Système d'Exploitation 2

Chapitre 3 : La gestion de la mémoire secondaire

(partie 3)

1 ère Année Second Cycle

Dr. M. Baba Ahmed

Les systèmes de fichiers (File System)

- Un système de gestion de fichier est la manière dont les fichiers sont nommés, organisés, et placés ou stockés
- Un SGF offre les primitives pour manipuler les fichiers/répertoires

Un SGF permet de :

- Définir les fonctions et les opérations sur les fichiers
- Définir et gérer les accès (droits d'accès)
- Gérer l'arborescence (l'organisation des fichiers/répertoires)
- Gérer et organiser les blocs dans le disque (cluster)

Les systèmes de fichiers (File System)

- Un SF utilisent des métadonnées pour stocker et récupérer des fichiers, comme (Date création/modification, dernier accès, ID utilisateur du créateur, taille …..)
- Les métadonnées sont stockées séparément du contenu du fichier

Les fichiers (Rappel)

- Fichier : ensemble d'octets (informations) enregistrées (stockées) dans sur un support de stockage secondaire
- Le bloc : est la plus petite unité de stockage d'un système de fichiers d'une mémoire de masse.
- Un enregistrement physique : correspond à la quantité de données écrites ou lues sur un support de stockage tel qu'un disque dur.
- Les enregistrements logiques : sont des enregistrements qui existent virtuellement, exemple : une table avec des enregistrements où une entrée est l'identifiant d'un employé (nom, prénom, salaire …)
- L'organisation logique d'un fichier décrit son contenu vu par les processus utilisateur.
- L'organisation physique d'un fichier décrit son implantation sur le support physique.

Les fichiers (Rappel)

Un fichier est caractérisé par :

Attributs (métadonnées) :

- **Nom**, **Identificateur:** Un nombre (unique) permettant au SE d'identifier le fichier, **Type**: Fichier simple, répertoire, **Position**, **Taille**, **Protection** (droits d'accès): Détermine qui peut écrire, lire, exécuter… **Dates**: Création, dernière modification, dernier accès
- Autres…

Type : Certains SE utilisent l'extension du nom du fichier pour identifier le type.

Microsoft : Un fichier exécutable doit avoir l'extension .EXE, .PPT, .IMG (sinon, le SE refusera de l'exécuter) (afin de le lier a l'application correspondante)

Linux : dépend du contenue du fichier plutôt que l'extension

Index-node (I-node)

Linux utilise index-node (I-node)

- 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 tel que (la taille, les droits d'accès, date de création….etc)
- La structure d'I-Node est conçue afin d'alléger le répertoire et d'en éliminer les attributs du fichier ainsi que les informations sur l'emplacement des données

Les fonctions/opérations sur les fichiers

Opérations sur les fichiers/répertoire :

- Création/suppression/renommage/copie
- Lecture/Écriture : pointeur de lecture / pointeur d'écriture qui donne la position d'écriture
- Ouverture : le fichier devient associé à un processus qui en garde les attributs, position, etc.
- Fermeture : Indiquer qu'on a finit d'opérer
- Lister un répertoire
- Recherche de fichier
- ….etc

Les droits d’accès

Mode d’accès au fichiers (r w x) :

- **r** : droit de lecture (read)
- **w** : droit d’écrire (write)
- **x** : droit d’exécuter (execute)

Autre :

- Effacement
- Listage: lister les noms et les attributs d ‘un fichier
- Changer les droits d’accès.

Les droits d’accès

Il existe trois catégories d'utilisateurs pour lesquels des droits d'accès peuvent être définis

- Propriétaire/ owner access (u)
- Groupe/ group access (g)
- Public/ others access (o)

L’administrateur: crée un nouveau groupe avec certain usagers et un certain propriétaire

Propriétaire: règle les droits d ’accès et ajoute des nouveaux usagers

Les droits d’accès

Modification des droits d’accès avec la commande ***chmod***

Donner le droit d’écrire pour les utilisateurs qui appartiennent au groupe (g+w)

#chmod g+w répertoire/fichier

Enlevé le droit de lire et exécuter aux autres utilisateurs (o-rx)

#chmod o-rx répertoire/fichier

exemple :

