

# THESE

PRESENTEE

A L'UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE

HOUARI BOUMEDIENE

POUR OBTENIR

LE GRADE DE MAGISTER EN INFORMATIQUE

OPTION : SOFTWARE

PAR

Moussa BENHAMADI



SUJET :

M I S T E P

2<sup>ème</sup> Partie:

SYSTEME DE GESTION DE FICHIERS ET  
EDITION DE TEXTE.

Soutenu le 07 Décembre 1982 devant la commission d'examen

M. B. SANSAL  
M<sup>me</sup> F. ANDRE  
MM. C. B. BENYELLES  
J. MOSSIERE  
S. A. LARIBI

}

Président

Examineur s

Rapporteur

# REMERCIEMENTS

====000§000====

Nous exprimons toute notre reconnaissance à Monsieur S.A. LARIBI, Professeur, Directeur du Département Informatique et Logique Mathématique à l'U.S.T.H.B. pour avoir constamment suivi ce travail et pour les conseils constructifs qu'il nous a toujours prodigués.

Nos vifs remerciements vont à Monsieur M. SANSAL, Professeur à l'U.S.T.H.B., Directeur de la Division Simulation et Contrôle au Centre des Sciences et de la Technologie Nucléaire (C.S.T.N) pour l'honneur qu'il a bien voulu nous faire en présidant ce jury.

Que Messieurs J.M. ALBERTINI et J.L. LEONHART respectivement Directeur de l'Institut de Recherche en Pédagogie de l'Economie et en Audiovisuel pour la Communication dans Les Sciences Sociales (IRPEACS) - CNRS - LYON et Chef du Département Informatique trouvent un témoignage de notre gratitude pour l'accueil et les moyens matériels et humains qu'ils ont mis à notre disposition durant notre séjour à l'IRPEACS.

Nous remercions Madame F. ANDRE, Professeur à l'Université de RENNES, Messieurs J. MOSSIERES, Professeur à l'U.S.M.G. et C.B. SENYELLES, Maître de Conférence à l'U.S.T.H.B., d'avoir accepté de faire partie de ce jury.

Nos remerciements vont au personnel du service I.D.R du C.S.T.N., particulièrement à Messieurs M. EL-MAOUHAB, M. LAZIB et son équipe.

Nous tenons enfin à remercier tous ceux qui ont participé, de près ou de loin, à la réalisation de cette thèse.

## TABLE DES MATIERES

-----o0\$0o-----

### CHAPITRE I : STRUCTURE DE STOCKAGE D'INFORMATIONS EN MEMOIRE SECONDAIRE

I	: Concepts et terminologie	1.1
II	: Evaluation des structures de stockage	<b>1.4</b>
11.1	: Diagramme de structure de donnkes	1.4
11.2	: Structures de stockage et allocation d'espace	1.5
11.3	: Exemples de structure de donnkes	<b>1.10</b>
III	: Structure d'allocation	<b>1.14</b>

### CHAPITRE II : STRUCTURES DE GESTION DE L'INFORMATION

I	: Les principales fonctions d'un système de fichiers	<b>11.2</b>
1.1	: Definition du rôle d'un SGF	<b>11.2</b>
1.2	: Fonctions de base d'un SGF	<b>11.3</b>
1.2.1	: Identification d'un fichier	11.3
a-	Le catalogage	11.3
b-	La protection'	11.5
1.2.2	: Organisation et méthodes d'accès	<b>11.8</b>
1.2.2.1	: Organisation de base d'un fichier	11.8
1.2.2.2	: Organisation d'accès	11.9
1.2.3	: Implantation sur mémoire secondaire	<b>11.10</b>
1.2.4	: Contrôle de fichiers	11.12

II : Classification des système de gestion de fichiers	II.13
II.1 : Définition fonctionnelle d'un SGF	II.14
II.2 : Présentation de quelques qualités fonctionnelles d'un SGF	II.14
III : Exemple d'un SGF : le FMS2 du MMT2	II.18
III.1 : Identification d'un fichier	II.18
III.2 : Organisation et modes d'accès	II.19
III.3 : Implantation des fichiers en mémoire secondaire	II.20
III.4 : Contrôle de fichiers	II.21

### CHAPITRE III : SYSTEME DE GESTION DE FICHIER

I : Description générale	111.2
1.1 : Objectifs du système de fichiers	111.2
1.2 : Support d'information	111.3
1.3 : Concepts de base du système	111.4
1.4 : Catalogage	111.5
II : Structure hiérarchique de fichiers	111.6
II.1 : Modèle général	111.6
11.2 : Structure de fichiers du système MISTEP	111.10
11.3 : Les différents éléments de la structure	111.10
11.4 : Manipulation de catalogues	111.13
III : Identification et contrôle d'accès	111.14
111.1 : Qualification d'un utilisateur	111.14
111.2 : Identification d'un utilisateur	111.17
111.3 : Identification d'un fichier	111.17
111.4 : Partage de fichiers	111.19
111.5 : Contrôle d'accès	111.20
IV : Allocation et gestion de l'espace disque	111.23
IV.1 : Allocation de l'espace disque	111.23

