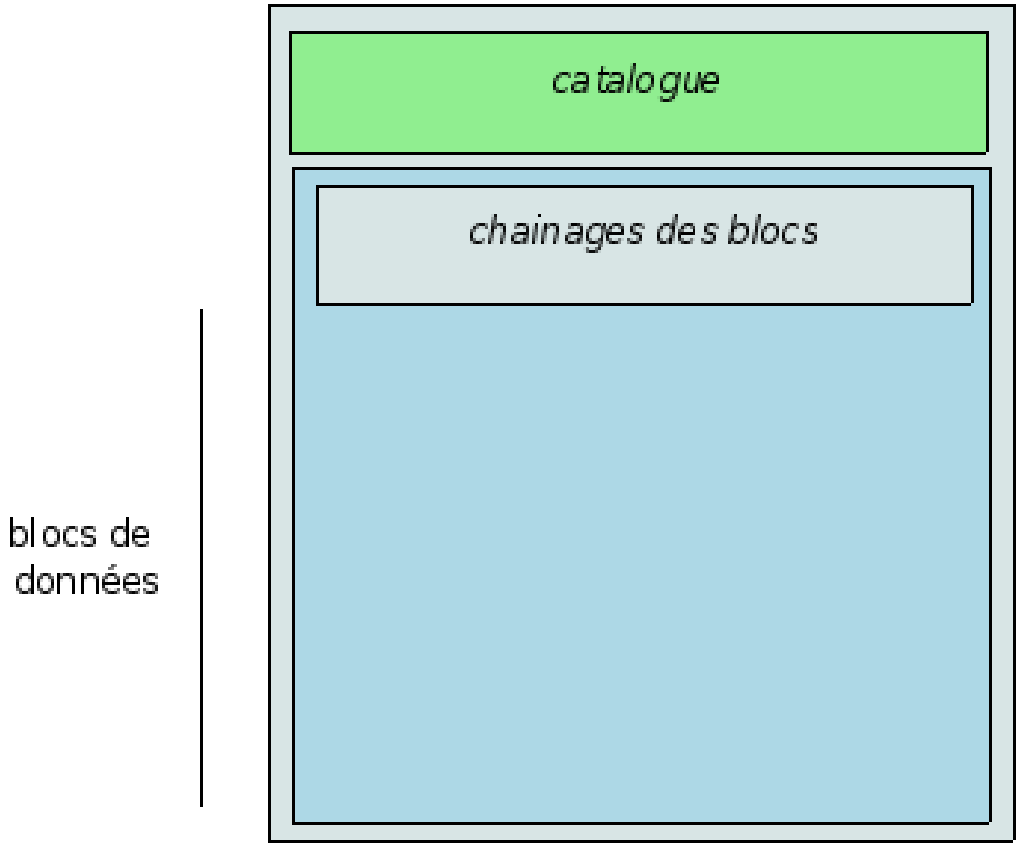
- Table des fichiers

nom du fichier	position du premier bloc	taille
CLIENTS	10	50
PRODUITS	12	500
FACTURES	15	2000
...	...	...

- et table de chaînage des blocs

indice	...	10	11	12	13	...
suivant	...	11	20	13	14	...

- blocs de données



FAT : occupation du disque

Autre représentation

Table des blocs intégrée dans le catalogue

nom	taille	B1	B2	B3	B4	....	B16
CLIENTS	3	10	11	22	-		-
PRODUITS	4	20	11	42	-		-
...							
FACTURES	20	101	102	103	104	...	116
...							
FACTURES	-	117	118	119	120	...	-
...							

(utilisation de “lignes de continuation”)

- Technique de représentation utilisée dans CP/M

Autre représentation (UNIX)

A chaque fichier est associé un *i-node*

- des attribut (taille, propriétaire, droits...)
- les adresses de ses premiers blocs de données

Autre représentation (UNIX)

A chaque fichier est associé un *i-node*

- des attribut (taille, propriétaire, droits...)
- les adresses de ses premiers blocs de données
- l'adresse d'un bloc d'indirection simple qui contient d'autres adresses de blocs de données.