#chmod 744 file.txt

L’organisation des répertoires/fichiers

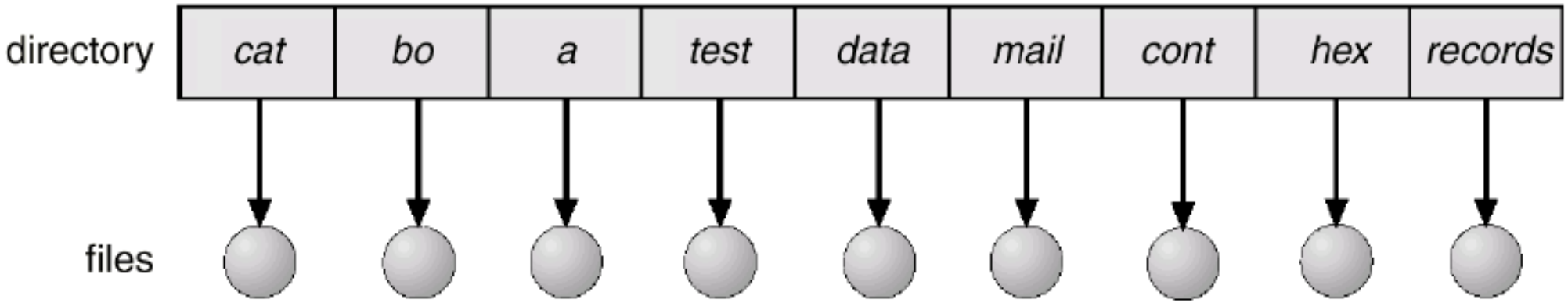
- Un répertoire (Directory) est une liste de descriptions de fichiers dont le contenu est la liste des fichiers référencés.
- Un répertoire a donc les mêmes types de propriétés qu'un fichier comme le nom, la taille, la date, les droits d'accès et les divers autres attributs.
- Ainsi des opérations tel que (Recherche, Création, suppression, listage, renommage ….etc)

Structure des répertoires

- Répertoire à un niveau
- Répertoire a deux niveaux
- Répertoire (arbres) hiérarchisé ou à plusieurs niveaux

Structure a un niveau

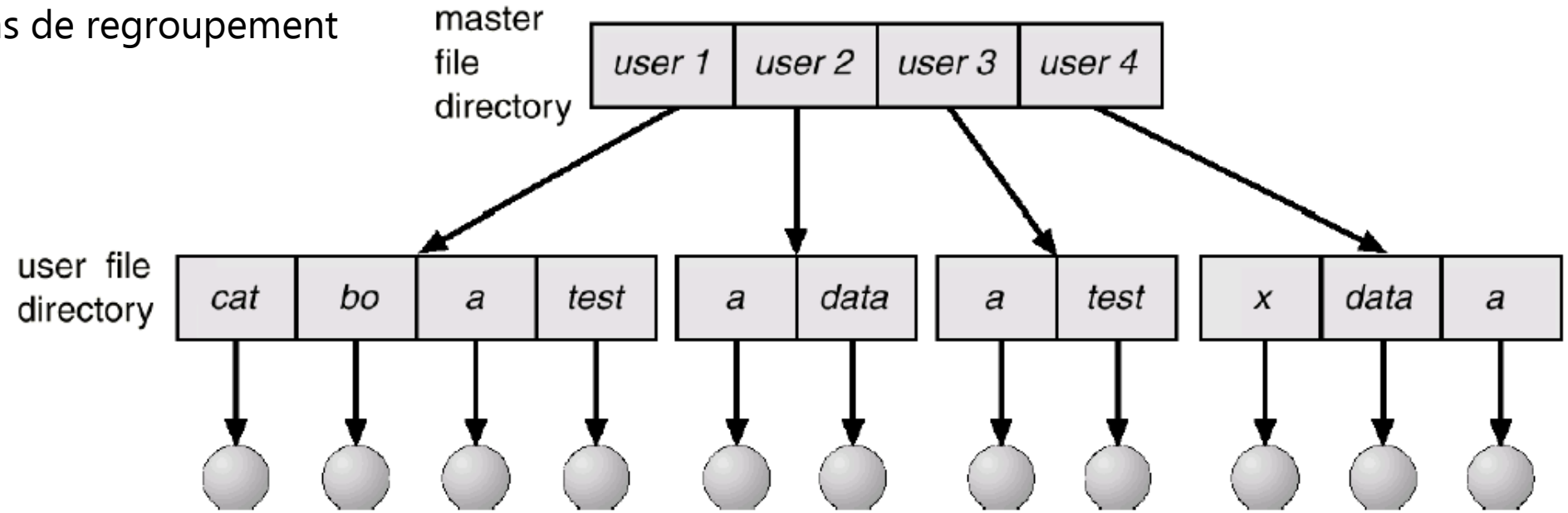
- Organisée en plusieurs répertoires mais chacun d'eux ne peut contenir que des fichiers.
- Les opérations telles que la création de fichiers, la recherche, la suppression, la mise à jour sont très simples dans une telle structure de répertoires.