IV.2	: Stratégie d'allocation du système	111.29
V	: Organisation et méthode d'accès	111.33
V.1	: L'organisation séquentielle simple	111.33
V.2	: L'organisation séquentielle partitionnée	111.36
V.3	: L'organisation séquentielle indexée	111.36
V.4	: L'organisation partitionnée	111.38
V.5	: L'organisation libre ou directe	111.39
VI	: Organisation du système de gestion de fichiers	111.39

#### CHAPITRE IV : EDITION LIE TEXTES

I	: Editeur de textes	IV.1
I.1	: Les différents types d'éditeurs	IV.1
I.2	: Différences entre éditeurs	IV.2
II	: Editeur de document	IV.4
III	: L'éditeur de document EDIDOC	IV.5
IV	: L'éditeur de texte du système MISTEP	IV.6
IV.1	: Langage de commande	IV.7
IV.2	: Etat d'édition	IV.8
IV.3	: Organisation des fichiers	IV.8
IV.4	: Particularité par rapport aux autres éditeurs	IV.9
IV.5	: Organisation de la T.P.E	IV.9
IV.6	: Organisation logique des articles	IV.11
IV.7	: Organisation d'une page de données éditable	IV.12
IV.8	: Exemple de recherche d'une ligne	IV.13
IV.9	: Reconstitution du fichier origine	IV.15
IV.10	: Les différents états du système	IV.15
IV.11	: Environnement associé à un utilisateur	IV.15

CONCLUSION

BIBLIOGRAPHIE

## INTRODUCTION

Tous les enseignants s'accordent à penser qu'un apprentissage, quel qu'il soit ne peut être bénéfique que s'il est accompagné d'une mise en pratique des notions théoriques que l'on désire transmettre à l'enseigné.

L'enseignement de l'informatique, en particulier, ne peut se faire correctement que si l'on dispose de l'outil principal de mise en pratique des connaissances théoriques : l'ordinateur.

Les techniques d'écriture et de mise au point de programmes, activité essentielle chez la plupart des postulants à une formation, soulèvent à l'heure actuelle un grand débat. Quelles que soient les solutions proposées à ce problème, les procédures d'utilisation de la machine peuvent se diviser en deux classes:

- soumission de programmes sur cartes perforées au "batch processing".
- utilisation d'un terminal (type machine à écrire ou console de visualisation) en mode conversationnel.

**En mode** "batch processing", l'utilisateur après correction manuelle de son programme par des insertions, remplacements et suppressions de cartes, le dépose dans une file d'attente; les résultats lui sont remis après un délai plus ou moins long. Ce processus est réitéré autant de fois que c'est nécessaire jusqu'à l'obtention des résultats exomptés.

On convient qu'une telle organisation est mal adaptée à un milieu d'enseignement. La soumission et la récupération des

travaux, se faisant généralement par un guichet, empêchant toute communication directe entre l'étudiant et l'ordinateur. Par ailleurs, le temps qui s'écoule entre le dépôt du travail et la récupération des résultats peut être relativement long et à l'issue duquel l'étudiant peut s'apercevoir que son travail a été rejeté à cause d'une erreur sur une carte de contrôle. Il aura ainsi perdu des heures et même une journée.

En mode conversationnel, la procédure d'écriture et de soumission de programmes est différente. L'enseigné dispose d'un terminal au moyen duquel il communique directement avec l'ordinateur pour enregistrer ou corriger son programme sur support magnétique et éventuellement l'exécuter. De tels systèmes, caractérisés principalement par le court délai entre le moment où l'on active un travail et celui où l'on récupère les résultats, supposent l'existence d'un logiciel spécifique.

Le système MITRA/125 (SEMS) disponible à l'université n'offre pas un tel environnement de travail. Nous avons donc été amenés à développer MISTEP (Multi-users Interactive System for Text Editing and Program Preparing) permettant la gestion de sept terminaux à partir desquels les utilisateurs peuvent constituer des fichiers, mettre au point des programmes et les soumettre au moniteur d'enchaînement de travaux.

Pour présenter ce travail d'équipe, nous avons été amenés, pour des raisons administratives, à le scinder en deux parties:

- Dans la première partie, consacrée à l'implémentation du système et son langage de commande, nous présenterons au chapitre I le concept de machine virtuelle I SRO-73 I, I SEA-79 I sur lequel est basé le moniteur MMT2 supportant MISTEP. Au chapitre II nous décrirons l'architecture et les composantes de ce système (tâches, mémoires, périphériques etc...).