Autre représentation (UNIX)

A chaque fichier est associé un *i-node*

- des attribut (taille, propriétaire, droits...)
- les adresses de ses premiers blocs de données
- l'adresse d'un bloc d'indirection simple qui contient d'autres adresses de blocs de données.
- l'adresse d'un bloc d'indirection double

Autre représentation (UNIX)

A chaque fichier est associé un *i-node*

- des attribut (taille, propriétaire, droits...)
- les adresses de ses premiers blocs de données
- l'adresse d'un bloc d'indirection simple qui contient d'autres adresses de blocs de données.
- l'adresse d'un bloc d'indirection double qui contient d'autres adresses de blocs d'indirection simple

Autre représentation (UNIX)

A chaque fichier est associé un *i-node*

- des attribut (taille, propriétaire, droits...)
- les adresses de ses premiers blocs de données
- l'adresse d'un bloc d'indirection simple qui contient d'autres adresses de blocs de données.
- l'adresse d'un bloc d'indirection double qui contient d'autres adresses de blocs d'indirection simple qui contiennent d'autres adresses de blocs de données.
- l'adresse d'un bloc d'indirection triple

Autre représentation (UNIX)

A chaque fichier est associé un *i-node*

- des attribut (taille, propriétaire, droits...)
- les adresses de ses premiers blocs de données
- l'adresse d'un bloc d'indirection simple qui contient d'autres adresses de blocs de données.
- l'adresse d'un bloc d'indirection double qui contient d'autres adresses de blocs d'indirection simple qui contiennent d'autres adresses de blocs de données.
- l'adresse d'un bloc d'indirection triple qui contient d'autres adresses de blocs d'indirection double

Autre représentation (UNIX)

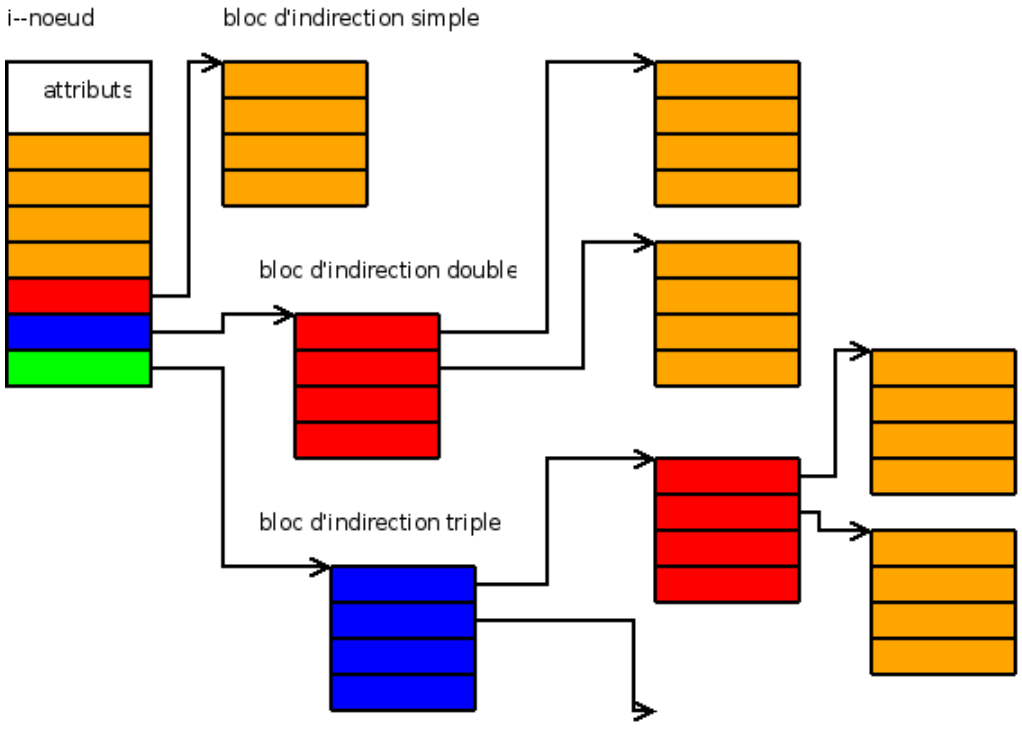
A chaque fichier est associé un *i-node*

- des attribut (taille, propriétaire, droits...)
- les adresses de ses premiers blocs de données
- l'adresse d'un bloc d'indirection simple qui contient d'autres adresses de blocs de données.
- l'adresse d'un bloc d'indirection double qui contient d'autres adresses de blocs d'indirection simple qui contiennent d'autres adresses de blocs de données.
- l'adresse d'un bloc d'indirection triple qui contient d'autres adresses de blocs d'indirection double qui contiennent d'autres adresses de blocs d'indirection simple

Autre représentation (UNIX)

A chaque fichier est associé un *i-node*

- des attribut (taille, propriétaire, droits...)
- les adresses de ses premiers blocs de données
- l'adresse d'un bloc d'indirection simple qui contient d'autres adresses de blocs de données.
- l'adresse d'un bloc d'indirection double qui contient d'autres adresses de blocs d'indirection simple qui contiennent d'autres adresses de blocs de données.
- l'adresse d'un bloc d'indirection triple qui contient d'autres adresses de blocs d'indirection double qui contiennent d'autres adresses de blocs d'indirection simple qui contiennent d'autres adresses de blocs de données.



