

CHAPITRE 6 :

LA GESTION DE FICHIERS

Objectifs spécifiques

- ➔ Connaître la notion de fichier, ses caractéristiques
- ➔ Connaître la notion de répertoires et partitions
- ➔ Connaître les différentes stratégies d'allocation de blocs pour fichiers et de gestion d'espace disque.

Eléments de contenu

- I. Introduction à la gestion de fichier
- II. Les fichiers
- III. Répertoires, noms et partitions
- IV. Objets du système de fichier
- V. Fonctions du système de fichier
- VI. Architecture d'un système de fichier
- VII. Allocation

Volume Horaire :

Cours : 3 heures

TD : 1 heure 30

6.1 Introduction

Le volume des données traitées par les applications informatiques atteignant plusieurs méga et giga octets, ces données ne peuvent pas être stockées dans la mémoire centrale. On souhaite également disposer d'un stockage à long terme qui ne disparaisse pas lorsqu'on éteint la machine. Le principe consiste à stocker ces données dans des mémoires secondaires sous forme de fichiers, c'est-à-dire de suites de blocs (la plus petite unité que le périphérique de stockage est capable de gérer). Le contenu de ces blocs, simple suite de chiffres binaires, peut être interprété selon le format de fichier comme des caractères, des nombres entiers ou flottants, des codes d'opérations machines, des adresses mémoires, etc... L'échange entre les deux types de mémoires se fait ensuite par transfert de blocs.

L'objectif du système de fichier est de permettre l'accès au contenu du fichier (l'ouverture du fichier, sa copie à un second emplacement ou sa suppression) à partir de son chemin d'accès, formé d'un nom précédé d'une liste de répertoires imbriqués.

6.2 Les fichiers

6.2.1 Définition :

Un fichier est une collection logique d'information. Un système de fichiers est une collection de fichiers.

6.2.2 Le système de gestion de fichiers

Une des fonctions d'un SE est de masquer les spécificités des disques et des autres périphériques d'E/S et d'offrir au programmeur un modèle de manipulation des fichiers agréable et indépendant du matériel utilisé. Les appels système permettent de créer des fichiers, de les supprimer, de lire et d'écrire dans un fichier. Il faut également ouvrir un fichier avant de l'utiliser, le fermer ultérieurement. Les fichiers sont regroupés en répertoires arborescents; ils sont accessibles en énonçant leur chemin d'accès (chemin d'accès absolu à partir de la racine ou bien chemin d'accès relatif dans le cadre du répertoire de travail courant).

Le SE gère également la protection des fichiers.

6.3 Répertoires, noms de fichiers et partitions

6.3.1 Répertoire

Les systèmes de fichiers permettent aux utilisateurs d'organiser des fichiers et d'autres objets de systèmes de fichiers au moyen de répertoires. Un répertoire (dossier) est généralement défini comme un objet du système de fichier contenant d'autres objets de systèmes de fichiers. Les entrées des répertoires déterminent le chemin d'accès absolu ou nom associé à un objet du système de fichiers. En commençant par le répertoire racine, le chemin d'accès absolu est construit en concaténant la séquence des noms parcourus séparé par un « *backslash* » : \ sous windows ou Dos et un « *slash* » / sous linux ou Unix.

La plupart des systèmes prennent en charge la notion de répertoire courant ; au lieu de recourir à un chemin d'accès complet, s'avérant très long, il est possible de spécifier un chemin d'accès relatif.

6.3.2 Noms de fichiers

Les noms de fichiers représentent des mécanismes d'abstraction qui permettent d'écrire des données et de les trouver plus tard : chaque fichier est référencé par son nom. Le nom d'un fichier possède une longueur maximale qu'il ne peut pas dépasser relativement au système d'exploitation (Dos : 8, Win_XP : 255). Le nom est généralement composé de la partie nom et la partie extension séparés par un point.

6.3.3 Partitions

Les fichiers sont généralement stockés sur des unités de mémoire secondaire ; la mémoire vive peut néanmoins être utilisée pour stocker des fichiers, tels que les fichiers temporaires, pour lesquels un accès rapide est souhaité. La notion de partition peut également entrer en compte pour déterminer sur

quelle unité doit être stockée un fichier. Sur certains systèmes, comme Dos et Windows, la partition est spécifiée dans le chemin d'accès. Le nom `c:\rules\II2.txt` (sous Dos ou Windows) indique que l'objet du système de fichiers nommé `\rules\II2.txt` se trouve sur la partition `c :`.