## STRUCTURES DE STOCKAGE D'INFORMATIONS EN MEMOIRE SECONDAIRE

Les systèmes de gestion de l'information ont rapidement évolué durant les quinze premières années de l'histoire du traitement de l'information par les machines numériques ou ordinateurs. Durant cette période de nouveaux concepts sont nés et une nouvelle terminologie est adoptée pour désigner et référencer ces concepts. De même qu'une représentation graphique des structures de données est développée pour illustrer les différences et les relations qui existent entre ces concepts. [ BAC-69-a ]

Dans ce chapitre après avoir défini certains termes qui permettent de désigner les concepts utilisés dans cette étude, nous présentons en nous basant sur l'étude faite sur ce sujet par C.W. Bachman [ BAC-72-b ] quelques modèles de structures de stockage de l'information en mémoire auxiliaire et qui seront cités dans les chapitres deux et trois du présent mémoire.

### I - CONCEPTS ET TERMINOLOGIE

La partie 'Structure de Stockage' des systèmes de gestion de l'information en mémoire secondaire est chargée du catalogage des fichiers et de ses articles sur support d'information de manière que toute donnée rangée puisse être récupérée.

La structure de stockage comporte une structure logique représentée par le fichier logique et ses éléments et une structure physique constituée par les **périphériques**, les volumes et leurs éléments. Chacune de ces structures est formée d'un ensemble de concepts que nous rappelons.

a - Stockage physique

Le principal concept sur lequel repose cette structure est le concept de 'Périphérique' qui est le support temporaire ou permanent d'un volume, et ayant un mécanisme de sélection de cylindre et de réalisation d'opérations d'entrée-sortie.

Le volume est le support de stockage d'information sur lequel on peut enregistrer ou lire des données. Il est associé à un type de périphérique et constitué d'un ou plusieurs cylindres.

Le cylindre est une unité de stockage d'information caractérisée par le fait que le transfert consécutif d'informations d'un même cylindre entre **la** mémoire secondaire et **la** mémoire centrale est **sensiblement** plus rapide que **le** transfert de deux cylindres. Il est formé d'une ou plusieurs pistes. Le nombre de pistes par cylindre est fixé pour un type d'unité disque.

Cette caractéristique d'optionnalité de transfert de données justifié **la** rapidité d'accès aux articles d'un fichier formé d'un espace mémoire secondaire continu (c'est-à-dire appartenant à un cylindre ou plusieurs cylindres contigus) par rapport au transfert de données d'un fichier 'éparpillé' sur des cylindres non nécessairement contigus.

### - 1.3 -

La piste est l'unité du support de stockage d'information caractérisée par un transfert sériel d'articles consécutifs plus rapide. Elle est subdivisée en un ou plusieurs enregistrements physiques ou secteurs.

Le nombre de secteurs par piste et la longueur d'un secteur sont spécifiques à l'unité disque pour laquelle ils peuvent être fixes ou variables selon les systèmes de gestion de l'information.

On appelle granule l'unité élémentaire ou quantum d'allocation de l'espace disque aux fichiers. La taille du granule, exprimée en nombre de secteurs, est choisie selon la stratégie d'allocation utilisée.

Le bloc est l'unité d'allocation de l'espace disque pour un fichier donné. Sa taille est exprimée en granules. Il constitue l'unité de transfert entre la mémoire centrale et la mémoire auxiliaire.

Le secteur est l'unité du support de stockage qui peut être transféré sans avoir besoin de lire ou écrire les secteurs qui lui sont adjacents sur la même piste.

Un volume peut être subdivisé en zones d'extension représentant un espace adressable, éventuellement continu en mémoire secondaire.

## b - Stockage logique

Le fichier et l'unité de stockage logique désigné ou assigné pour un éventuel traitement sur son contenu. Il sert de conteneur aux articles logiques. Il est formé d'une ou plusieurs pages.

La page est l'unité de stockage logique d'information, représentant un espace continu. Elle peut contenir un ou plusieurs articles.

L'article ou enregistrement logique est l'unité logique créée par l'exécution d'une commande **ECRIRE** et consécutivement accessible par l'exécution d'une commande **LIRE**. Un article est formé de champs valeurs.

## II - EVALUATION DES STRUCTURES DE STOCKAGE

### 11.1 - Diagramme de structure de données

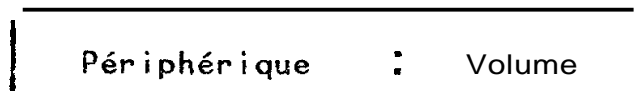
**La** technique graphique utilisée pour illustrer les différentes structures de stockage est basée sur deux éléments : le rectangle et la flèche. Chaque rectangle représente une classe d'entités qui supporte un concept de base telque 'fichier' ou 'volume'. Chaque flèche représente une relation entre les occurrences de deux entités. Le sens de la flèche montre qu'une occurrence d'une classe (origine) est associée avec plusieurs occurrences d'une autre classe (arrivée).

