

1- Qu'est ce qu'un SGF :

Le système de gestion de fichiers (SGF) est la partie la plus visible d'un système d'exploitation qui se charge de gérer le stockage et la manipulation de fichiers (sur une unité de stockage : partition, disque, CD, disquette).

2- Les fonctions d'un SGF :

les fonctions attendus d'un SGF sont :

- 1- Indépendance vis-à vis du type de périphérique.
- 2- Création et suppression de fichiers.
- 3- Ouverture des fichiers avec gestion des accès concurrents.
- 4- Suppression des fichiers avec libération d'espace.
- 5- Lecture et écriture de données.
- 6- Extension automatique de la taille des fichiers.

3- Structure de fichiers :

Selon les systèmes, les données sont organisées dans les fichiers comme :

- Suites d'octets : Le SGF considère le fichier comme une suite d'octets sans aucune structure visible. C'est le programme utilisateur qui doit interpréter ces octets. Du point de vue SGF, c'est l'organisation la plus simple.

- Suites d'enregistrements : Le SGF considère le fichier comme une suite d'enregistrements de tailles fixes. Ces enregistrements ne peuvent être lus ou écrits qu'en totalité sans possibilité d'insertion au milieu.

- Arbres d'enregistrement : Le SGF considère le fichier comme une suite d'enregistrements de tailles variables, indexés chacun par une clé, et groupés par blocs de façon hiérarchique.

4-Stockage des fichiers sur le disque :

4.1- Organisation contiguë : Le fichier est constitué de plusieurs blocs contigus.

- Avantage :

- la rapidité d'accès (les blocs étant contigus, on limite les déplacements de la tête le lecture/écriture, coûteux en temps).

- Inconvénients :

- gaspillage de la place. Ainsi on risque d'avoir des petits espaces qui ne peuvent pas être utilisés.
- fragmentation externe : c'est l'espace perdu entre les fichiers. On peut effacer des données ou supprimer des fichiers ce qui libère des blocs sur le disque.

4.2- Organisation chaînée : Le fichier est constitué de plusieurs blocs chaînés. Ainsi, seul le numéro du premier bloc est stocké dans le répertoire. Une zone spécifique de chaque bloc contient le numéro du prochain bloc.

-Avantage :

- facilité d'extension de taille de fichiers.
- Facilité d'insertion au milieu d'un fichier.
- Absence de fragmentation externe.

-Inconvénients :

- L'accès au fichier est totalement séquentiel,
- La perte d'un chaînage entraîne la perte de tout le reste du fichier.

4.3- Organisation non contiguë indexée : les inconvénients de l'organisation chaînée peuvent être résolus d'une manière simple : les pointeurs des blocs sont dans une structure de données gardée en mémoire centrale, ainsi, les informations sur les numéros de blocs peuvent être obtenue à tout moment. Le système FAT est l'exemple le plus célèbre de ce type d'allocation.

On parle généralement de système de fichiers FAT16 et FAT32.

Le FAT16 est utilisé par MS-DOS. En FAT16, les numéros de blocs sont écrits sur 16 bits. Si on suppose que la taille d'un bloc est 32Ko, la taille maximale adressables est alors 2Go ($2^{16} \times 32 \text{ Ko} = 2097152 \text{ Ko} = 2\text{Go}$)

Le FAT32 est pris en charge par Windows 95 et les versions qui ont suivis. Les numéros de blocs sont écrits sur 32 bits (en réalité, sur 28bits, 4 bits étant réservés). Si on suppose que la taille d'un bloc est de 32 ko, la taille maximale adressable théoriquement est de 8 To ($2^{28} \times 32 \text{ Ko} = 8 \text{ To}$).

4.4- Organisation par I-noeud :

La structure d'I-Noeud est utilisée par le système de gestion de fichier ext3fs d'Unix ou GNU/Linux (ext3fs pour third extended file system). Un nœud d'index est constitué d'attributs décrivant le fichier ou le répertoire et d'adresses de blocs contenant des données. Cette structure possède plusieurs entrées, elle permet au système de disposer d'un certain nombre de données sur le fichier :

- la taille,
- l'identité du propriétaire et du groupe : un fichier en Unix est crée par un propriétaire, qui appartient à un groupe,
- Les droits d'accès : pour chaque fichier, Unix définit trois droits d'accès (lecture (r), écriture (w) et exécution (x)) pour chaque classe d'utilisateurs (trois types d'utilisateur {propriétaire, membre du même groupe que le propriétaire, autres}). Donc à chaque fichier, Unix associe neuf droits,
- les dates de création, de dernière consultation et de dernière modification,
- le nombre de références existant pour ce fichier dans le système,
- les dix premiers blocs de données,
- d'autres entrées contiennent l'adresse d'autres blocs (on parle alors de bloc d'indirection) :
 - une entrée pointe sur un bloc d'index qui contient 128 ou 256 pointeurs sur bloc de données (simple indirection)
 - Une entrée pointe sur un bloc d'index qui contient 128 ou 256 pointeurs sur bloc d'index dont chacun contient 128 ou 256 pointeurs sur bloc de données (double indirection).
 - Une entrée pointe sur un bloc d'index qui contient 128 ou 256 pointeurs sur bloc d'index dont chacun contient 128 ou 256 pointeurs sur bloc de données (triple indirection).

