
Chapitre 3 : Les Moyens Informatiques, Modes d'organisation et Méthodes d'accès

Partie I : Les moyens Informatiques

3.0. Introduction :

Nous avons vu au chapitre précédent que l'information permanente ou temporaire était structurée puis mise dans des fichiers. Ce sont les informations permanentes qui sont vitales pour les organisations.

Les fichiers de gestion étant volumineux sont stockés au niveau des mémoires secondaires. Pour effectuer des traitements sur ces données elles doivent être ramenées en mémoire centrale. La gestion de ces fichiers nécessite :

- Des techniques de stockage appelées organisations et méthodes d'accès aux fichiers.
- Des supports de stockage qui sont les mémoires secondaires.
- Un système de gestion permettant :
 - La définition logique du fichier, c'est à dire le descripteur de classe.
 - La définition physique du fichier, en termes d'organisation et d'accès.
 - La définition d'opérations d'E/S permettant le transfert des données des mémoires secondaires vers la mémoire centrale.

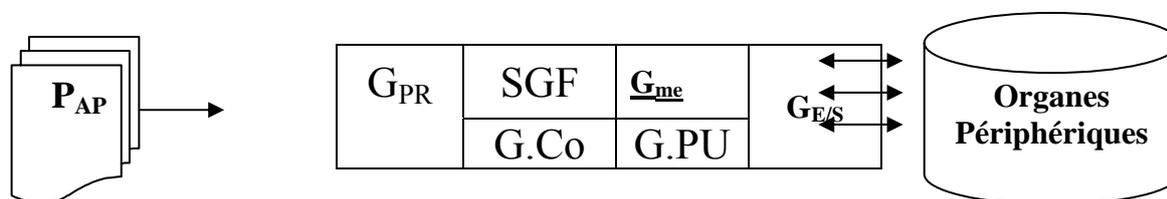
Ce chapitre traitera les points suivants :

- ❖ Les généralités sur les SGF.
- ❖ Les supports de fichiers.
- ❖ Les organisations et méthodes d'accès.

3.1. LES S.G.F (Système de Gestion de Fichiers) :

3.1.1. Généralités :

Un S.G.F est un ensemble de programmes résidant en M.C suivant le schéma suivant :

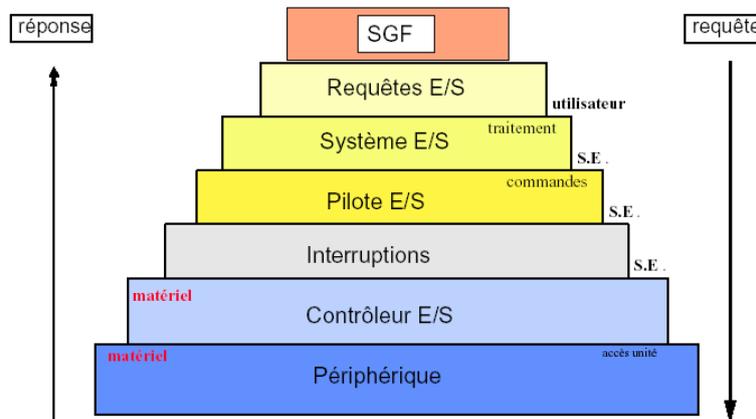


SITUATION DU S.G.F DANS LE SYSTEME OPERANT

- ❖ G_{PR} : gestionnaire des programmes usagers, il assure la transformation des programmes usagers en programmes exécutables.
- ❖ G_{me} : gestionnaire de la mémoire qui assure le partage de la mémoire entre les différents programmes usagers.
- ❖ G.PU : gestionnaire des processus qui permet l'exécution simultanée des programmes usagers et système.

- ❖ G.Co : gestionnaire des communications qui effectue le transport des données vers ou en provenance d'autres unités ou processus.
- ❖ G_{E/S} : gestionnaires des E/S, il assure l'échange des données entre les unités périphériques et l'unité centrale.
- ❖ S.G.F : il permet le stockage des données sur mémoire secondaires.

Le SGF fonctionne de la façon suivante dans le système d'exploitation de l'ordinateur :



3.1.2. Objectifs d'un S.G.F :

1 - Utilisation de mémoires secondaires :

L'informatisation de la gestion d'organisation nécessite le stockage de volumes des données importants. Pour cela, le stockage s'effectue sur des mémoires secondaires.

Définition : Mémoire secondaire

Ce sont des mémoires non directement adressables par les instructions du processeur central mais par des instructions d'E/S spécialisé et dont les temps d'accès sont très supérieurs à ceux de la mémoire centrale.

2 - Utilisation de langages hotes :

Les programmes usagers sont écrits en des langages évolués tel que COBOL, PL/1, CLIPPER,

Ces langages contiennent des verbes de manipulation de fichier du type READ, WRITE. Le S.G.F doit être utilisable par ce type des programmes.

On appelle langage **hote**, un langage de programmation accueillant les verbes de manipulation et de définition des données du fichier.

3 - Assure l'indépendance entre programmes d'application et mémoires secondaires :

Les mémoires secondaires sont partagées par plusieurs programmes d'application; ceux-ci ne gèrent par directement l'espace physique; cette gestion est assurée par le **S.G.F**. Autrement dit, un programme est indépendant de l'emplacement des données sur mémoire secondaires.

Exemple :

Un fichier client utilisé par un programme peut être stocké physiquement dans divers endroits physiques à des adresses différentes. Le programme utilise les données à travers leur nom et non à partir de leur adresse physique.

4 - offrir différentes possibilités d'accès :

Les possibilités d'accès aux fichiers dépendent des types de traitement que l'on effectue. Deux types sont remarquables :

- ▶ Traitement par lot (différé) : la totalité des articles sont utilisés \Rightarrow intérêt de l'accès séquentiel .
- ▶ Traitement en transactionnel (temps réel) : ce type de traitement affecte une partie des articles d'un fichier \Rightarrow intérêt de l'accès sélectif.

Ceci a conduit à l'introduction de deux types de méthodes d'accès **séquentiel** et **sélectif** .

5 - Accès Concurrents :

Deux programmes différents peuvent faire appel au même fichier ceci nous conduit à définir la simultanéité entre usagers .

Un type de simultanéité consiste à exécuter un programme d'application par le processeur central, pendant qu'un autre exécute des entrées – sorties. Les S.G.F. gèrent alors des files d'attente sur les processeurs.