Le diagramme de la structure de donnée de la figure 1.1 peut être lu de la manière suivante :

1. Il existe une classe d'entités appelée 'FICHIER'
2. Il existe une classe d'entités appelée 'ARTICLE'
3. Il existe une classe d'entités appelée 'CHAMP'
4. Pour toute occurrence de la classe d'entité 'FICHIER' il existe une relation avec plusieurs occurrences de la classe d'entité 'ARTICLE'
5. Toute occurrence de la classe d'entité 'ARTICLE' est en relation avec une ou plusieurs occurrences de la classe d'entité 'CHAMP'.

Le type de diagramme de structure est aussi utilisé pour décrire l'éclatement d'une classe d'entités en plusieurs classes avec de nouvelles relations entre-elles. Si un rectangle contient plus d'un nom de concept, une occurrence de l'un est en relation 1 : 1 avec une occurrence de l'autre.

Exemple :



Une occurrence de l'entité périphérique est en relation avec une occurrence de l'entité volume.

## 11.2 - Structures de stockage et allocation d'espace

Parmi les concepts cités précédemment, il existe trois importantes hiérarchies pour l'étude des structures de stockage ce sont :

- La hiérarchie dans la structure de stockage physique : (périphérique, volume, cylindre piste, secteur)

- La hiérarchie dans la structure de stockage logique : (fichier, page, article, champ valeur)
- La hiérarchie dans la structure d'allocation mémoire : (extend, bloc, granule)

On retrouve ces hiérarchies dans toute étude de structure de stockage de l'information [ KAT-71 ]. Ce sont les relations qui existent entre les éléments de ces hiérarchies qui permettent d'effectuer une opération d'entrée-sortie.

Mais si ces relations varient selon les implémentations (environnement matériel et logiciel) et les objectifs des systèmes de gestion de l'information, les hiérarchies définies conservent leur intégrité.

a - Structure de stockage physique

Un volume est formé de  $n$  cylindres, chaque cylindre est constitué de  $p$  pistes, chaque piste est subdivisée en  $m$  secteurs. Les valeurs de  $n$ ,  $p$ ,  $m$  dépendent de la nature et du type du périphérique supportant le volume et du volume lui-même, dans certains cas.

Soient les types de périphériques suivants :

- La bande magnétique

Un volume bande magnétique ne comporte qu'un seul cylindre formé d'une piste qui peut être formée de plusieurs enregistrements physiques. (fig. 2)

FIG. 1.1

STRUCTURE DE FICHIERS

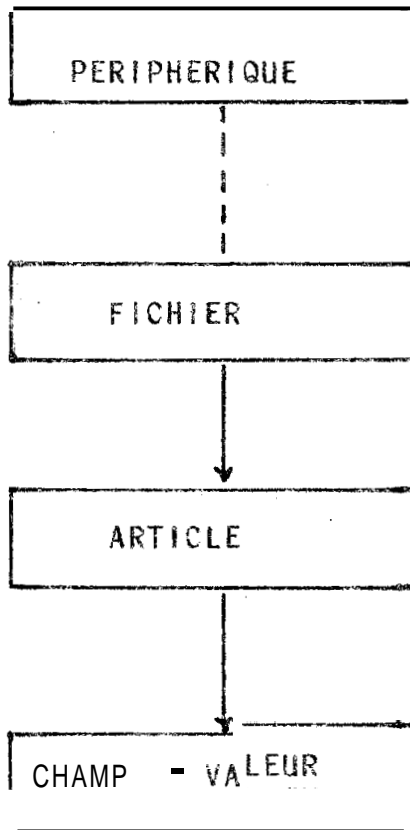
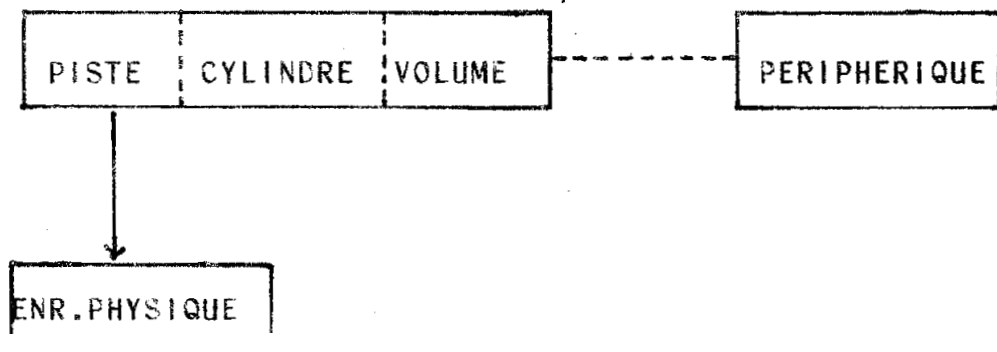


FIG. 1.2

STRUCTURE PHYSIQUE SUR BANDE MAGNETIQUE



- Le disque magnétique

Le nombre de cylindres par volume disque dépend **ties** caractéristiques de l'ensemble matériel : périphérique - disque.