La structure d'I-Noeud est conçue afin d'alléger le répertoire et d'en éliminer les attributs du fichier ainsi que les informations sur l'emplacement des données.

Une entrée dans un I-Noeud d'un répertoire contiendra donc un nom d'un fichier ou sous-répertoire et l'I-Noeud associé.

5- Gestion de l'espace libre :

Les systèmes d'exploitation utilisent essentiellement deux approches pour mémoriser l'espace libre : une statique et une dynamique.

5.1- Bitmap : Approche statique qui utilise une table de bits (vecteur de n bits) comportant autant de bits que de blocs sur le disque. A chaque bloc du disque, correspond un bit dans la table, positionné à 1 si le bloc est occupé, à 0 si le bloc est libre (ou vice versa).

5.2- Liste chaînée : Approche dynamique utilise une liste chaînée constituée d'éléments, chacun mémorisant des numéros de blocs libres. Tous les blocs libres sont liés ensemble par des pointeurs.

6- Structure d'un disque dur :

Un disque dur est construit comme un empilement de disques magnétiques dont la surface est balayée par un ou plusieurs jeux de bras portant des têtes de lecture, Il en découle la terminologie suivante :

- piste : zone couverte par une tête de lecture en un tour de disque lorsque le bras reste dans une position donnée.
- cylindre : zone couverte sur tous les disques par l'ensemble des têtes de lecture en un tour de disque lorsque le bras reste dans une position donnée.
- Secteur : portion de disque représentant une fraction de la surface angulaire

Le temps d'accès à un bloc de disque est composé de :

temps de recherche "seek time" : temps nécessaire pour le déplacement du bras afin de positionner la tête de lecture/écriture sur une piste.

Temps de rotation "rotational delay" : temps d'attente pour que le bloc tourne sous la tête de lecture/écriture.

temps de transfert "transfer time" : temps de transfert des données vers/de la surface du disque.

Le temps de recherche est le plus important pour la plupart des disques ; sa réduction améliore sensiblement les performances du système.

7- Ordonnancement des requêtes d'un disque :

Il existe plusieurs algorithmes d'ordonnancement des requêtes du disque : FCFS, SSF, SCAN , ... etc.

7.1- Algorithme FCFS : Cet algorithme exécute les requêtes en fonction de leur ordre d'arrivée.

Considérons, par exemple, un disque de 40 cylindres. Une demande de lecture du cylindre 11 arrive. Pendant que la recherche est en cours, le pilote du disque reçoit respectivement de nouvelles requêtes qui concernent les cylindres 1, 36, 16, 34, 9 et 12. Elles sont placées dans la table des requêtes en attente qui contient la liste chaînée pour les différents cylindres. A la fin de la requête courante (concernant le cylindre 11), le pilote doit choisir la requête suivante, c'est à dire 1, puis 36, puis 16, et ainsi de suite.

Cet algorithme requiert que le bras se déplace de 10, 35, 20, 18, 25 et 3 cylindres, soit un total de 111 cylindres.

7.2- Algorithme SSF (Shortest Seek First) : Cet algorithme consiste à permettre au pilote de choisir la requête qui concerne le cylindre le plus proche de la position actuelle de manière à réduire le temps de recherche. L'ordre des cylindre pour l'exemple précédent est donc : 12, 9, 16, 1, 34, 36. Le nombre de cylindres parcourus est donc : $1+3+7+15+33+2=61$

7.3- Algorithme Scan (Ascenseur) :

La gestion d'un ascenseur dans un immeuble ressemble à celle du bras du disque. Les requêtes arrivent continuellement pour demander aléatoirement l'ascenseur aux différents étages (cylindres). Le processeur qui contrôle l'ascenseur mémorise facilement l'ordre des appels puis les traite selon la technique suivante : Il se déplace dans un certain sens tant qu'il y a des appels à traiter dans ce sens,

puis ils changent de sens. Le logiciel doit mémoriser un seul bit : le sens du déplacement, vers le bas ou vers le haut. A la fin d'une requête, le pilote du disque ou de l'ascenseur vérifie ce bit. S'il est à haut, le bras (ou la cabine) se place au niveau immédiatement plus élevé, indiqué par la première requête en attente. S'il n'y a pas de requêtes en attente, le bit de sens est inversé et la recherche se fait dans l'autre sens.

Le nombre de cylindres parcourus pour le même exemple est : $1+4+18+2+27+8=60$.