La gestion des fichiers

1. Notion de fichiers

D'un point de vue pratique, le fichier est le seul moyen de sauvegarder les informations à l'issue de la terminaison d'un processus ou tout simplement lors de l'arrêt de l'ordinateur. Le support de sauvegarde, à long terme, de ces informations est une mémoire secondaire (un disque, un cd-Rom,…) qu'on pourrait y accéder à tout moment sur simple demande. Les informations qu'on conserve, pour des besoins ultérieures, peuvent être des programmes (sources, binaires,…) et/ou des données (privées/partageables,…). De toute manière et quelque soit sa nature, un fichier peut être vu sous deux angles différents, selon qu'on se place du côté utilisateur (point de vue logique) ou du côté système d'exploitation (point de vue physique).

De point de vue logique, l'utilisateur décrit un fichier comme étant un ensemble d'articles de même nature et ayant une signification particulière vis-à-vis des traitements envisagés. De plus, l'utilisateur demande au système d'exploitation de fournir un type abstrait « fichier » avec des opérations telles que créer, détruire, dupliquer, renommer ou éditer un fichier ou encore lire, écrire, retrouver, insérer et supprimer un article dans un fichier.

De point de vue physique, la quantité de données qui peut être lue ou écrite en une seule opération sur un périphérique d'entrée/sortie correspond à une unité d'allocation (**bloc** ou enregistrement) physique. Sa taille est donc attachée au périphérique et est fixée par le matériel, et son contenu peut être décrit comme étant une suite d'octets. Puisque le système d'exploitation n'a pas à s'occuper du sens attribué au contenu des fichiers, on peut définir un fichier comme étant un ensemble d'unités d'allocations physiques allouées sur un support de mémoire secondaire.

Le système de gestion de fichiers (SGF) est la partie du système d'exploitation qui s'occupe de la gestion physique des fichiers. Son rôle est de rendre transparentes à l'utilisateur toutes les fonctions de manipulation du support physique en réalisant la correspondance entre l'organisation logique et l'organisation physique des fichiers.

Par extension, le SGF s'occupe également de la gestion des volumes qu'il faut installer sur les périphériques d'entrée/sortie appropriés. Cette gestion regroupe diverses opérations telles que le formatage ou l'initialisation des volumes, la conversion de support, l'archivage et la restauration du contenu d'un volume ou encore le montage d'un volume.

2. Organisation logique des fichiers

Les systèmes d'exploitation modernes adoptent une structure hiérarchique des fichiers. Chaque fichier appartient à un groupe d'autres fichiers et chaque groupe appartient lui-même à un groupe d'ordre supérieur. On appelle ces groupes, des répertoires ou des dossiers, suivant les terminologies.

Le schéma de la structure générale d'un système de fichiers prend l'aspect d'un arbre (voir Figure 1), formé au départ d'un répertoire « racine » recouvrant des périphériques et notamment un ou plusieurs disques. Dans chacun des répertoires on pourra trouver d'autres répertoires ainsi que des fichiers de données ordinaires.

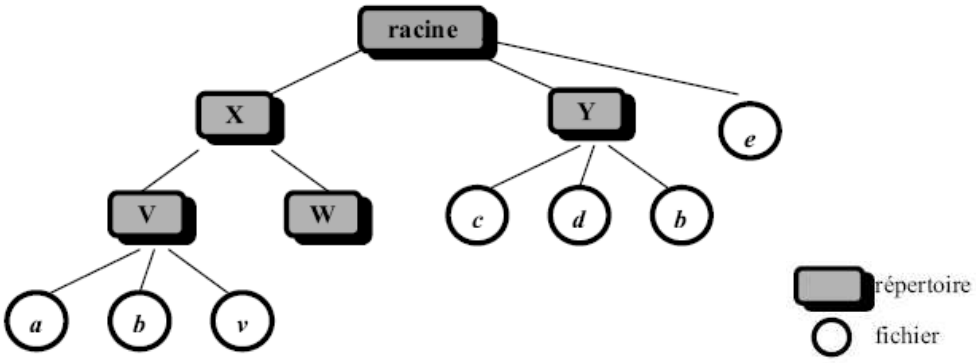

Figure 1 : Structure logique d'un système de fichier

Les répertoires sont, eux aussi, des fichiers, constitués des noms et des références de tous les fichiers qu'ils contiennent. Cette structure permet alors de construire l'arborescence du système.