Le nombre de pistes par cylindre est **égal** au nombre de faces utiles. A chaque face utile correspond au moins une tête de lecture-écriture. (*Fig. 1.3 et 1.4*)

**La** taille du secteur est aussi une caractéristique du disque. Elle est soit variable pour un type de disque et dans ce cas, elle doit être fixée pour un volume donné, soit elle est fixe pour tous les volumes-disque utilisés. Dans les deux cas cette taille ne doit dépasser celle d'une piste.

Comme on le constate sur les figures **1.2**, **1.3** et **1.4**, les deux concepts 'périphérique' et 'volume' sont reliés par un trait discontinu non orienté. Ceci signifie que cette relation ne devient réelle que **si** le volume est monté sur l'unité périphérique (montage physique par l'opérateur).

Ces deux concepts sont en relation 1:n avec  $n > 1$  si en un site donné, **on** dispose d'une unité disque amovible et d'un certain nombre de disques. Si l'unité disque est à tête fixe (disque non amovible), ils sont en relation 1:1.

b - Structure de stockage logique

La structure de stockage logique est constituée par les concepts de fichier, page, article et champ de valeurs, ainsi que leur accès. (Fig. 1.5)

Le concept de page représente un espace continu dont les éléments sont directement accessibles par numéro d'ordre ou par adresse relative au **début de** cet espace.

Les articles sont séquentiellement ou sélectivement accessibles si la page les contenant est chargée en mémoire centrale. Les champs de valeurs sont accessibles de la même manière, **dans** un article.

c - Structure d'allocation d'espace

Les concepts de structure de stockage logique requièrent une "présence" physique, afin de remplir les fonctions de recipients de données qui leurs sont assignés.

L'espace physique de stockage est alloué aux entités de la structure logique. Ceci est réalisé à deux niveaux d'une structure hiérarchique d'allocation constituée par les deux concepts **EXTEND** et **BLOC**.

**EXTEND** est le concept d'allocation physique d'une portion de l'espace disque à un fichier. **BLOC** est le détail de cette allocation et aussi le concept qui relie une page de la structure

logique à des secteurs (regroupés par granule) de la structure physique.

Le rôle des concepts d'allocation est de réunir effectivement les deux concepts de stockage physique et de stockage logique pour constituer une structure de données.

### 11.3 - Exemples de structure de données

#### a - La bande magnétique

Pour ce type de support d'information, nous avons cinq classes d'entités. Chacune d'elle est formée d'un ou plusieurs concepts.

Les concepts de fichier, extend, volume, cylindre et piste constituent la première de ces classes d'entité et sont donc en relation 1:1 mais en supposant que le volume est monofichier.

Les concepts de page, bloc, secteur constituent la seconde classe d'entité dont les occurrences sont en relation n:1 avec les occurrences de la première classe.

La troisième classe représente le concept d'article ou enregistrement logique d'une page, alors que la quatrième classe représente le champ valeur à l'intérieur d'un article. Enfin, la cinquième classe représente le concept de périphérique.

La figure 1.6 résume cette structure de données.

FIG. 1.3

STRUCTURE PHYSIQUE SUR DISQUE MAGNETIQUE

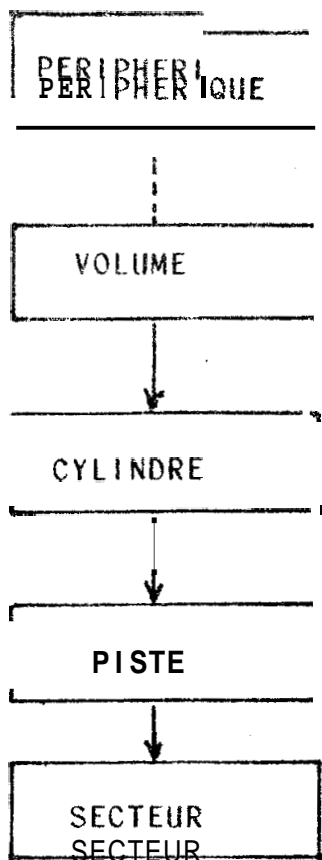
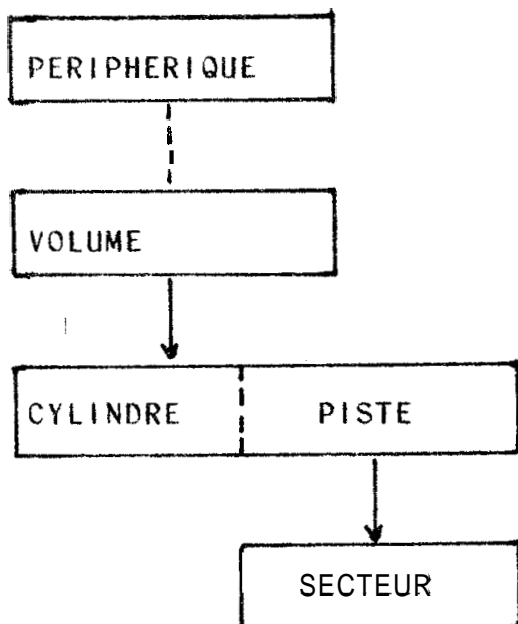