6 - Confidentialité /Protection :

Lorsque des fichiers sont mis en commun, le propriétaire de ces fichiers désire en contrôler les accès pour :

- ▶ Les autoriser à certains.
- ▶ Les interdire à d'autres .

Les S.G.F assure cette protection à l'aide de mots de passe, de clé d'accès,

7 - Sécurité :

Le S.G.F doit garantir la conservation des fichiers en cas de panne de matériels ou de logiciels, il doit être capable de repartir après panne avec des fichiers correctes. On considère en général deux types de pannes.

- a) Les pannes simples avec perte du contenu de la Mémoire Centrale et les pannes nécessitent une reprise à chaud.
- b) Les pannes catastrophiques avec perte du contenu de mémoire secondaire. Ces pannes peuvent avoir des graves conséquences.

3.1.3. Fonctions du noyau du S.G.F :

Les fonctions de base du noyau du S.G.F sont :

A - Gestion des données :

- ▶ Création /destruction des fichiers.
- ▶ Ouverture / fermeture des fichiers.
- ▶ Manipulation des fichiers grâce à l'exploration de la table des matières des adresses où se trouvent les différents fichiers.
- ▶ Gestion des articles :

Le S.G.F exploite les données sur :

- Taille de l'article
 - Numéro d'article (accès séquentiel) .
 - Clé d'article (accès sélectif).
- ▶ Gestion des données :
- Stockage dans la zone physique allouée au fichier .
 - Recherche (localisation – accès – restitution) .

B - Allocation des zones en Mémoire secondaire :

Les S.G.F réserve des zones appelées "AREA" pour chaque fichier. Comme les fichiers sont vivants et de taille différente, les régions allouées à un fichier ne sont pas toujours contiguës. Le S.G.F doit pouvoir retrouver les « AREA ». Pour cela, il peut :

- ◆ Soit garder la liste des régions allouées à un fichier dans une table.
- ◆ Soit chaîner les régions allouées à un fichier.

C - Localisation des fichiers sur les volumes :

Un volume est une unité de mémoire secondaire amovible (Bande, Disk-pack, Disquette, CD-ROM, ...). Pour cela il est nécessaire de connaître :

- ☀ **Label de volume** : Premier secteur d'un volume permettant d'identifier ce volume et contenant en particulier son numéro.
- ☀ **Descripteur de fichier** : Identificateur de fichier ⇒ nom fichier, localisation .
- ☀ **Table des matières d'un volume** : table située sur un volume et contenant les descripteurs des fichiers du volume .

D – Contrôle des fichiers :

- ☀ Partage des fichiers.
- ☀ Sécurité et résistance aux pannes.

3.2. Les supports de fichier :

3.2.1. Définition :

- a) Un périphérique de stockage de l'information est un organe d'E/S relié directement à l'unité centrale d'un ordinateur.
 - Lecteur CD-ROM : (organe de sortie)
 - Graveur de CD-ROM (E/S)
 - Lecteur disquette (3 " ½)
 - Lecture Zip (disquette Zip de 100 à 250 Mo)
 - Disque Dur

Tous ces périphériques existant en modèles internes ou externes.

- b) Un support de fichier est un véhicule matériel de l'information. Par exemple, Disquette – CD-ROM –Cartouche de streamer- Disque Durs, etc
- c) Un fichier physique est un fichier déjà enregistré sur un support.

3.2.2 Différents types de supports :

Généralement, il existe 2 grandes classes de supports très utilisés :

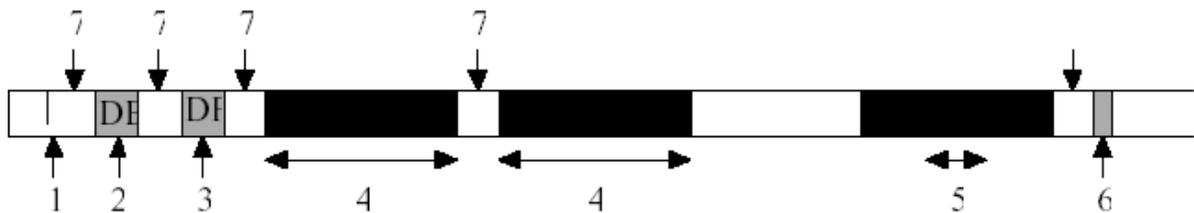
- Les bandes magnétiques et les disques.

Les bandes magnétiques font partie de la famille des supports non adressable. Par contre les disques font partie de la famille des supports adressables.

- a) Les Bandes magnétiques (BM) : on dit aussi « support séquentiel ». C'est un support sur lequel on ne peut lire ni écrire que l'info suivant celle que l'on vient de lire ou d'écrire.

- Les BM sont peu coûteuses et utilisées pour l'archivage. Nous pouvons aussi trouver des BM sous forme de cartouches (streamer).

Schéma d'une BM :



1 : Marque de début de bande ; 2 : Descripteur ou label de bande ; 3 : Descripteur ou label de fichier ; 4 : Bloc ; 5 : Enregistrement ; 6 : Marque de fin de fichier ; 7 : GAP.

- Une BM avoisine une longueur de 700 mètres et à une largeur de 1 cm, c'est un film en plastique ou nylon recouvert d'une couche sensible d'oxyde de fer ou de chrome capable d'enregistrer des données. L'épaisseur du film est d'environ 40 (micron).

Principe de fonctionnement :

La BM se défile devant une tête de Lecteur/Ecriture et devient aimantée. Par cet effet, l'info est transmise sous forme d'impulsion électrique. Le processus inverse permet de créer, de conserver et de modifier l'information.

1 inch(") = 1 pouce = 2.54 cm ex : 800 BPI, 1600 BPI, 6250 BPI,.....

- L'unité de transfert entre BM et Mémoire Centrale c'est le bloc

- 1 bloc peut contenir 1 ou plusieurs enregistrements.
- Pour écrire ou lire un bloc, la BM doit défiler à vitesse constante devant la tête de L/E du dérouleur.
- Les zones du GAP servent de zones d'accélération et de freinage.

Fonctionnalités d'une BM :

- Rembobiner : N° du dérouleur.
- Ecrire le bloc suivant : N° du dérouleur, @MC, Longueur. (E)
- Lire le bloc suivant : N° du dérouleur, @MC, Longueur. (S).
- Sauter des blocs : N° du dérouleur, 'avant' ou 'arrière', ' nombre de bloc '.