6.4 Type des objets du système de fichiers

Dans de nombreux systèmes d'exploitation, les noms des systèmes de fichiers peuvent faire référence à des objets qui ne sont ni des fichiers ni des répertoires. Les objets qui peuvent se trouver dans un système de fichiers comprennent les éléments suivants :

→ **Un raccourci** : un raccourci est un pointeur vers un autre nom dans le système de fichiers. Dans la plupart des cas, le fait de se référer à un raccourci revient à se référer au nom pointé par le raccourci. La suppression du nom vers lequel pointe le raccourci laisse généralement le raccourci en suspens.

→ **Un périphérique** : un élément matériel, comme un port parallèle.

→ **Un tube** : un canal de communication entre 2 processus. Un processus envoie des données dans le tube l'autre processus lit ces données depuis le tube. Le tube met en mémoire tampon les données écrites à l'intérieur, jusqu'à ce qu'un autre processus lise les données. Le tampon est généralement de taille limitée et un processus qui écrit sur le tube doit être suspendu lorsque le tampon est plein. Les tubes peuvent être nommés ou non. Comme ils n'ont pas de nom les identifiant, les tubes sans nom ne sont généralement accessibles que par le processus les ayant créés ou les processus qui sont des descendant du processus.

→ **De la mémoire partagée** : une allocation d'emplacement de mémoire utilisable par un ou plusieurs processus. Comparable à un fichier sur un disque virtuel.

6.5 Fonctions des systèmes de fichiers

Le système de fichier doit offrir aux utilisateurs la possibilité d'accomplir des opérations abstraites sur les objets au sein du système de fichiers. Il doit, au minimum, proposer les fonctions suivantes : créations, suppression, lecture et écriture.

6.6 Architecture du système de fichiers

6.6.1 Structure de fichiers

Les fichiers peuvent être structurés de deux manières sous formes de suites d'octets non structurés ou d'une suite d'enregistrements. Un fichier est une séquence d'enregistrements de longueur fixe qui ont la même structure interne. Un fichier prend la forme d'un arbre d'enregistrements qui ne sont pas nécessairement de même longueur. Un enregistrement est une collection logique d'informations (par

exemple, une ligne de texte, des informations relatives à une personne). Les opérations d'E/S s'effectuent généralement en termes d'enregistrements. Le système d'exploitation peut gérer des structures d'enregistrements fixes et/ou variables. Les enregistrements peuvent à leur tour être divisés en champs, un champ représentant une donnée élémentaire (telle que le nom et l'âge). La figure suivante décrit la structure logique d'un fichier

	Chp 1	Chp 2	...	Chp m	
Enregistrement 1	Ali	Salah	1982	...	Brun
Enregistrement 2	Amal	Ali	1986	Roux
			.		
			.		
Enregistrement n	Basem	Faker	1910	Blond

Figure 7 : Structure logique d'un fichier

Transparent à l'utilisateur, le système d'exploitation considère un fichier comme une collection de blocs logique à taille fixe. Un bloc est l'unité de base d'une opération d'E/S entre le disque et la mémoire tampon du système de fichiers. Le disque en tant que tel est un ensemble de blocs physiques. Chacun d'entre eux stocke un bloc logique et éventuellement d'autres données administratives. La taille du bloc est un multiple de l'unité d'E/S de base fournie par le pilote du disque.

6.6.2 Méthodes d'accès

Il existe deux méthodes fondamentales pour accéder à des informations au sein d'un fichier : séquentielles et directes. Dans l'accès séquentiel, il faut accéder aux informations du fichier dans l'ordre dans lequel elles ont été stockées dans le fichier. L'accès se déroule de manière séquentielle depuis le début jusqu'à la fin. Les opérations de lecture ou d'écriture sur le fichier n'ont pas besoin de spécifier l'emplacement logique au sein du fichier, car le système d'exploitation maintient un pointeur de fichier qui détermine l'emplacement du prochain accès.

Avec l'accès direct, il est possible d'accéder à tout emplacement logique à l'intérieur du fichier. Généralement, l'accès direct peut être réalisé de 2 manières : Soit en spécifiant l'emplacement logique auquel accéder comme un paramètre à l'opération de lecture ou d'écriture, soit en spécifiant l'emplacement d'une opération de positionnement pour qu'il soit appelé avant la lecture ou l'écriture.