FIG. 1.4

STRUCTURE PARTICULIERE SUR DISQUE (une piste par secteur)



b - Le disque magnétique

Si la structure précédente est transférée de la bande magnétique sur disque en maintenant la relation 1:1 entre les deux concepts volume et fichier, la structure physique change alors que la structure logique demeure constante. Car, si dans la structure physique de la bande magnétique, les concepts volume, cylindre et piste sont en relation 1:1, cela n'est pas le cas lorsqu'il s'agit d'un volume disque, puisque quelque soit le type d'unité périphérique, le disque est formé de plusieurs cylindres.

Un autre changement fonctionnel apporté par cette nouvelle structure est le passage de la mise à jour par réécriture de tout le bloc de données à la mise à jour directe d'un enregistrement physique. Car l'opération LECTURE/ECRITURE sur disque peut être effectuée directement sur un secteur.

Ceci est très avantageux, mais l'insertion d'article entraîne souvent des problèmes lorsque l'ensemble des des enregistrements logiques doivent être maintenus dans un certain ordre.

Etant donné que l'insertion d'articles se traduit par une insertion physique dans la page entraînant ainsi un éventuel débordement, des approches ont été faites pour permettre l'extension du fichier en général. Mais leur efficacité dépend de la taille du fichier et du mode de traitement qui s'y applique. L'une de ces approches consiste à avoir des pages de taille variable :

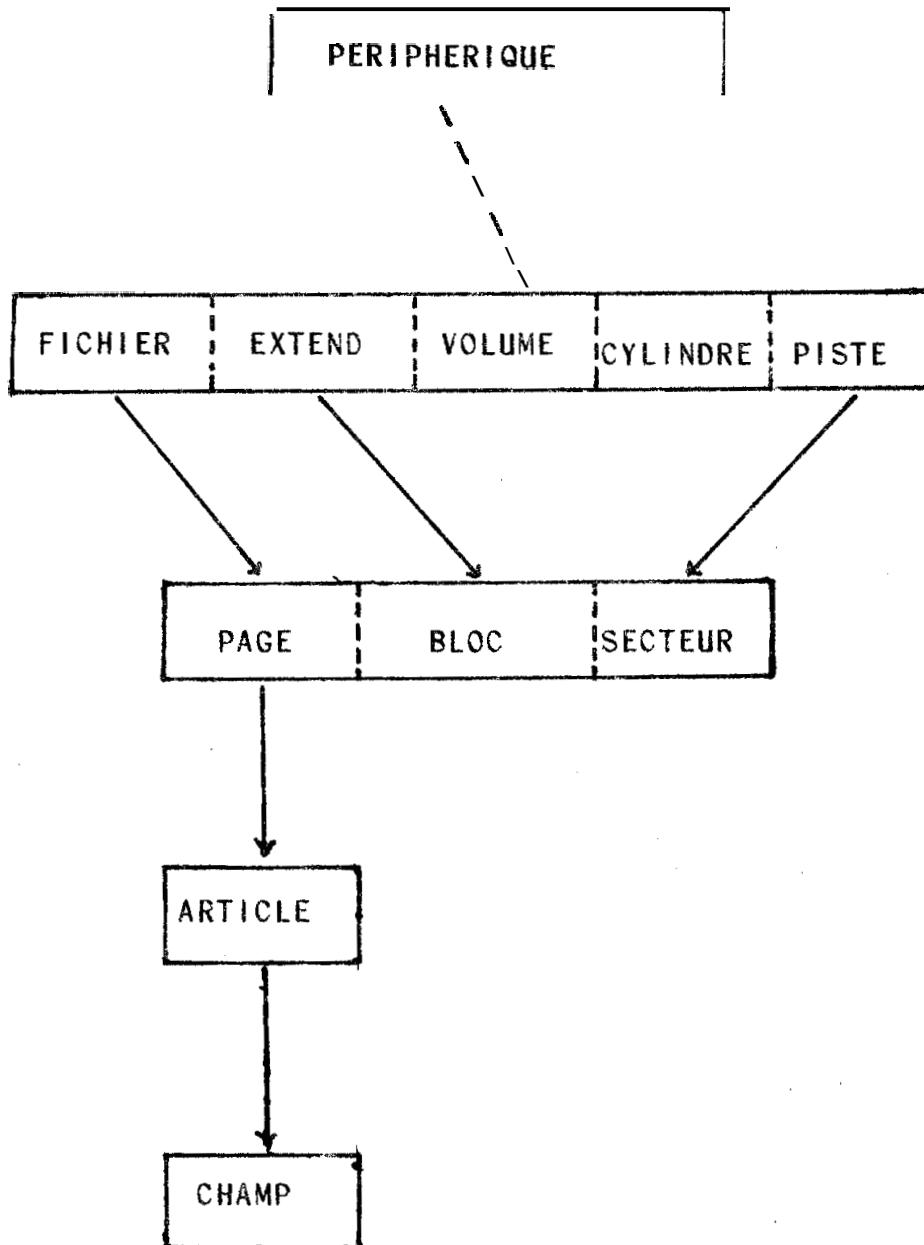


FIG. 1.6

STRUCTURE DE STOCKAGE SUR BANDE MAGNETIQUE

La page augmente d'une unite d'allocation à chaque débordement.