Disque Magnétique : on appelle support adressable tout support sur lequel chaque bloc d'information possède une @ et peut être lu ou écrit quelque soit l'accès précédent sur support. Ceci est similaire au fonctionnement de la mémoire centrale.

- L'info est ordonnée en pistes concentriques sur la surface du disque magnétique. Certains disques amovibles se présentent en piles de 6 à 11 disques superposés et l'ensemble de ces disques tournent solidairement au tour d'un axe commun.

- Les têtes de L/E peuvent être fixes ou mobiles. Dans ce dernier cas les accès sont plus lents car ils requièrent un mouvement mécanique, ce qui demande plus de temps pour accéder à l'information.

- Les disques son en rotation perpétuelle (à quelques rotations/seconde pour les disques).

- Le temps d'attente moyen pour que l'information passe sur la tête de L/E est de quelques 10 milli-sec.

- L'unité de transfert peut être constituée de 1 ou plusieurs secteurs logiquement contigus de la même piste.

Chapitre III : Les moyens informatiques, Modes d'organisation & Méthodes d'accès

Exemple :

- Pour un disque : 100 millions d'octets.
- Pour une disquette : = 100 00 ¢
- Pour un disque optique : 10⁹ ¢
- Pour une page d'un livre : 250 caractères.

- Les caractères sont enregistrés en série.
- L'unité de transfert peut être définie en fonction de 1 ou plusieurs secteurs ou d'une piste complète.
- Le nombre de secteurs constitue un segment.

Fonctionnalités :

- Ecrire : (N° disque, AdrMc, N° cycle, N° piste, N° secteur, nb secteurs)
- Lire : (N° disque, AdrMc, N° cycle, N° piste, N° secteur, nb secteurs)

Remarque Importante :

Il est possible de classer les mémoires secondaires en volumes fixe ou amovible. Les volumes amovibles par exemple sont les BM, Disquettes, CD-ROM. On les appelle ainsi car ils ne sont connus du SGF qu'au moment où on les place dans le lecteur correspondant sinon ils sont inaccessibles.

Pour que le SGF puisse identifier ces volumes amovibles, chaque volume doit contenir une zone spéciale appelée aussi « zone système » qui sert à stocker les informations décrivant le volume et son contenu.

Pour une disquette ou un disque dur, la piste n°1 est utilisée pour conserver les informations décrivant le volume et le contenu de celui-ci (disquette ou DD). On l'appelle aussi répertoire de la disquette ou du disque dur. Le répertoire contient les informations système suivantes :

- Le premier secteur contient le descripteur de la disquette ou du disque dur, i-e : Nom du volume + paramètres physiques du support.....
- Les secteurs suivants contiennent les descripteurs des fichiers stockés sur le support. Chaque descripteur de fichier contient essentiellement le nom du fichier, le nom du créateur du fichier, les droits d'accès au fichier, le type d'organisation du fichier, et la liste des pistes sur lesquelles le fichier se trouve.

En conclusion :

L'évolution constante des prix et des performances implique qu'il faut pouvoir changer les supports de fichier facilement tout en continuant l'utilisation des programmes manipulant le fichier.

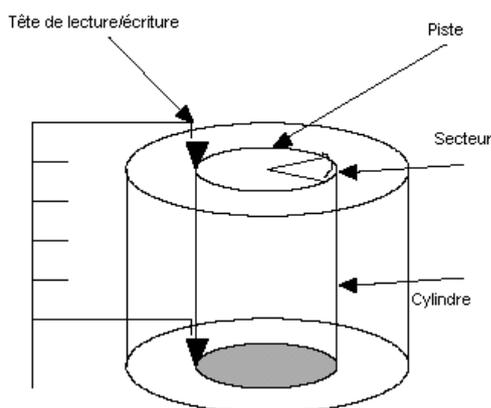
3.2.3. Un exemple de support magnétique : les disques durs et les disquettes

Les Disquettes existent en différentes dimensions et capacités dont la majorité a disparu aujourd'hui.

Capacité / Dimension	SD	DD	HD
3 " 1/2	360 K¢	720 K¢	1.44 M¢
5 " 1/4	180 ¢	360	1.2 M¢

Le Disque dur :

Leurs capacités évoluent de plus en plus : de 10 M¢ (Année 80) vers 80 G¢ et plus (en 2003/2004).



- Structure d'un support adressable : Disque Dur -

- Le DD est constitué d'un empilement de plateaux recouvert de matière magnétique. Les plateaux sont rigides, et sont accessibles en simple face ou en double face.
- Toutes les têtes de lecture sont montées sur 1 support.
- Lors de la lecture/Ecriture, une seule tête travaille à la fois, même si toutes les têtes se déplacent en même temps.
- L'ensemble des pistes de même numéro constituent un cylindre.

Fonctionnement :

- L'unité la plus petite de l'espace disque est le segment. Un segment est formé d'un petit nombre d'octets consécutifs. Par exemple on a des segments de 128 K ϕ , 512 K ϕ , ... etc.
- La surface du disque comporte un nombre entier de segments et le contrôleur de disque transfère un nombre entier de segments consécutifs à chaque accès.
- Un fichier est un ensemble d'enregistrements logiques. En général ces enregistrements sont regroupés en blocs que l'on appelle aussi enregistrements physiques.
- L'enregistrement physique est l'unité de transfert ou d'E/S.
- La taille du bloc dépend de la machine et le nombre d'enregistrement logique par blocs s'appelle facteur de blocage.

Exemple : soit la structure logique d'un fichier = 64 ϕ . On définit des blocs de 4096 ϕ

- Facteur de blocage = $4096/64 = 64$ enregistrements logiques / Bloc.
- D'après cet exemple numérique, lors d'un accès au disque on lira 1 bloc de 4096 ϕ , i-e, que 64 enregistrements logiques seront transférés à la fois vers la mémoire centrale.
- Si le segment = 512 ϕ alors $4096/512 = 8$ segments.
- La zone mémoire qui va recevoir ce bloc s'appelle zone tampon ou Buffer d'E/S.
- En général, la notion de bloc devient page au niveau du disque.