Pour désigner un fichier quelconque, il suffit de spécifier l'enchaînement des répertoires nécessaires à son accès, à partir de la racine. Dans le système Unix, les répertoires de cet enchaînement sont séparés par une oblique : « / ». Dans le système DOS, par une contre-oblique : « \».

Dans le système Unix, chaque répertoire contient aussi sa propre référence, ainsi que celle du répertoire immédiatement supérieur. « . » désigne le répertoire courant, et « .. », le répertoire supérieur. L'inclusion de ces deux références permet de désigner un fichier quelconque, relativement au répertoire courant.

3. Organisation physique des fichiers

Les fichiers de données sur disques se composent d'un ensemble de blocs, comprenant un nombre fixes d'octets. La première structure possible de ces fichiers correspond à la suite des octets des blocs; ces blocs étant ordonnés. L'accès à un octet se fait alors par un déplacement à partir de l'origine du premier bloc : le début du fichier. Le système Unix, ainsi que le système DOS ne connaissent que cet accès.

3.1. Les blocs disque

Les fichiers de données sur les disques se répartissent dans des blocs de taille fixe correspondant à des unités d'entrées-sorties du contrôleur de ce périphérique. La lecture ou l'écriture d'un élément d'un fichier impliquera donc le transfert du bloc entier qui contient cet élément. On peut formater les disques, – les rendre utilisables par le système d'exploitation –, avec une taille particulière de blocs. Cette taille résulte d'un compromis entre la vitesse d'accès aux éléments des fichiers et l'espace perdu sur le disque.

Lors d'un transfert de données d'un disque vers l'espace d'adressage d'un processus, le temps de lecture ou d'écriture sur le disque est négligeable devant le temps d'accès au bloc, et ceci quelle que soit la taille du bloc. Pour un accès rapide, on aura donc intérêt à prendre des blocs de grande taille. Cependant, les fichiers, y compris les fichiers de 1 octet, ont une taille minimale de 1 bloc. Si un disque comprend beaucoup de fichiers de petite taille et si les blocs sont de grandes dimensions, l'espace gaspillé (fragmentation interne) sera alors considérable.

Des études sur de nombreux systèmes ont montré que la taille moyenne d'un fichier est de 1 Ko. Ce chiffre recouvre bien sûr de grandes variations entre, par exemple, une base de données et un ordinateur à usage pédagogique. Il conduit à des tailles courantes de blocs de 512, de 1024 ou de 2048 octets.

Chaque disque conserve, dans un ou plusieurs blocs spécifiques, un certain nombre d'informations de fonctionnement, telles par exemple que le nombre de ses blocs, leur taille, … Il mémorise aussi en général l'ensemble de ses blocs ainsi que leur état dans une table. Si cette table est binaire, un disque de n blocs devra alors réserver une table de n bits; la position de chaque bit indiquant si le bloc est libre ou s'il est utilisé par un fichier, par exemple, 0 pour un bloc occupé et 1 pour un bloc libre. Certains systèmes stockent l'ensemble des blocs libres dans une liste chaînée.

3.2. Répartition physique des fichiers en blocs

À chaque fichier correspond une liste de blocs contenant ses données. L'allocation est généralement non contiguë et les blocs sont donc répartis quasi-aléatoirement sur le disque. Les fichiers conservent l'ensemble de leurs blocs suivant deux méthodes : la liste chaînée et la table d'index.