La nouvelle structure permettant l'application de cette approche consiste à avoir les concepts de 'page' et de 'bloc' dans deux classe d'entité distinctes.

Initialement on alloue un bloc à la page et on réserve un groupe de blocs de débordement. Cette reservation de blocs peut être soit réelle soit fictive (logique) et donc on doit prévoir les éléments de gestion de ces blocs. La figure 1.8 illustre cette approche qui est utilisée pour résoudre les problèmes de débordement dans les fichiers séquentiels indexés [ KEE-74 ] .

Les changements qu'on peut noter sur la figure 1.8 sont les extensions de la hiérarchie de structure d'allocation. Sur la figure 1.7, page et bloc constituent une seule classe d'entité; sur la figure 1.8, ils sont séparés pour constituer deux classes d'entités, et une relation 1:n est établie entre ces classes.

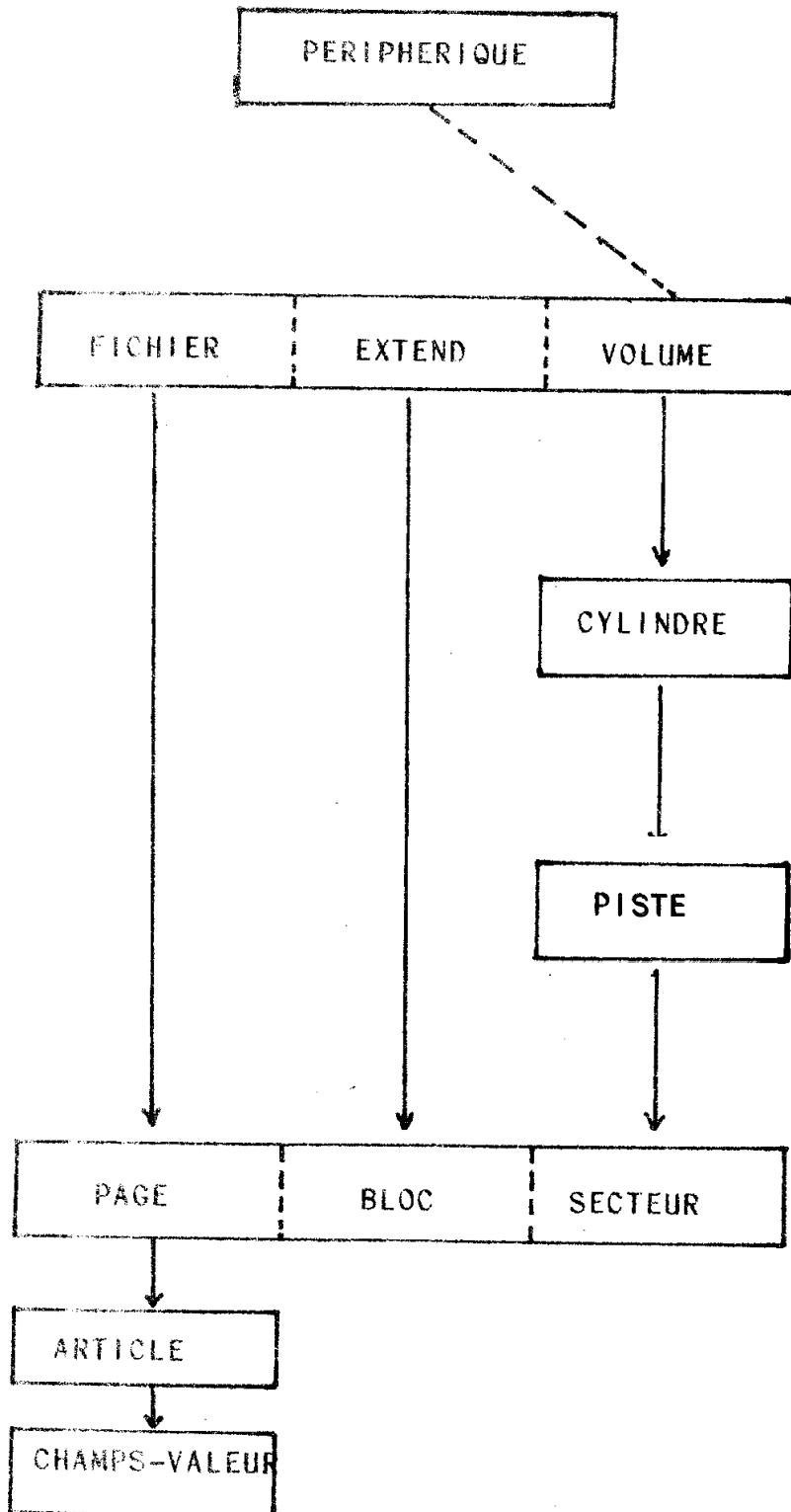
La flèche en pointillés spécifie que, pendant qu'une page consiste en un nombre variable de blocs, à tout instant un bloc peut ne pas être associé à une page. Dans ce cas il est tenu dans l'inventaire des blocs de débordement, comme le montre la flèche verticale en pointillés.