3.3. Etude de la typologie des fichiers :

3.3.1 Différentes Typologies :

A) Fichier permanent :

Ce sont des fichiers dont les occurrences de rubriques sont stables pendant le temps, et qui sont réutilisables à chaque traitement. On les appelle aussi fichier signalétique. Par exemple : le fichier des étudiants.

B) Fichier Archive :

Ce sont des fichiers qui permettent de sauvegarde des événements survenus durant une période déterminée. A chaque période va correspondre une version de ce fichier. Par exemple :

- fichier des facteurs de l'année 2000
- fichier des facteurs de l'année 2003
- Fichiers des notes étudiants pour 2001-2002
- Fichiers des notes étudiants pour 2002-2003

C) Fichier de situation et Historique :

Ce sont des fichiers permanents mais qui ont des propriétés de situation. La propriété de situation garde trace de la valeur courante de la propriété. Selon la nature du traitement, on peut utiliser uniquement le fichier de situation si seule la valeur en cours de la propriété nous intéresse. Sinon, on utilisera un autre fichier qui retracera les changements de la propriété de situation et ce fichier sera considéré comme un fichier historique pour la propriété de situation concernée. Dans ce cas, il faudra prévoir un fichier de situation et un fichier historique.

Par exemple : soit le fichier CLIENT_BANCAIRE {NumCompt, Nom, ..., Solde actuel} où Solde actuel est une propriété de situation. Si l'on veut garder trace de l'historique des soldes on utilisera un fichier historique HISTO_SOLDE {NumCompt, Date, Solde à cette date}.

Application numérique :

CLIENT_BANCAIRE {2467CLE50, SARL TOTO, 70 000,00 DA}

L'historique de la propriété Solde actuel est :

HISTO_SOLDE {2467CLE50, 18/10/2002, 150 000,00 DA}

HISTO_SOLDE {2467CLE50, 15/11/2002, 30 000,00 DA}

HISTO_SOLDE {2467CLE50, 20/12/2002, 50 000,00 DA}

HISTO_SOLDE {2467CLE50, 31/12/2002, 70 000,00 DA}

Lorsque l'on garde trace de la succession des valeurs que peut prendre une propriété de situation, on aura alors généré un fichier historique pour cette situation.

D) Fichier mouvement :

Ce sont des fichiers qui gardent trace des souvenirs d'un événement survenu à un instant donné. Il est constitué par les entités en relation de mouvement et ses informations sont conjoncturelles et non réutilisables à des instants autres que l'instant de leur arrivée. Par exemple :

- Enregistrer tous les mouvements (débit et crédit) d'un compte client pour une semaine donnée.
- Garder trace des mouvements de la paye pour un salarié donné (mois, année, net à payer, salarié).

Un fichier mouvement ne sera utilisé qu'à l'instant t de la modification. Il sert aussi à mettre à jour des fichiers de situation ou bien des fichiers permanents contenant des rubriques de situation.

Remarque importante :

Un fichier historique peut être vu comme un fichier de mouvement. L'inverse n'est pas toujours vrai. Les mouvements sont de deux natures :

Fichier mouvement de MAJ (mise à jour) :

Il s'agit de MAJ exceptionnelle de la valeur d'une ou plusieurs rubriques signalétiques ou de situation.

Fichier mouvement de données :

Ils servent à créer des fichiers permanents ou de situation. Ils servent aussi à ajouter de nouveaux articles dans un fichier permanent ou d'en supprimer d'autres. Par exemple :

Exemple :

Soit le fichier PRODUIT {CodeProd, Fournisseur, PU, QteStock} où Fournisseur et PU changent rarement et QteStock est une propriété de situation.

- Pour changer Fournisseur et PU, on utilisera un fichier mouvement de MAJ. De même, on utilisera un fichier mouvement de MAJ pour la QteStock dans lequel on gardera trace de toutes les entrées et sorties du stock.

- Pour calculer la nouvelle valeur de QteStock, on appliquera la formule :

$$\mathbf{QteStock = QteStock + \Sigma Entrées en stock - \Sigma Sorties du stock}$$

- Le fichier PRODUIT pourra être mis à jour à l'aide d'un fichier mouvement de données qui permettra d'ajouter les nouveaux produits et d'éliminer ceux qui n'existent plus. Cela signifie que l'on aura enregistré sur un fichier mouvement la liste des nouveaux produits et sur un autre fichier mouvement la liste des produits à éliminer.

Remarque :

A chaque type d'événement est associé un fichier mouvement. Après son utilisation, le fichier mouvement sera soit archivé soit détruit.

E) Fichier de travail :

Ce sont des fichiers brouillons dans lesquels on enregistre des résultats intermédiaires. Leur existence est liée à la durée qui sépare deux étapes d'un traitement et ils seront détruits à la fin du traitement. Ce type de fichier sert généralement à faire des calculs intermédiaires.

F) Fichier d'attente :

Ce sont des fichiers contenant des articles d'un même fichier et qui attendent de subir un traitement particulier.

Partie II : Modes d'organisation & Méthodes d'accès

3.3. Introduction :

Un **mode d'organisation** est une **méthode** de rangement, d'implémentation qui permet d'attribuer un emplacement physique sur le support (Disque dur, disquette, ...) à tout article d'un fichier.

Une **méthode d'accès** sur un fichier est la **manière** ou la façon de retrouver l'emplacement d'un article sur le support.

Les méthodes d'accès aux articles d'un fichier sont liées au mode d'organisation de ce fichier. On distingue deux grands modes d'organisation :

1- mode d'organisation séquentielle :

- ❖ Consécutive (accès séquentiel).
- ❖ Distribuée ou chaînée (accès séquentiel).

2 - mode d'organisation sélective :

- ❖ Relative (accès sélectif)
- ❖ Aléatoire (accès sélectif)
- ❖ indexée (séquentiel indexé)

Les Facteurs de choix d'un mode d'organisation dépendent de l'estimation du coût de traitement des fichiers qui fait intervenir :

- ❖ Le temps d'exécution d'une opération.
- ❖ Le temps de transfert des données des mémoires secondaires (MS) vers la mémoire centrale (MC).

Les temps de transfert vont dépendre du choix de l'organisation. Un certain nombre de facteurs vont guider le concepteur dans son choix entre autre :

- ❖ La taille du fichier.
- ❖ La croissance du fichier.
- ❖ Le type d'activité sur le fichier.
- ❖ ...