Les systèmes de base de données utilisent deux opérations d'accès au système d'exploitation élémentaire pour mettre en œuvre un large éventail de méthodes d'accès de haut niveau. Certains systèmes d'exploitation mettent en œuvre des méthodes d'accès de haut niveau par eux-mêmes. Parmi toutes ces méthodes, l'accès indexé est peut-être le plus significatif. Avec ce dernier, chaque enregistrement de fichiers dispose d'un ou plusieurs champs. Un champ sert de champ d'indexe. Les

opérations de lecture et d'écriture comprennent un paramètre d'index. L'enregistrement avec la valeur d'index correspondante est l'enregistrement sur lequel est effectuée l'opération.

Pour toute méthode d'accès, les opérations de lecture et d'écriture peuvent être synchrones ou asynchrones. Les blocs d'E/S synchrones bloquent le processus jusqu'à la fin de l'opération d'E/S. Les E/S asynchrones renvoient immédiatement le contrôle au processus, laissant le processus libre de continuer à s'exécuter pendant l'E/S. Si les opérations d'entrées sont asynchrones, il faut faire appel à certains mécanismes pour avertir le processus que l'opération est terminée. Pour cela, il est possible d'envoyer un signal au processus, d'attribuer une valeur particulière aux variables du processus ou de lancer un appel système spécial pour tester l'état de l'opération d'E/S. Les opérations de sortie asynchrones peuvent recourir aux mêmes techniques de notifications ou n'offrir aucune notification.

L'E/S synchrone représente la norme qui simplifie considérablement la programmation d'application. La grande vitesse des opérations d'E/S de fichier n'incite en rien à l'utilisation des E/S asynchrones. Cependant ces dernières peuvent parfois être employées avec un périphérique d'E/S.

6.6.3 Contrôle des droits d'accès

Le contrôle des droits d'accès établit une limite quant aux personnes pouvant accéder aux fichiers et à la manière dont elles peuvent y accéder. Le mécanisme de contrôle des droits d'accès le plus simple attribue un accès illimité à tous les utilisateurs. Il s'agit là du modèle de contrôle d'accès choisi par DOS. Sur un tel système, les utilisateurs qui souhaitent contrôler l'accès à leurs fichiers doivent mettre en place une limite d'accès physique (et sur le réseau) de leur machine.

Un aspect important du contrôle des droits d'accès est le type d'opérations à réaliser sur le fichier. Les opérations contrôlées comprennent entre autre :

- **La lecture** : lecture des informations contenues dans le fichier.
- **L'écriture** : écriture de nouvelles informations dans un fichier ou écrasement des informations d'un fichier.
- **L'adjonction** : écriture de nouvelles informations à la fin du fichier seulement.
- **La suppression** : suppression d'un fichier et libération de son espace de stockage en vue d'une utilisation dans d'autres fichiers.
- **La liste** : lecture des noms contenus dans un répertoire.
- **L'exécution** : chargement du contenu d'un fichier dans la mémoire principale et création d'un processus pour l'exécuter.
- **Le changement des droits d'accès** : modifications de certains droits d'accès d'utilisateur en vue d'une opération de contrôle.

L'autre caractéristique majeure du contrôle des droits d'accès est la manière dont il détermine ou non d'octroyer l'accès. Le mécanisme le plus commun consiste à baser la décision sur l'identité de l'utilisateur. Sur un système qui utilise une liste des droits d'accès, le système d'exploitation associe à

chaque fichier le type d'opérations autorisé à chaque utilisateur. Dans un modèle de droits d'accès illimité, un ensemble indépendant de permissions est conservé pour chaque utilisateur, ce qui représente un volume important de données.

Un mécanisme de contrôle des droits d'accès limité réduit ce volume en regroupant les permissions d'accès pour un certain nombre d'utilisateurs ou de fichiers. Ainsi, de nombreux systèmes d'exploitation mettent en œuvre la notion de groupes d'utilisateurs. Chaque utilisateur et chaque fichier sont associés à un ou plusieurs groupes d'utilisateur. Au lieu d'avoir un ensemble de permissions d'accès pour chaque utilisateur, le fichier possède uniquement un ensemble de permissions d'accès pour son propriétaire et pour chaque groupe auquel il est associé. Tous les utilisateurs d'un groupe partagent les mêmes permissions d'accès. Un autre groupement fréquemment utilisé consiste à demander à ce que tous les fichiers d'un répertoire partagent les mêmes permissions.

