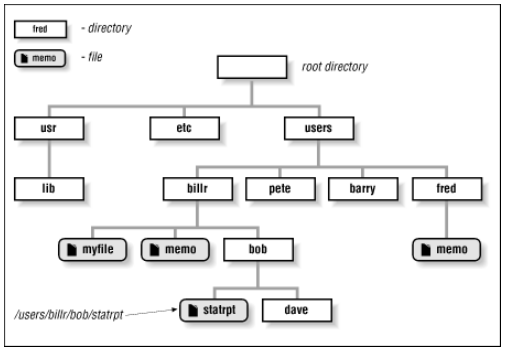
Gestion des disques : plan et illustrations

1 Définition

2 Pour l'utilisateur



pour l'utilisateur : arborescence

Un fichier a un contenu et des méta-données
- taille
- propriétaire
- droits d'accès
- date de création
- date de dernier accès
- ...

2.1 fonctions du SGF

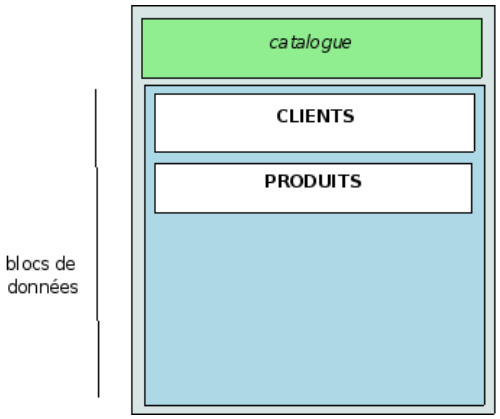
- Manipulation des fichiers : créer/détruire des fichiers, ...
- Allocation de la place sur mémoires secondaires
- Localisation des fichiers : accès au contenu
- Sécurité et contrôle des fichiers
- Fiabilité en cas de panne
- ...

3 Catalogue de fichiers

VTOC = Volume Table of Contents (IBM)
- pas de répertoires,
- fichiers contigus

Table des fichiers située au début du disque.

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



VTOC : occupation du disque

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

4 FAT

Table d'allocation. Fichiers non contigus : allocation plus facile à gérer.

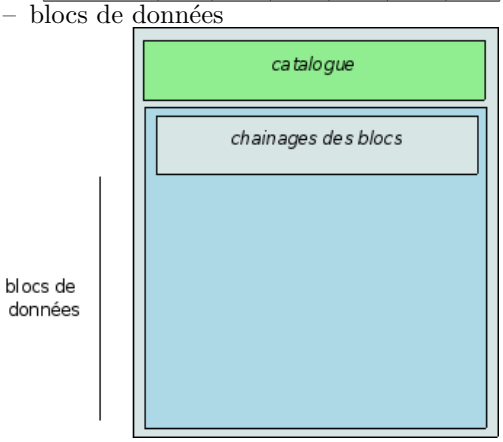
table supplémentaire : index du bloc suivant
- Table des fichiers

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

- et **table de chaînage** des blocs

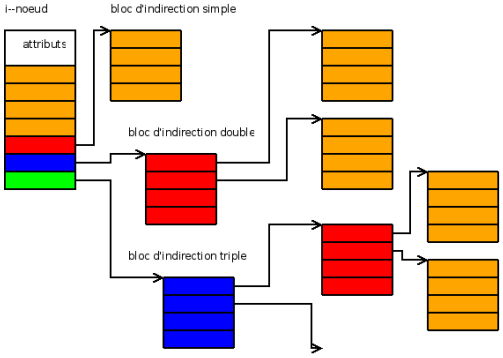
indice	...	10	11	12	13	...
suivant	...	11	20	13	14	...

– blocs de données



FAT : occupation du disque

– 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.



4.1 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

5 Tables des blocs 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.

I-nodes et blocs d'indirection

5.1 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

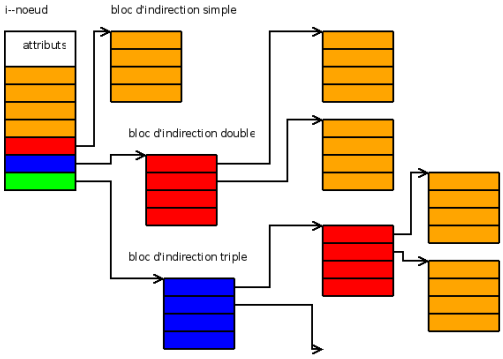
$$2^{32} \times 2^{12} = 2^{44}\text{octets} = 16\text{Tera octets}$$

Question : taille maximum d'un fichier ?

– 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 1024×4Ko 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

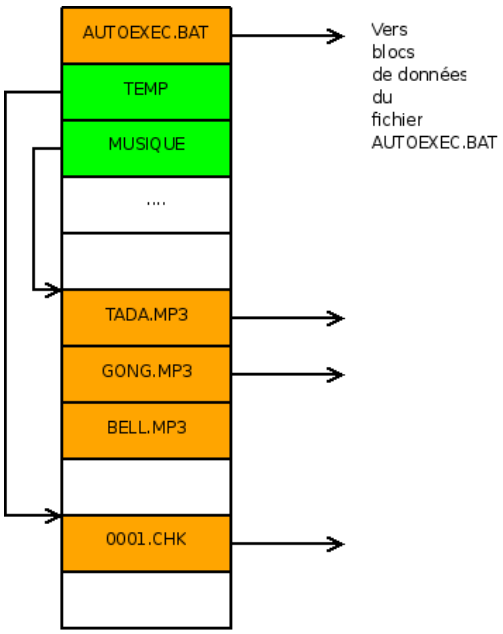
5.2 Bilan

- *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

6 Représentation des répertoires

Représenter les arborescence ?

6.1 Comme des catalogues de fichiers



entrées spéciales dans la table des 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

6.2 Comme des fichiers de données

SGF Unix

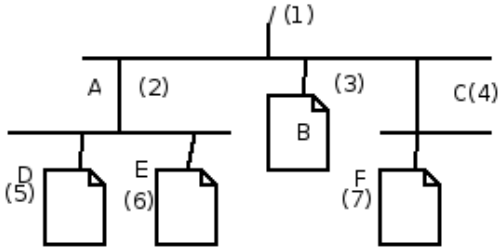
Un système de fichiers contient

- 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

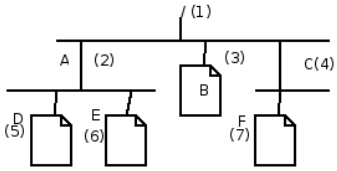
types	donnée
fichiers	Blocs = contenu du fichier
répertoires	Blocs = table de noms et numéros d'i-node
liens symboliques	chemin d'accès
périphériques	type, majeur, mineur
...	

6.3 Exemple



Exemple d'arborescence

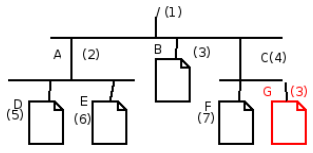
table des i-nodes



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. 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.

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**
		...	

6.4 Gestion des blocs libres

Le système possède
- une liste des **blocs libres**
- un **tableau de marquage des blocs occupés**

6.5 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

7 Autres caractéristiques

7.1 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*)

7.2 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.

8 Conclusion

Objectifs
- API pour accès aux fichiers/répertoires
- indépendance vis à vis du support
- fiabilité

Fonctionnalités
- arborescences
- droits d'accès
- accès concurrents, ...

Performances
- liées à l'implémentation
- liées au contexte d'usage