3.3.1 – Les organisations séquentielles :

Définition :

On mémorise les articles sur le support dans l'ordre où ils se présentent, c'est à dire, on met les enregistrements dans le fichier les uns après les autres selon leur ordre de création ou d'arrivée sans se soucier de l'endroit où ils ont été écrits.

Dans cette organisation, seul l'accès séquentiel est possible. Ce genre d'organisation est possible sur tout type de supports (adressable ou non). Cependant, elle fut destinée à l'origine pour les supports de type « bandes magnétiques ». Il existe 2 types d'implantation d'un fichier à organisation séquentielle :

Implantation consécutive :

Les articles sont implantés de manière consécutive. Cette méthode est utilisée surtout sur les bandes magnétiques.

Implantation distribuée :

Les articles sont distribués sur un ensemble de régions allouées au fichier. L'ordre séquentiel est retrouvé grâce à un chaînage.

Parmi les méthodes basées sur cette approche, nous avons :

A - Organisation séquentielle consécutive :

Cette méthode est valable aussi bien sur support séquentiel (BM) qu'adressable (DD, Disquette, ...). Les adresses sont consécutives. Dans ce qui suit, nous donnons une description des différentes opérations effectuées par le SGF dans une telle organisation.

A.1 - Création :

Les articles peuvent être de longueur fixe ou variable, bloqués ou non (i-e, organisés en blocs ou pas), mais ils doivent être toujours triés à l'entrée pour faciliter la recherche. D'où la nécessité d'exécuter l'opération de tri des fichiers mouvements de données pour la création des fichiers permanents.

A.2 - Modification :

Type de support	Procédé
Support séquentiel (BM)	Obligation de recopier sur une autre bande (puisque le support est non adressable)
Support adressable (DD, Floppy, ...)	L'article est modifié puis réécrit à la même place.

A.3 - Insertion / suppression :

Type de support	Procédé
Support séquentiel (BM)	Création d'un autre fichier en insérant (ou en supprimant) l'article concerné au bon endroit : le fichier restera trié et sans trous.
Support adressable (DD, Floppy, ...)	Création d'un autre fichier de la même manière.

A.4 - Améliorations :

Certains S.G.F évitent la recopie totale du fichier en :

- Allouant un espace physique initialement plus grand que l'espace nécessaire.
- Les espaces libres peuvent être répartis à l'intérieur de l'espace physique initial.
- Les suppressions se font logiquement et non physiquement par l'ajout d'un caractère spécial (ou marquage de l'article) qui est inséré par le S.G.F à l'intérieur de chaque article pour indiquer sa validité.

Une opération de compactage permettra de libérer réellement la place occupée par les articles supprimés logiquement (on parle de récupération de trous).

A.5 - Méthodes d'accès :

- ❖ Recherche pour balayage systématique.
- ❖ Recherche dichotomique.

En Conclusion :

- ✚ Cette Méthode a l'avantage d'être simple et la densité d'occupation du support est importante.
- ✚ Elle présente l'inconvénient d'être rigide et les opérations de MAJ sont longues à exécuter.

B – Organisation Séquentielle distribuée :

Cette méthode est valable uniquement sur support adressable. Dans ce qui suit, nous donnons une description des différentes opérations effectuées par le SGF dans une telle organisation.

B.1 - Création :

A sa création, le fichier est constitué de trois zones :

- ✓ ZP : zone principal où les articles sont enregistrés à la création (au chargement).
- ✓ ZS : zone secondaire où les articles seront enregistrés ultérieurement.
- ✓ ZD : zone de disponibilité destinée à recevoir la table de disponibilité chargée de tenir la comptabilité des emplacements libre en ZP.

Structure :



Les articles peuvent être :

- ✓ Chaîné avant.
- ✓ Chaîné arrière.
- ✓ Ou les deux à la fois.

Ils peuvent être aussi :

- ✓ Triés suivant l'ordre croissant ou décroissant de l'indicatif principal.

B.2 - Consultation :

La consultation se fait séquentiellement par lecture du pointeur.

B.3 - Insertion :

- ❖ L'article à insérer est enregistré à la 1^{ère} place libre indiquée par la ZD.
- ❖ Le chaînage et mis à jour.
- ❖ La table des disponibilités est M.A.J.

B.4 - Suppression :

- ❖ Elle s'effectue par marquage de l'article à supprimer.
- ❖ Mise à jour des pointeurs de chaînage.
- ❖ Mise à jour de la ZD.

B.5 - Mécanismes de l'Insertion et de la suppression :

Lors de l'une de ces opérations, il est nécessaire de ramener en mémoire centrale un certain nombre d'articles.

Pour la Suppression :

- ❖ L'article à supprimer
- ❖ Le précédent
- ❖ le suivant
- ❖ Et la table des disponibilités.

Pour l'Insertion :

- ❖ Article précédent pour M.A.J du pointeur avant.
- ❖ Article suivant pour M.A.J pointeur arrière.
- ❖ Et la table des disponibilités.

B.6 - Modification :

Dans le cas d'enregistrements de tailles variables (le problème ne se pose pas pour des enregistrements de longueur fixe), la taille de l'article de M.A.J doit être inférieure à la taille de l'article avant la M.A.J. Dans ce cas, on procède à l'écriture de l'article à la même place. Sinon, on procède comme suit :

- ❖ Suppression de l'article Avant la M.A.J.
- ❖ Insertion de l'article de M.A.J et restauration des chaînages.

En Conclusion :

- ✓ Les insertions et les suppressions ne nécessitent pas la réécriture complète du fichier.
- ✓ Cette technique est avantageuse pour de fréquentes insertions ou mises à jour.

3.3.2 - Les organisations sélectives :

A – Organisation à accès direct :

Un fichier à accès direct peut être vu comme une collection d'articles numérotés qui possède les propriétés suivantes :

1. On peut lire tout enregistrement dans la même durée de temps quelque soit son numéro.
2. La consultation et l'insertion se font en spécifiant le numéro d'enregistrement.
3. Si l'on essaie d'insérer un enregistrement sous un numéro qui existe déjà, l'ancien enregistrement sera écrasé.

Il existe un inconvénient majeur avec de tels fichiers car on manipule des adresses physiques qui sont considérées comme le numéro d'enregistrement, c'est pour cette raison que cette méthode n'est pas utilisée. Afin de palier à cela, l'organisation relative a été créée.