I-nodes et blocs d'indirection

Chiffrage

Supposons :

- des blocs de 4 Ko (2^{12})
- des adresses sur 32 bits

Capacité maximale du disque ?

Chiffrage

Supposons :

- des blocs de 4 Ko (2^{12})
- des adresses sur 32 bits

Capacité maximale du disque ?

En théorie, le disque peut contenir

$$2^{32} \text{blocs}$$

Chiffrage

Supposons :

- des blocs de 4 Ko (2^{12})
- des adresses sur 32 bits

Capacité maximale du disque ?

En théorie, le disque peut contenir

$$2^{32}\text{blocs}$$

soit

$$2^{32} \times 2^{12} = 2^{44}\text{octets} = 16\text{Tera octets}$$

Chiffrage

Supposons :

- des blocs de 4 Ko (2^{12})
- des adresses sur 32 bits

Capacité maximale du disque ?

En théorie, le disque peut contenir

$$2^{32}\text{blocs}$$

soit

$$2^{32} \times 2^{12} = 2^{44}\text{octets} = 16\text{Tera octets}$$

Question : taille maximum d'un fichier ?

Taille maximum d'un fichier

- une adresse = 32 bits = 4 octets
- un bloc = 4 Ko :

Taille maximum d'un fichier

- une adresse = 32 bits = 4 octets
- un bloc = 4 Ko :peut contenir 4096 = 1024 adresses.

Taille maximum d'un fichier

- une adresse = 32 bits = 4 octets
- un bloc = 4 Ko :peut contenir 4096 = 1024 adresses.

Donc

- un bloc d'indirection simple conduit à 1024 blocs de données

Taille maximum d'un fichier

- une adresse = 32 bits = 4 octets
- un bloc = 4 Ko :peut contenir 4096 = 1024 adresses.

Donc

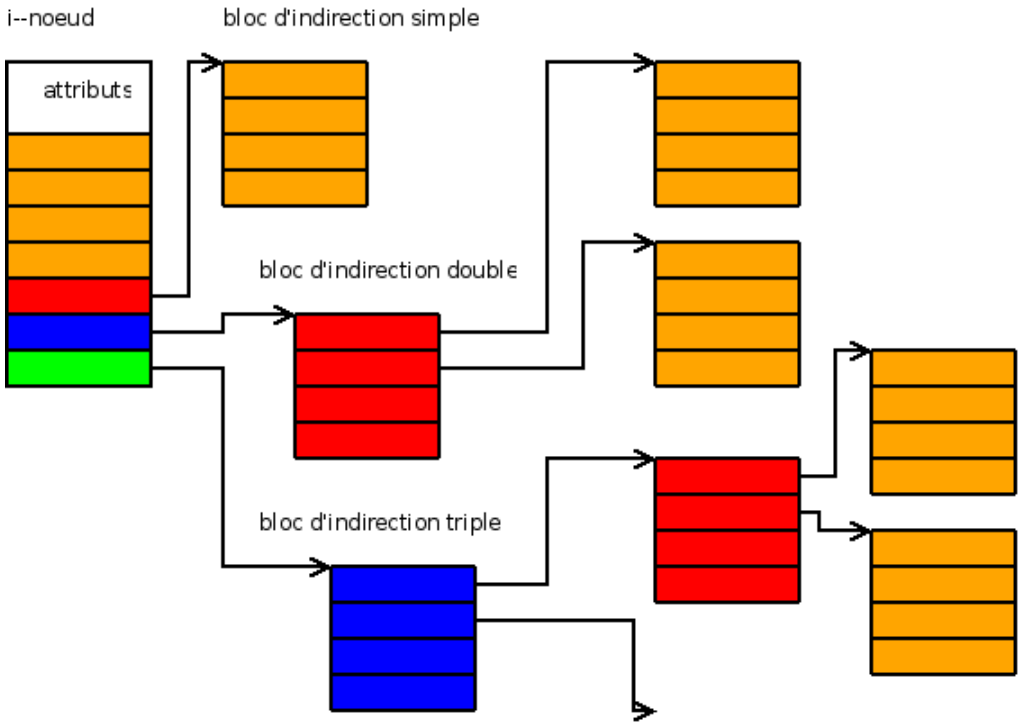
- un bloc d'indirection simple conduit à 1024 blocs de données soit 1024 $\times$ 4Ko soit 4 Mo de données

Taille maximum d'un fichier

- une adresse = 32 bits = 4 octets
- un bloc = 4 Ko :peut contenir 4096 = 1024 adresses.