3.2.1. La liste chaînée

L'ensemble des blocs d'un fichier peut être chaîné sous la forme d'une liste. Chaque bloc contiendra des données ainsi que l'adresse du bloc suivant. Le fichier doit mémoriser indépendamment le numéro du 1er bloc. Par exemple, si un bloc comporte 1024 octets et si le numéro d'un bloc se code sur 2 octets, 1022 octets seront réservés aux données et 2 octets au chaînage du bloc suivant (voir Figure 2).

Figure 2 : Représentation d'un fichier sous la forme d'une liste chainée de blocs

Cette méthode rend l'accès aléatoire aux éléments d'un fichier particulièrement inefficace lorsqu'elle est utilisée telle quelle. En effet pour atteindre un élément sur le bloc n d'un fichier, le système devra parcourir les n-1 blocs précédents.

Le système MS-DOS utilise des listes chaînées. Il conserve le premier bloc de chacun des fichiers dans son répertoire. Il optimise ensuite l'accès des blocs suivants en gardant leurs références dans une Table d'Allocation de Fichiers (FAT). Chaque disque dispose d'une FAT et cette dernière possède autant d'entrées qu'il y a de blocs sur le disque. Chaque entrée de FAT contient le numéro du bloc suivant.

Exemple :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	…
X	X	EOF	13	2	9	8	L	4	12	3	L	EOF	EOF	L	BE	…

où « XX » indique la taille du disque, « L » désigne un bloc libre et « BE » un bloc endommagé, le fichier commençant au bloc 6, sera constitué des blocs : 6 → 8 → 4 → 2.

3.2.2. La table d'index

Le système Unix répertorie chaque fichier par un numéro unique pour tout un disque. À chaque numéro, correspond un enregistrement, un nœud d'index, (i-node), comportant un nombre fixe de champs. Parmi ces champs, certains, au nombre de N, mémorisent l'emplacement physique du fichier. Les (N - 3) premiers champs contiennent les numéros des (N – 3) premiers blocs composant le fichier. Pour les fichiers de plus de (N – 3) blocs, on a recours à des indirections.

Dans le cas du système Unix (voir Figure 3), N=13. Les 10 premiers champs (numérotés de 0 à 9) contiennent les numéros des 10 premiers blocs composant le fichier. Le champ n° 10 contient le numéro d'un bloc composé lui-même d'adresses de blocs de données. Si les blocs ont une taille de 1024 octets et si ils sont numérotés sur 4 octets, le champ n° 10 pourra désigner jusqu'à 256 blocs. Au total, le fichier utilisant la simple indirection aura alors une taille maximale de 266 blocs.

De la même manière, le champ n° 11 contient une adresse, un pointeur, à double indirection, et le champ n° 12, un pointeur à triple indirection. Avec l'exemple que nous avons donné, un fichier peut avoir une taille maximale de 16 Go quand il utilise ces pointeurs à triple indirection.

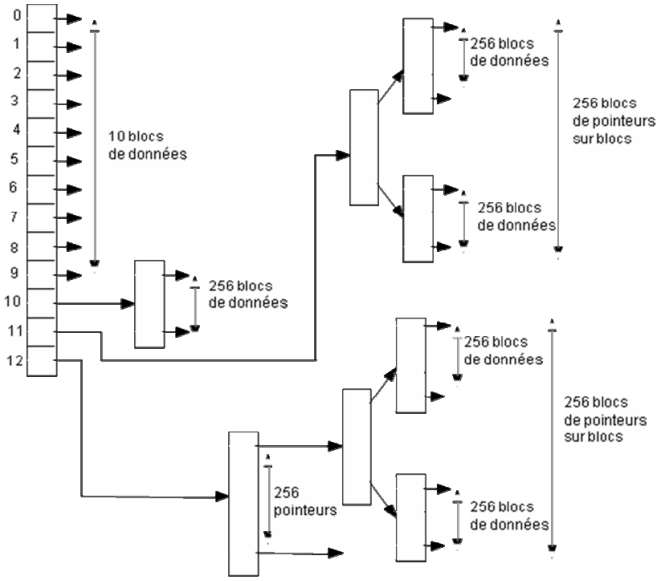

Figure 3 : Table d'index sous UNIX