B – Organisation relative :

Tous les enregistrements doivent avoir une longueur fixe et ont chacun un numéro relatif (1,2,3, ...) appelé clé qui est leur numéro d'ordre dans le fichier.

Le principe est que la clé est un numéro d'enregistrement qui est un nombre entier permettant de déterminer directement l'adresse relative de l'article par rapport du début du fichier qui le contient.

Le fichier est stocké de manière discontinue sur le support mais les clés d'accès sont de valeurs continues.

L'organisation relative est l'équivalent d'un grand tableau à une seule dimension stocké sur MS et n'est possible que sur support adressable.

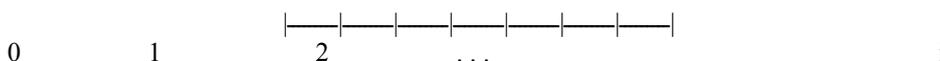
B.1 - Réalisation d'un fichier relatif :

Le fichier est composé d'un ensemble de pages (bloc). Une adresse relative est un numéro de page plus un déplacement par rapport au début de la page.

Un fichier relatif a des articles de longueur fixe. Chaque article possède une clé unique qui est son numéro d'ordre dans le fichier.

Calcul de l'adresse :

AR (Adresse Relative) = $L * i$ où L : longueur d'un article si $i \geq 0$
Ou bien $AR = L * (i-1)$ si $i \geq 1$ (on numérote à partir de 1)



Avantages :

- ✓ Lorsqu'on connaît la clé de l'article, on y accède en une entrée/sortie (E/S).
- ✓ Si le fichier a des articles de longueur variable alors on fixera la longueur par $L = \text{Max}(L_0, \dots, L_n)$.
- ✓ Dans ce cas, le fichier relatif est construit avec des articles de longueur celle de l'article de longueur maximum.

Inconvénients :

- ✓ Il faut être capable d'associer un numéro à chaque article et il faut conserver cette association dans le programme.
- ✓ Si les articles ne sont pas de longueur fixe alors on aura une perte de place.
- ✓ Si les numéros relatifs ne sont pas continues alors le fichier sera plein de trous d'où on aura aussi une perte de place.

C – Organisation Aléatoire :

Son principe est que la clé d'accès est une donnée figurant dans l'un des attributs de l'article (N°SS, Matricule, ...).

A partir de la clé, la méthode d'accès calcule l'adresse relative d'une zone appelée paquet dans lequel est placé l'article.

La fonction de calcul est appelée « *fonction de hachage* ».

Définition :

Le fichier dans lequel les articles sont placés est constitué de paquets dont l'adresse est calculée à l'aide d'une fonction de hachage appliquée à la clé. A l'intérieur d'un paquet, les articles sont rangés à la suite dans l'ordre d'arrivée.

- ✓ Adresse relative au paquet $ARP = N * L$ ou $ARP = (N-1) * L$ où L : Longueur du paquet et N : numéro du paquet.
- ✓ Adresse relative à l'article $ARA = ARP + \text{déplacement dans le paquet}$.

Structure interne d'un paquet :

Adresse 1 ^{ère} octet libre dans paquet	
L1	Article a1
L2	Article a2
.	
.	
.	



Répertoire (ou Table d'index) :

Clé Art1	N° paquet
Clé Art2	" "

C.1 - Création / Consultation :

- ✓ Définir le nombre de paquets.
- ✓ Trouver la fonction de hachage.
- ✓ Rechercher le N° de paquet par recherche séquentielle.

C.2 - Insertion :

- ✓ Elle doit se faire à la 1^{ère} adresse libre du paquet qui doit le contenir.
- ✓ Avant d'effectuer l'insertion que l'article n'existe pas déjà.
- ✓ S'il n'y a pas de place libre alors il y a débordement.

C.3 - Suppression :

L'article est repéré dans un paquet puis supprimé logiquement.

C.4 – Techniques de Hachage :

Une fonction de hachage doit être choisie de sorte à distribuer uniformément les articles dans les paquets (parfois on parle aussi de blocs). Plusieurs techniques sont possibles :

1 - Le modulo :

C'est la fonction la plus utilisée. Elle consiste à prendre pour numéro de paquet le reste de la division euclidienne de la clé par le nombre de paquets. On utilise la fonction « modulo le nombre total de paquets ».

2 - Le pliage :

Consiste à choisir et combiner ou additionner les bits de la clé à l'aide d'opérateurs binaires (par exemple le « ou » exclusif XOR). On utilise souvent le pliage + le modulo.

3 - Les Bits du milieu :

Consiste à prendre dans le milieu du code binaire de la clé un nombre de bits suffisant pour numéroter le paquet ou bloc.

4 - Changement de base :

Consiste à changer la base (par exemple, passer vers la base 5) de numérotation de la clé puis à lui appliquer généralement la fonction modulo.

5 - Fonction aléatoire :

Le numéro du paquet est calculé sur la base de la formule suivante :

$$\text{N}^\circ \text{ paquet} = \text{partie_entière}(\text{Aléatoire}(\text{clé}) * \text{nombre_de_blocs})$$

C.5 - Collisions / Débordements :

On parle de collision lorsqu'une clé d'un nouvel article à insérer donne l'adresse d'un paquet plein. Ceci conduit à la situation de débordement. Il y a plusieurs techniques pour la gestion des débordements :

1 - adressage ouvert :

Un article en débordement est placé dans le 1^{er} paquet suivant qui a de la place. Il faut alors mémoriser tous les paquets dans lesquels un paquet plein a débordé et ce dans l'entête du paquet ou bloc ou bien chaîner les enregistrements ayant débordés.

2 - chaînage de paquets :

On constitue des paquets logiques en chaînant le paquet plein à ses paquets de débordement.

3 - rehachage :

On applique une 2^{ème} fonction de hachage aux articles qui débordent d'un paquet plein. Ils sont alors placés dans des paquets spéciaux de débordement (c'est le hachage virtuel).

⚡ Remarque :

La gestion des débordements dégrade les performances de cette organisation car elle a pour effet d'augmenter le mouvement des têtes en lecture/écriture.

En Conclusion :

- ✓ L'organisation aléatoire s'adapte à des fichiers de clé quelconque.
- ✓ Elle est simple et a d'excellentes performances tant qu'il n'y a pas de débordements.
- ✓ Si un paquet est une unité de transfert et si la table de correspondance est résidente en mémoire centrale alors une lecture nécessitera 1 E/S et une écriture nécessitera 2 E/S (une lecture et une écriture).