- Un répertoire à un seul niveau présente toutefois une limitation importante lorsque le nombre de fichiers augmente ou lorsque le système compte plusieurs utilisateurs.
- Problème de regroupement

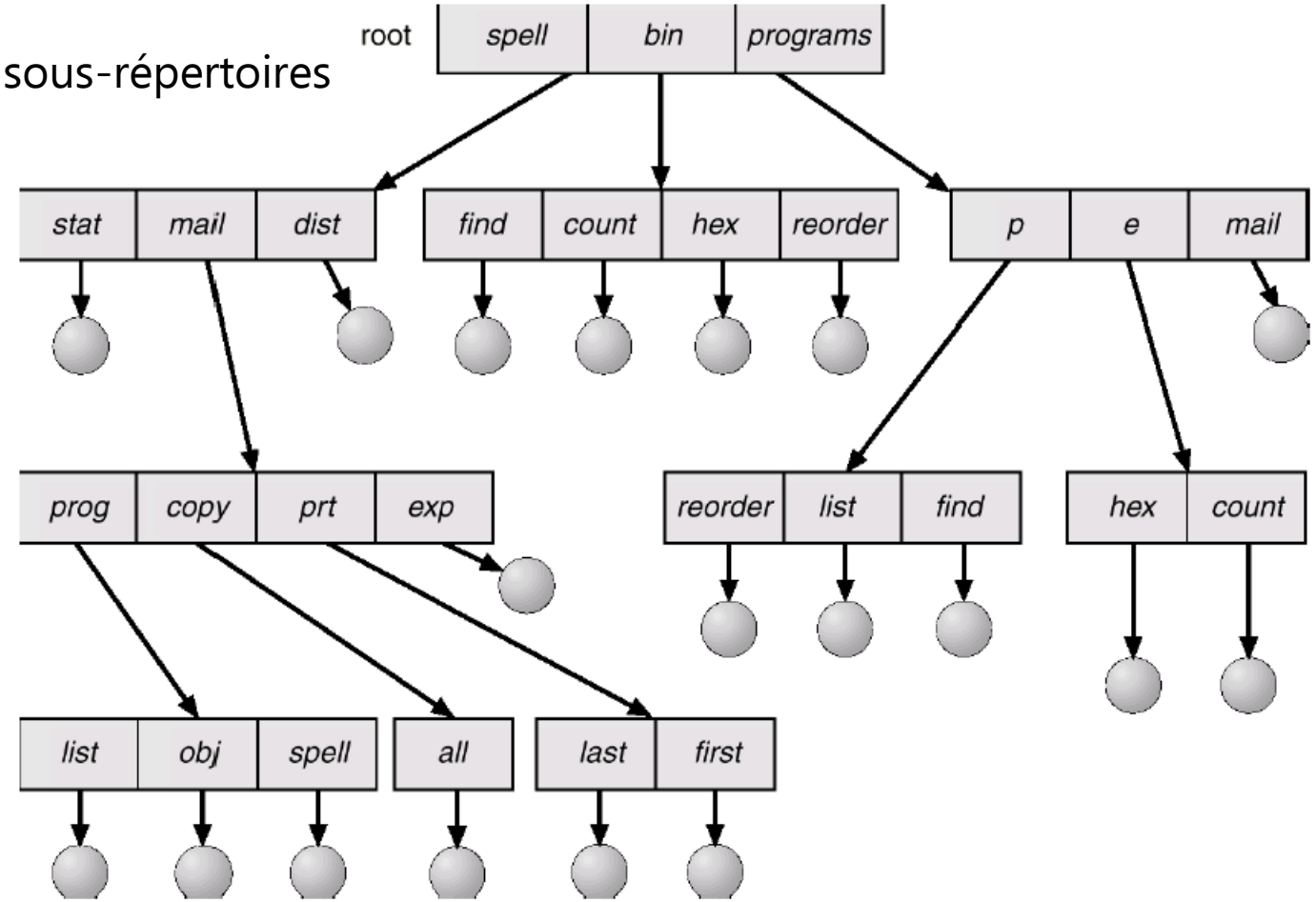
Structure a deux niveaux

- Création d'un répertoire séparé pour chaque utilisateur.
- Dans la structure de répertoires à deux niveaux, chaque utilisateur possède son propre répertoire qui répertorie que les fichiers de cet utilisateur.
- Pas de regroupement