Donc

- un bloc d'indirection simple conduit à 1024 blocs de données soit 1024 $\times$ 4Ko soit 4 Mo de données
- un BI doubles conduit à 1024 BI simple (4 Go)
- un BI triple conduit à 1024 BI double (4 To)



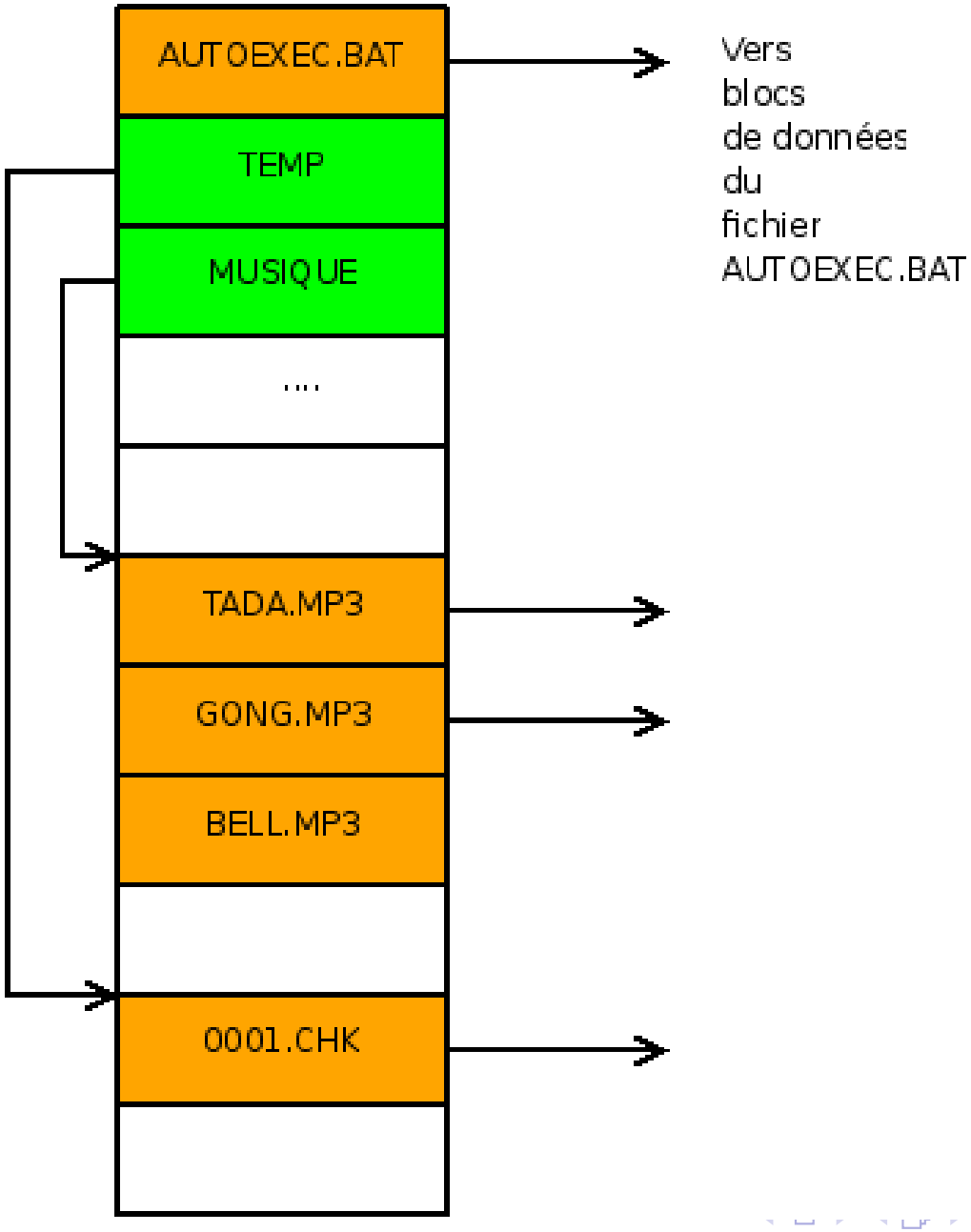
Pour la plupart des accès, une indirection suffit

Bilan

- Fichiers contigus :
  - temps d'accès : très bonne performances
  - problème de gestion des espaces libres
  - convient très bien à des supports en lecture seulement (CD, DVD)
- table des blocs, blocs chainés, i-nodes
  - gestion souple et efficace de l'espace
  - problèmes de performance si les données sont dispersées

Représenter les arborescences ?