3.3.3 - Organisation indexée :

Notion d'index :

Le principe de base des organisations indexées et des méthodes d'accès indexées est d'associer à la clé d'un article son adresse relative dans le fichier à l'aide d'une table des matières du fichier appelée « *index* ».

Exemple :



Index :

A5	A2	A57	A3
0	4	7	12

Pour accéder à un article dans le fichier indexé, il est nécessaire d'exécuter les étapes suivantes :

- ❖ accès à l'index.
- ❖ recherche dans l'index de la clé d'article désiré.
- ❖ récupération dans l'index de l'adresse relative de l'article ou de l'adresse relative de paquet qui le contient.
- ❖ conversion de l'adresse relative en adresse réelle .
- ❖ accès à l'article (ou au paquet qui contient l'article)
- ❖ transfert de l'article dans la zone du programme utilisateur .

Il existe différentes techniques de réalisation d'index. Parmi les types d'index nous avons :

1. Index combiné : l'indexation est faite sur plusieurs champs combinés
2. Index compacté : ce sont des index dont la taille est réduite ou compressée
3. Index hiérarchisé : facilite l'exploration d'un index important. Cette technique est utilisée pour les index volumineux

Nous allons nous intéresser aux index hiérarchisés.

Index hiérarchisé :

Un index peut être vu comme un fichier de clés. On peut créer de nouveau un index sur ce fichier de clé. On parle alors d'*index d'index* et ainsi de suite. Ce qui nous permet d'avoir plusieurs niveaux d'index.

Définition :

Un index hiérarchisé à 2 niveaux est un index trié divisé en blocs ou paquets possédant lui même un index. La clé de chaque entrée de ce dernier étant la plus grande du paquet.

Plusieurs variantes sur un index sont possibles :

- ❖ index trié ou non trié.
- ❖ index dense ou non dense.

Notion de densité :

C'est le rapport entre le nombre de clé présente dans le fichier d'index et celle présentes dans le fichier de données.

$$D = \frac{\text{Nbre clé de l'index}}{\text{Nbre d'articles}}$$

$$D = \begin{cases} 1 & \text{IndexDense} \\ \text{inférieure à 1} & \text{IndexNonDense} \end{cases}$$

L'index dense contient toutes les valeurs des clés.

A - ORGANISATION INDEXEE SIMPLE :

Les articles sont rangés en séquentiel dans un fichier dont l'index est dense et organisés sous forme d'un B-Arbre.

Définition :

Un fichier indexé simple est un fichier non trié, d'index trié dense organisé sous forme d'un B-Arbre.

1^{ère} problème : comment conserver l'index trié ?

solution :

- l'index est divisé en pages .
- lors de la 1^{ère} écriture, les pages ne sont pas complètement remplies.
- lors d'une 1^{ère} insertion, si une page est pleine, elle est éclatée en deux pages à demi-pleines. La clé médiane est remontée au niveau supérieur.

2^{ème} problème : comment conserver la densité de l'index ?

En réalité, l'index est dense au dernier niveau et donc toutes les clés sont gardées au plus bas niveau. Quand une page éclate, la clé médiane devient la plus grande clé de la 1^{ère} page résultante de l'éclatement. Cette clé est dupliquée au niveau supérieur de l'index.

Avantages :

- stockage séquentiel des articles .
- garder un historique des index.

Inconvénients :

- lecture séquentielle de la clé dans l'index très coûteuse.

B – Organisation séquentielle indexée : (ISAM)

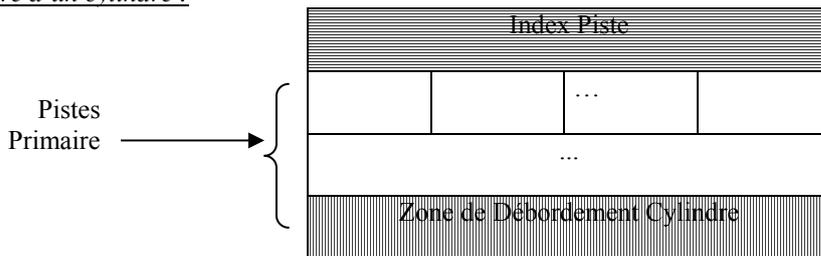
C'est une organisation d'IBM pour les systèmes d'exploitation DOS et OS-VS. Un fichier séquentiel indexé ISAM (Indexed Sequential Access Method) est un fichier trié, d'index trié non dense, composé d'une zone de débordement et d'une zone index.

Donc un fichier ISAM comprend trois zones :

ZP : la zone primaire.

Elle se compose de cylindres successifs dont certaines pistes sont réservées pour les index et les zones de débordement cylindre. Les articles sont stockés à la création du fichier par ordre croissant des clés.

Structure d'un cylindre :



ZD : Zone de Débordement.

Elle reçoit les articles rajoutés au fichier par insertion ultérieure. On distingue 2 types de zone de débordement :

- ZDC : la Zone de Débordement Cylindre composée de quelques pistes sur chaque cylindre.
- ZDI : la Zone de Débordement Indépendante se trouve généralement sur les derniers cylindres.

En zone de débordement, les articles ne sont pas bloqués mais chaînés pour former une piste logique qui débute par une piste physique en ZP et s'étend grâce au chaînage en ZD.

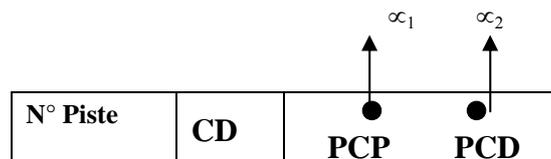
ZI : Zone Index.

Il existe obligatoirement 2 niveaux d'index et optionnellement un 3^{ème} index de piste qui sont :

- index piste.
- index cylindre.
- index maître (niveau optionnel).

Index Piste :

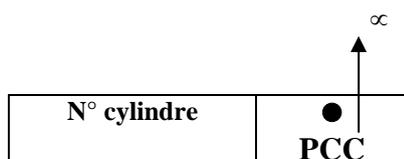
Il a la structure suivante :



- Cd : contrôle de débordement, précise l'adresse du dernier article de la zone de débordement.
- PCP : Plus grande Clé en zone Primaire avec adresse α_1 .
- PCD : Plus grande Clé en zone de Débordement avec adresse α_2 .
- α_1 : adresse de la piste primaire.
- α_2 : adresse de la piste de débordement contenant l'enregistrement suivant le dernier enregistrement de la piste primaire.