L'autre base communément utilisée pour contrôler l'accès est le mot de passe. Pour réaliser une opération sur un fichier, un utilisateur doit spécifier le mot de passe de fichier associé à cette opération (un mot de passe de fichier est distinct de tout éventuel mot de passe d'ouverture de session). Comme avec les listes de droits d'accès, le groupement peut réduire le volume des données maintenues par le système d'exploitation (et réduire le nombre de mots de passe que doit retenir un utilisateur). Ainsi, tous les fichiers d'un répertoire peuvent partager le même mot de passe.

Les capacités constituent une variante intéressante des listes de droits d'accès, mais cependant moins répandue. Sur les systèmes basés sur les capacités, les droits d'accès, au lieu d'être associés aux fichiers, sont associés aux processus. Lorsqu'un accès au fichier est tenté, le système d'exploitation vérifie le droit correspondant au fichier dans les droits d'accès associés au processus.

6.6.4 Verrouillage des fichiers

Le verrouillage des fichiers offre aux processus la possibilité de mettre en œuvre un accès exclusif à un fichier. Trois grandes options existent dans la mise en œuvre du verrouillage :

- Le verrouillage peut se limiter à l'ensemble des fichiers ou la mise en œuvre peut autoriser de verrouiller certaines parties d'un fichier.
- Le verrouillage peut s'appliquer à tout accès ou il peut exister différents niveaux. Sur certains systèmes, il y a à la fois des blocs de lecture et d'écriture. Un fichier verrouillé pour la lecture n'empêche pas un autre accès en lecture, mais l'accès en écriture par d'autre processus est refusé.
- Le verrouillage peut être soit obligatoire soit consultatif. Avec le verrouillage obligatoire, le système d'exploitation refuse l'accès pendant que le fichier est verrouillé. Avec le verrouillage consultatif, les primitives de verrouillage fournissent uniquement des informations sur l'état de verrouillage du fichier. L'accès n'est restreint que si un processus vérifie l'état de verrouillage du fichier et respecte le verrouillage qui est indiqué.

Dans certaines circonstances, un système d'exploitation peut mettre en œuvre un mécanisme de verrouillage implicite.

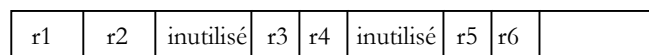
6.6.5 Attribution des blocs

La méthode d'attribution de blocs détermine comment les enregistrements d'un fichier sont attribués dans des blocs.

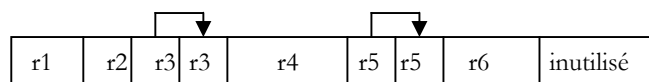
➔ **Attribution fixe des blocs** : pour les fichiers avec des enregistrements de taille fixe, un nombre intégral d'enregistrements est stocké dans chaque bloc. Aucun enregistrement ne peut être plus grand qu'un bloc. Si la taille du bloc n'est pas un multiple de la taille de l'enregistrement, il y aura de l'espace inoccupé à la fin du bloc. Le système d'exploitation peut calculer le bloc et l'offset à l'intérieur du bloc de tout enregistrement, en fonction de la taille de l'enregistrement et du bloc.

➔ **Attribution non étendue de blocs** : pour les systèmes avec des enregistrements de taille variable, plusieurs enregistrements peuvent être stockés dans chaque bloc, mais aucun ne peut s'étendre sur plusieurs blocs. Les enregistrements ne peuvent pas non plus être plus grands que la taille du bloc. L'espace à la fin d'un bloc est gaspillé si le prochain enregistrement est plus important que cet espace. Le système d'exploitation ne peut calculer l'emplacement d'un enregistrement, à moins qu'il ne connaisse la taille de tous les enregistrements qui le précède.

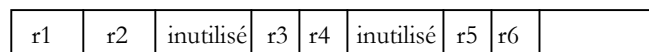
➔ **Attribution étendue de blocs** : les enregistrements peuvent être stockés dans plusieurs blocs. Il n'existe aucune limite quant à la taille d'un enregistrement et il n'y a pas d'espace inutilisé à l'intérieur d'un bloc. La seule manière de calculer l'emplacement d'un enregistrement d'un enregistrement consiste à additionner la taille de tous les enregistrements qui le précède.