Catalogue arborescent


AUTOEXEC.BAT
TEMP
MUSIQUE
....
TADA.MP3
GONG.MP3
BELL.MP3
0001.CHK
Vers
blocs
de données
du
fichier
AUTOEXEC.BAT

Catalogues de fichiers

- Répertoires matérialisés dans le catalogue par des lignes spéciales qui renvoient vers d'autres lignes
- ne permet pas d'avoir des liens, seulement des *raccourcis*
- Solution adoptée par CP/M, MS/DOS, Windows...

Types de lignes :

types de ligne	information
fichier	taille, blocs
vide	
répertoire	numéro de ligne
raccourci	chemin destination

Représentation des répertoires comme des fichiers de données

SGF Unix

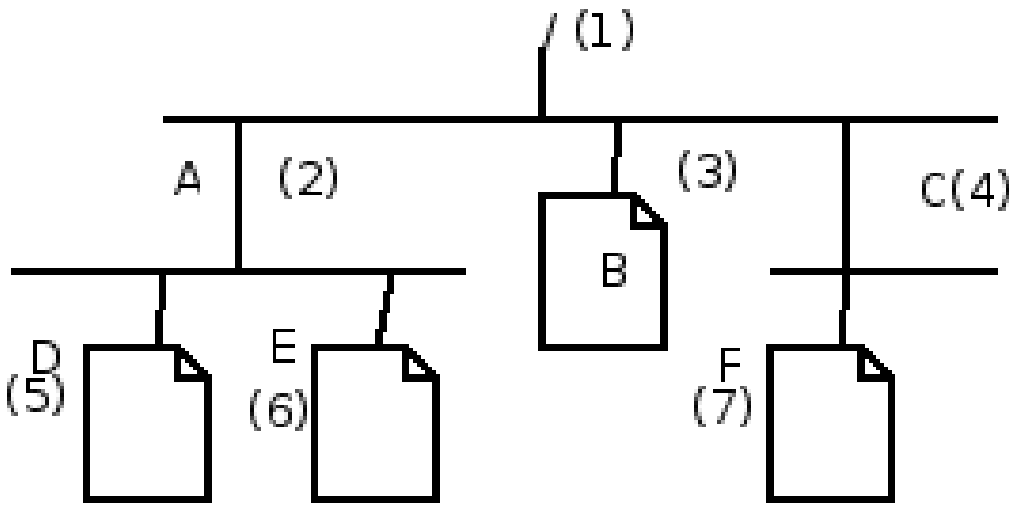
Un système de fichiers contient

- une table d'*i-nodes* (noeuds d'information)
- des blocs de données liés à ces i-nodes

Un fichier/répertoire... est identifié par son numéro d'i-node

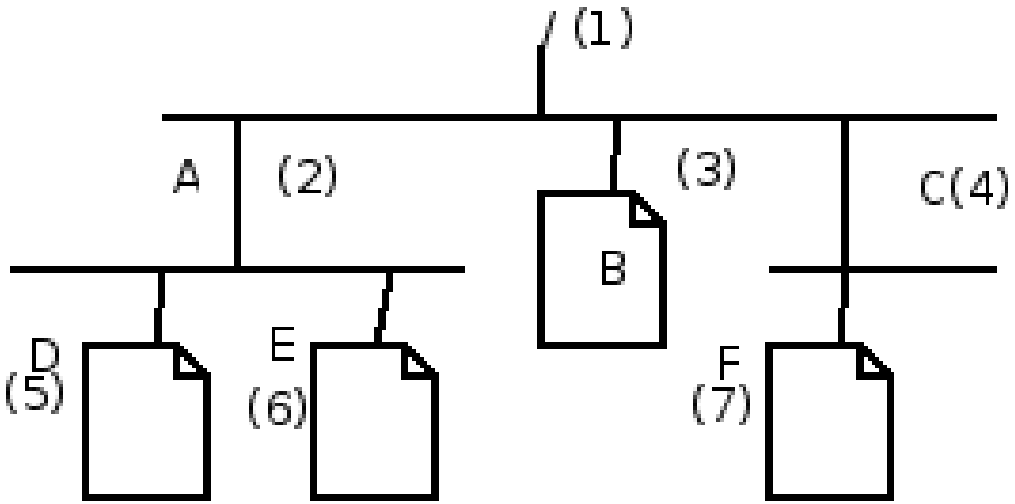
Différents types d'i-nodes

types	donnée
fichiers	Blocs = contenu du fichier
répertoires	Blocs = table de noms et numéros d'i-node
liens symboliques	chemin d'accès
périphériques	type, majeur, mineur
...	