Index Cylindre :

Il nous renseigne sur chaque cylindre. Il a la structure suivante :



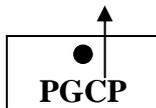
- PCC : Plus grande Clé du cylindre avec adresse α .

Il existe un index de cylindre par fichier. L'index de cylindre est généralement rangé dans une zone particulière comme par exemple au début de la zone de débordement indépendante (ZDI).

Il permet à partir d'une clé d'article de sélectionner le cylindre où doit se trouver l'article.

Index Maître :

C'est un index optionnel qui est utilisé lorsque l'index cylindre est trop grand. Il devient index piste de l'index cylindre. Il contient une entrée par piste de l'index cylindre.



- PGCP : Plus grande Clé dans la piste de l'index cylindre.

B.1 - Insertion :

1. recherche de sa séquence dans la piste logique.
2. s'il est en zone primaire, les articles suivants sont déplacés et le dernier est écrit en zone de débordement.
3. mise à jour des chainages.
4. par contre, s'il est en zone de débordement alors il est écrit en zone de débordement (ZDC ou ZDI selon le cas).
5. mise à jour des chainages.

Les articles qui débordent de la ZP d'abord insérés en ZDC. Lorsque la ZDC est saturée on utilise la ZDI.

Quand il y a trop de débordement, il est possible que le chainage nous envoie de la ZDC à la ZDI puis à la ZDC (et ainsi de suite).

Cela conduit à des déplacements nombreux du bras du disque, chose qui altère les performances de cette organisation.

Remarque :

Le nombre de paramètres dans chaque entrée de l'index d'un cylindre peut être plus important. On peut par exemple trouver le nombre d'octets vides par piste, le nombre d'article par piste, ...

Avantages de la méthode :

- Le fichier est trié, ce qui facilite l'accès en séquentiel.
- 3 accès au maximum sont nécessaires pour récupérer un article quand il n'y a pas de débordement.

{ 1 accès à l'index cylindre.
1 accès à l'index piste.
1 accès à l'article.

Inconvénients :

La gestion des débordements est complexe et dégrade les performances de la méthode.

C - Organisation séquentielle indexée régulière : (VSAM)

VSAM : Virtual Sequential Acces Method, proposée par IBM system OS/VS.

Définition :

Les fichiers VSAM sont des fichiers triés d'index trié non dense dont l'ensemble fichier + index est organisé sous forme d'un B- arbre.

Principe :

Le fichier est divisé en aires.

- une aire = ensemble de piste d'un même cylindre ou de cylindres contigus ou une aire = ensemble d'intervalles.
- intervalle = partie de piste ou plusieurs pistes consécutives lues en une seule E/S.
- Lorsqu'un intervalle est saturé, il est éclaté en 2 intervalles.
- Lorsqu'une aire est saturée, elle est éclatée en 2 aires.
- Ainsi, le fichier se réorganise de lui même.

Avantage par rapport à ISAM :

Le fichier est moins dépendant des mémoires secondaires qu'un fichier ISAM car on utilise les notions virtuelles d'aire et d'intervalle qui remplacent les notions physiques de cylindre et de piste.

Les débordements ne perturbent plus l'organisation puisque l'évolution du fichier doit au principe du B-arbre.

1 - Etude de la zone de donnée ZD :

C'est un ensemble d'aires et d'intervalles fixés dès la 1^{ère} écriture du fichier avec des articles triés par ordre croissant des clés. A l'initialisation, de la place libre est prévue pour les mises-à-jour (application de principe du B-arbre chaque neud, excepté le noeud et les feuilles a au moins $\lfloor d/2 \rfloor$ fils). Dans chaque aire des intervalles libres sont laissés. Dans chaque intervalle de la place est laissée libre suivant un certain pourcentage % défini (c'est le facteur de blocage).

Exemple :

si taux de remplissage initial = 75 % :

Zone index						
A4	A5	A6		A10	A11	A13
0			1			

On a 04 intervalles (nous avons choisi 1 intervalle = un peu plus qu'une piste.

Etude de la zone d'index :

Comme pour ISAM, on a 2 ou 3 niveaux d'index.

- Index d'intervalles (par analogie, index de piste dans ISAM). Il en existe un par fichier. La structure d'une entrée de l'index est : (plus grande clé de l'aire, adresse de l'aire).

Si l'index d'aire est trop grand (plus d'un intervalle) , alors il y a création d'un index maitre.

Exemple : index d'intervalle

Aire0	A20				Aire 1	A30			

index d'aire :

A20	0	A30	1
-----	---	-----	---

Insertion :

- ✓ Rechercher l'intervalle concerné.
- ✓ Si intervalle est non plein alors l'article est rangé dans l'intervalle en bonne place dans l'ordre croissant des clés.
- ✓ Si l'intervalle est plein, alors il est éclaté en 2 intervalle à demi-pleins.

Deux Cas sont possibles :

- Il existe un intervalle libre dans l'aire alors il est utilisé comme 2^{ème} intervalle d'éclatement.
- Allocation d'une nouvelle aire au fichier et on équilibre les 2 aires afin d'obtenir 2 aires à demi-pleines toujours en respectant le tri.

Avantage de la méthode :

- ✓ Comme ISAM séquentiel trié facilite l'accès en séquentiel .
- ✓ Le temps d'accès à un article toujours en 3 E/S.

Inconvénients :

Dans le cas de l'éclatement, l'écriture des articles dans le cadre de rééquilibrage peut être de longue durée. Il faut prévoir suffisamment d'espace dans les intervalles pour éviter au maximum la réorganisation des aires

CONCLUSION :

Les méthodes d'organisation en séquentiel indexé sont les plus utilisées pour la gestion des bases de données et permettent des niveaux de sophistication plus élaborés.

Index multiples (plusieurs index sur un même fichier) qui autorisent plusieurs chemins d'accès dans une même base de données (index primaire suivant l'identifiant).

Tableau récapitulatif :

		FICHER	
INDEX	dense	Trié	Non trié
		Possible	IS3
	Non trié		Possible
	Non dense	Trié	ISAM / VSAM
	non trié		