Formation de blocs fixe



Formation de blocs étendue



Formation de blocs non étendue

6.7 Allocation

Il existe 3 modèles de base pour allouer l'espace de la mémoire auxiliaire à des fichiers. Le schéma d'allocation est chargé d'associer des blocs logiques d'un fichier à des blocs physiques de la mémoire

auxiliaire. Dans la plupart des systèmes d'exploitation, la taille d'un bloc physique est une puissance de 2 comprise entre 512 et 4096.

6.7.1 Allocation contiguë

Le modèle le plus simple est l'allocation contiguë. Les blocs logiques d'un fichier sont stockés dans une partition de blocs physiques contigus. L'entrée du répertoire a uniquement besoin de stocker l'adresse de la mémoire auxiliaire de départ du fichier ainsi que la taille de ce dernier. L'emplacement physique de tout octet du fichier peut être calculé en ajoutant l'offset approprié à l'adresse de la mémoire auxiliaire de départ de fichier.

Lorsqu'un fichier est créé, l'allocation contiguë requiert une pré-allocation d'espace pour le fichier. Le système d'exploitation peut être conçu pour étendre l'allocation du fichier, si nécessaire. Si l'expansion est autorisée et que l'espace de stockage au-dehors de la fin du fichier est utilisé, un ou plusieurs fichiers doivent être déplacés pour loger le fichier le plus important. Les fichiers à déplacer doivent recevoir de nouvelles partitions sur le disque, puis être copiés sur ces nouvelles partitions.

6.7.2 Allocation chaînée

Dans l'allocation chaînée, les blocs physiques dans lesquels est stocké un fichier peuvent être dispersés dans l'ensemble de la mémoire auxiliaire. Les blocs physiques sont plus importants que les blocs logiques et stockent à la fois le bloc logique et un pointeur vers le bloc physique dans lequel est stocké le prochain bloc logique du fichier. L'entrée du répertoire stocke l'emplacement du premier bloc physique. Le bloc physique associé au N^{ème} bloc logique peut être déterminé uniquement en lisant les précédents blocs N-1 et en suivant les liens qu'ils contiennent. Les performances des opérations d'adjonctions (écriture à la fin d'un fichier) peuvent être améliorées de façon significative en incluant également dans l'entrée du répertoire un pointeur vers le dernier bloc de la file.

6.7.3 Allocation indexée

L'allocation indexée est une variante de l'allocation chaînée. Le bloc physique stocke seulement le bloc logique, qui est par conséquent de la même taille qu'un bloc logique. Les liens vers les blocs physiques d'un fichier sont stockés de manière contiguë dans une table d'index. L'entrée du répertoire contient soit la table d'index soit un pointeur vers celle-ci.

Avec les allocations contiguës et chaînées, il suffit au système de fichiers de stocker l'emplacement physique de départ du fichier. Toutes les autres adresses peuvent être déterminées à partir de l'emplacement de départ du fichier. Avec l'allocation indexée, le système de fichiers doit avoir une entrée d'index pour chaque bloc du fichier. Pour minimiser le volume d'espace requis dans les structures de répertoire, l'indexation à plusieurs niveaux peut être utilisée.

6.7.4 Espace libre

Outre le fait de garder la trace de l'emplacement de chaque fichier dans la mémoire auxiliaire, un système d'exploitation doit aussi être capable d'identifier les blocs physiques qui ne sont alloués à aucun fichier. Un certain nombre de mécanismes ont été utilisés pour maintenir la liste d'espace libre.

Un bloc est soit utilisé soit non utilisé ; son statut peut donc être stocké dans un seul bit. L'utilisation d'une série de bits, un bit par bloc physique, crée un champ de bits d'espace libre. Les bits sont regroupés en aussi peu de mots que possible. Bien que ce mécanisme fasse un usage efficace de l'espace, l'accès aux informations sur un bloc particulier est ralenti par le traitement supplémentaire nécessaire à l'extraction des informations en bits individuels depuis un mot.

Un tableau d'espace libre stocke les adresses de tous les blocs libres. Un bloc est utilisé si son adresse n'est pas dans le tableau. Un tableau permet une implémentation efficace d'opérations d'ajout ou de suppression des blocs de la liste d'espace libre.

Cette liste peut également être maintenue en créant une liste chaînée de tous les blocs libres. L'avantage de ce schéma est l'efficacité de l'utilisation de l'espace ; les blocs libres stockent eux-mêmes des informations relatives à la liste d'espace libre.