Exemple d'arborescence

table des inodes



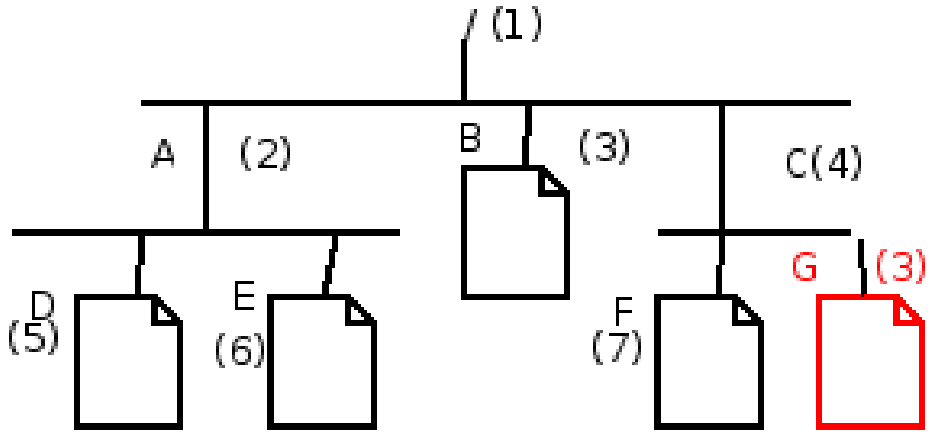
N°	type	CR	contenu des blocs
1	d	4	..=1, .=1, 1=2, B=3, C=4
2	d	2	..=1,.=2, D= 5, E=6
3	f	1	“coucou”
4	d	2	.. :1, .=4, F=7
		...	

CR = compteur de références

Compteur de référence

Dans un i-node, indique combien de fois l'objet est référencé
Quand le CR est à 0, on peut récupérer l'espace qu'il occupe.

table des inodes



Après `ln /B /C/G`

N°	type	CR	contenu des blocs
1	d	4	..=1, .=1, 1=2, B=3, C=4
2	d	2	..=1,.=2, D= 5, E=6
3	f	2	“coucou”
4	d	2	.. :1, .=4, F=7, G=3
		...	

Gestion des blocs libres ?

Blocs libres

Le système possède

- une liste des blocs libres
- un tableau de marquage des blocs occupés

Vérification du système de fichiers

Utilitaire `fsck`, descente de l'arborescence :

1. vérification des i-noeuds, des blocs et des tailles
2. vérification de la structure des répertoires
3. vérification de la connectivité des répertoires
4. vérification des compteurs de référence
5. vérification de l'information du sommaire de groupe

C'est long parfois

vérification des i-noeuds, des blocs et des tailles

- i-noeuds non-détruits $\Rightarrow$ blocs alloués.
- blocs alloués $\Rightarrow$ i-noeud.

vérification de la structure des répertoires

- numéro d'i-noeud cité dans un répertoire $\Rightarrow$ i-noeud existant

vérification de la connectivité des répertoires

- i-noeuds actifs $\Leftrightarrow$ accessibles depuis la racine

vérification des compteurs de référence

Nombre de références recalculé = nombre de références indiqué dans l'i-noeud

vérification de l'information du sommaire de groupe

Autres caractéristiques : la journalisation

Journalisation

Les systèmes de fichiers journalisés
Journal :

- garde une trace des opérations d'écriture non terminées
- permet de les reprendre en cas d'arrêt brutal

Avantages

- pas de pertes d'informations
- reprise sur incidents plus rapide (évite le *fsck*)

Snapshots (clichés)

Pendant la durée d'une sauvegarde,

- on ne veut pas que le système de fichiers soit modifié
- on ne veut pas arrêter l'exploitation

Cliché : copie de l'état du système de fichiers à un moment donné
On ne copie que *ce qui a changé* à partir du moment du cliché.