Structure hiérarchisée (arbres)

- Permet à l'utilisateur de créer ses propres sous-répertoires et d'organiser ses fichiers en conséquence.
- Une arborescence est la structure de répertoire la plus courante.
- La recherche devient très facile
- Possibilité de regrouper
- L'accès à un fichier peut passer par plusieurs répertoires.



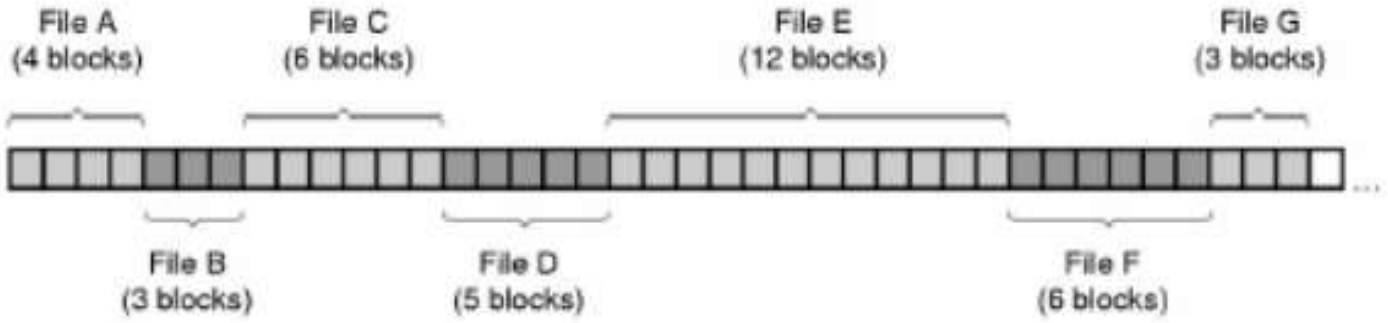
Allocation des blocs sur le disque

❑ On distingue trois manières d’organiser les blocs d’un fichier :

- Allocation contiguë
- Allocation enchaînée
- Allocation indexée

Allocation contiguë

- Chaque fichier occupe un ensemble de blocs contiguë sur disque
- Nous n'avons besoin que d'adresses de début et longueur
- Supporte tant l'accès séquentiel, que l'accès direct



Table

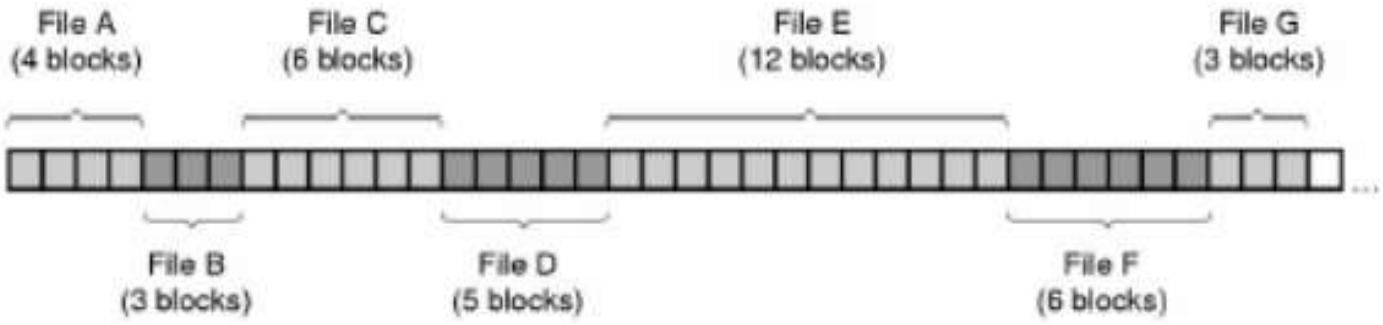
Fichier	Début	Fin
A	0	3
B	4	6
C	7	12
D	13	17
E	18	29
F	30	35
G	36	38

- Cette méthode présente l'avantage de la rapidité de l'accès (les blocs étant contigus, on limite les déplacements de la tête le lecture/écriture).

Allocation contiguë

Inconvénients :

- Problème de fragmentation externe : c'est l'espace perdu en dehors des fichiers. Lors de la suppression d'un fichier (ce qui libère des blocs sur le disque). Au fil de l'utilisation, il peut se créer un grand nombre de petites zones dont la taille ne suffit souvent pas pour allouer un fichier

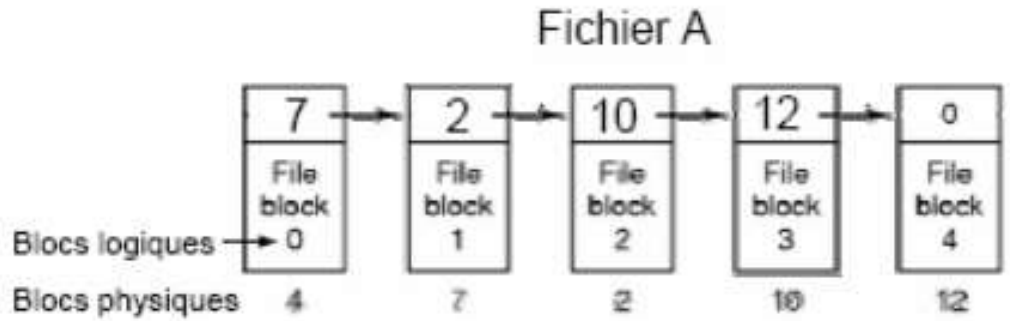


Table

Fichier	Début	Fin
A	0	3
B	4	6
C	7	12
D	13	17
E	18	29
F	30	35
G	36	38

Allocation chainée

- Un fichier peut être éparpillé sur le disque ou chaque bloc permet de retrouvé un autre bloc

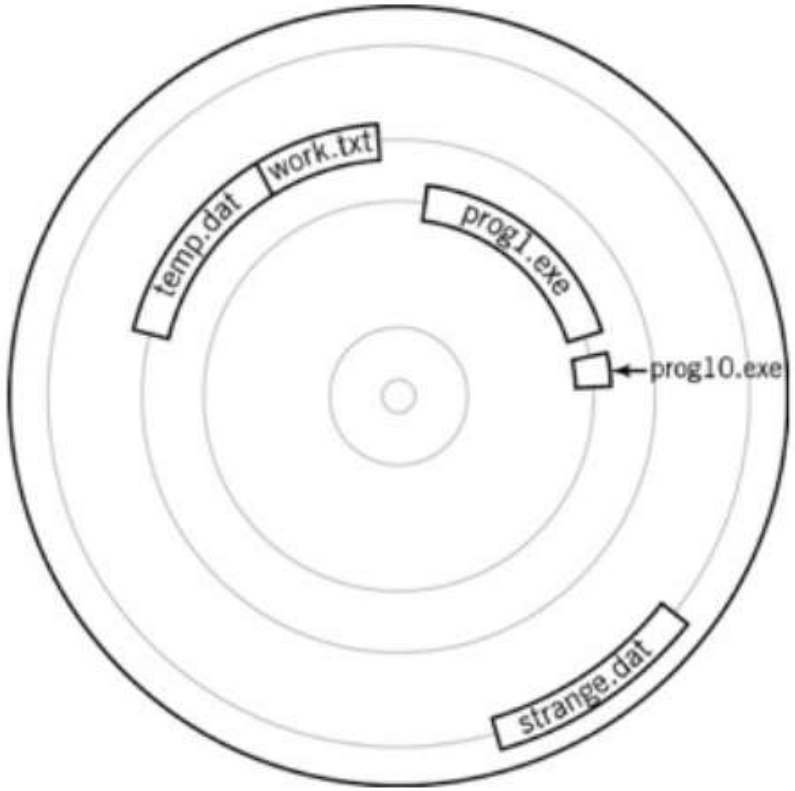


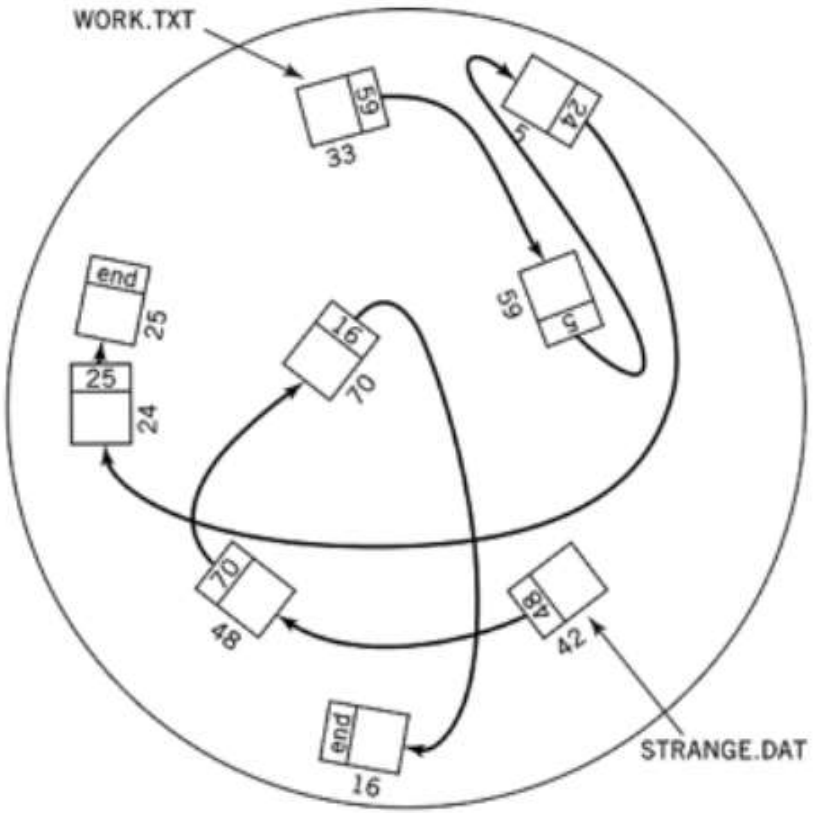
- Réduit le problème de fragmentation externe.

Inconvénients :

- L'accès au fichier est totalement séquentiel, on doit toujours commencer le parcours du fichier à partir du début.
- La perte d'un chaînage entraîne la perte de tout le reste du fichier.

Allocation contiguë/chainée





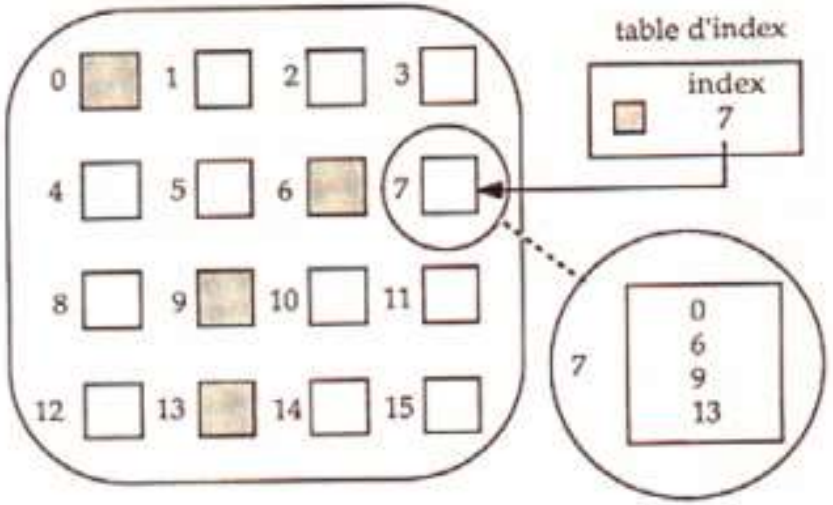
Allocation indexée

- Retirer les pointeurs des blocs et les placer dans une structure de données gardée en mémoire centrale
- Généralement pas de fragmentation externe
- Permet un accès direct

Inconvénients :

- Les index prennent de l'espace mémoire
- Chaque fichier doit posséder un bloc d'index dont il

est souhaitable qu'il soit le plus petit possible.



Cependant s'il est trop petit, il ne pourra pas ranger des pointeurs en nombre suffisant pour un grand fichier et il faudrait un mécanisme pour traiter ce détail, solution : avoir plusieurs niveaux d'index