### III - STRUCTURES D'ALLOCATION

La structure d'allocation peut avoir trois formes selon qu'il s'agit d'un volume mono-fichier, d'un volume multi-fichiers ou encore d'un fichier multi-volumes.

FIG. 1.7

STRUCTURE DE STOCKAGE D'ARTICLES BLOQUES SUR DISQUE



Nous avons vu en 11.2 c que dans le cas d'un volume **mono-fichier**, les concepts volume et fichier sont en relation 1:1 et ils appartiennent à la même classe.

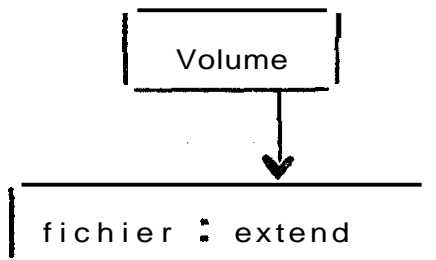
Mais, dans le cas d'un volume **multi-fichiers** ou d'un fichier **multi-volumes**, la relation 1:1 entre ces deux concepts est dépassée et le concept constitue lui-même une classe d'entité indépendante.

Cette évolution peut être schématisée par les quatre formes de structure suivantes :

1° - fusion des concepts fichier, volume, extend.

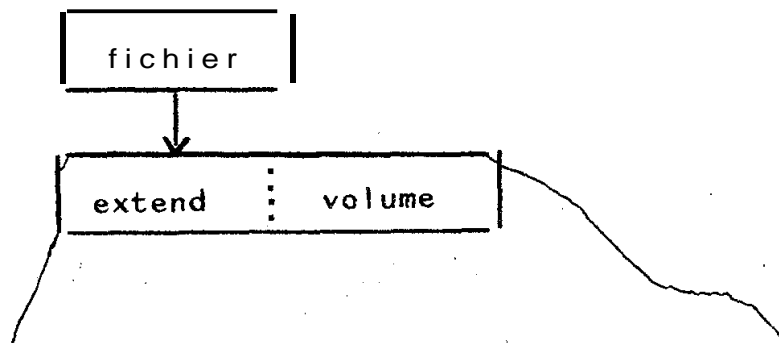


2° - separation du concept volume des deux autres concepts.

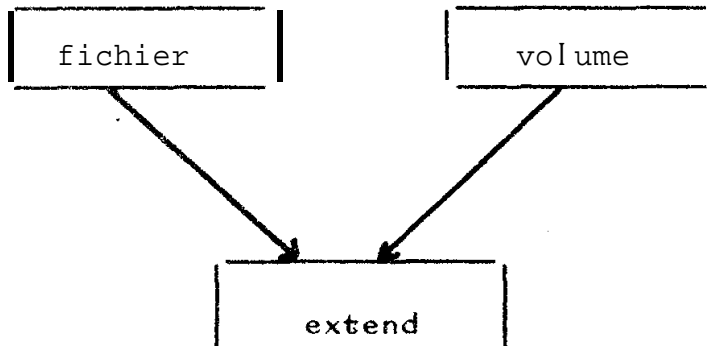


cette structure schématise la notion de volume multi-fichiers.

3° - pour avoir des fichiers répartis sur plusieurs volumes, on introduit une structure où le concept fichier constitue une classe et les concepts volume et extend une autre classe.



4" - Cette forme généralise le problème d'allocation en utilisant **les** trois précédentes structures et offre une liberté totale **dans** la liaison des structures **logique** et **physique**.



Ces quatre structures sont utilisées pour l'allocation statique de l'espace disque à un fichier.

La création et l'évolution dynamique de fichiers sont réalisées à l'aide d'une structure particulière qui permet la liaison fichier-volume dans des conditions avantageuses de gestion d'espace. Elle consiste à allouer au fichier un seul bloc à la fois, à partir de la zone extend pour satisfaire la demande de pages. Les techniques de gestion de pages d'un fichier seront décrites dans le chapitre II.

Soit la structure, figure II.9

La liaison en pointillé entre la page et le bloc signifie que la page peut exister logiquement sans qu'il lui soit alloué un bloc.

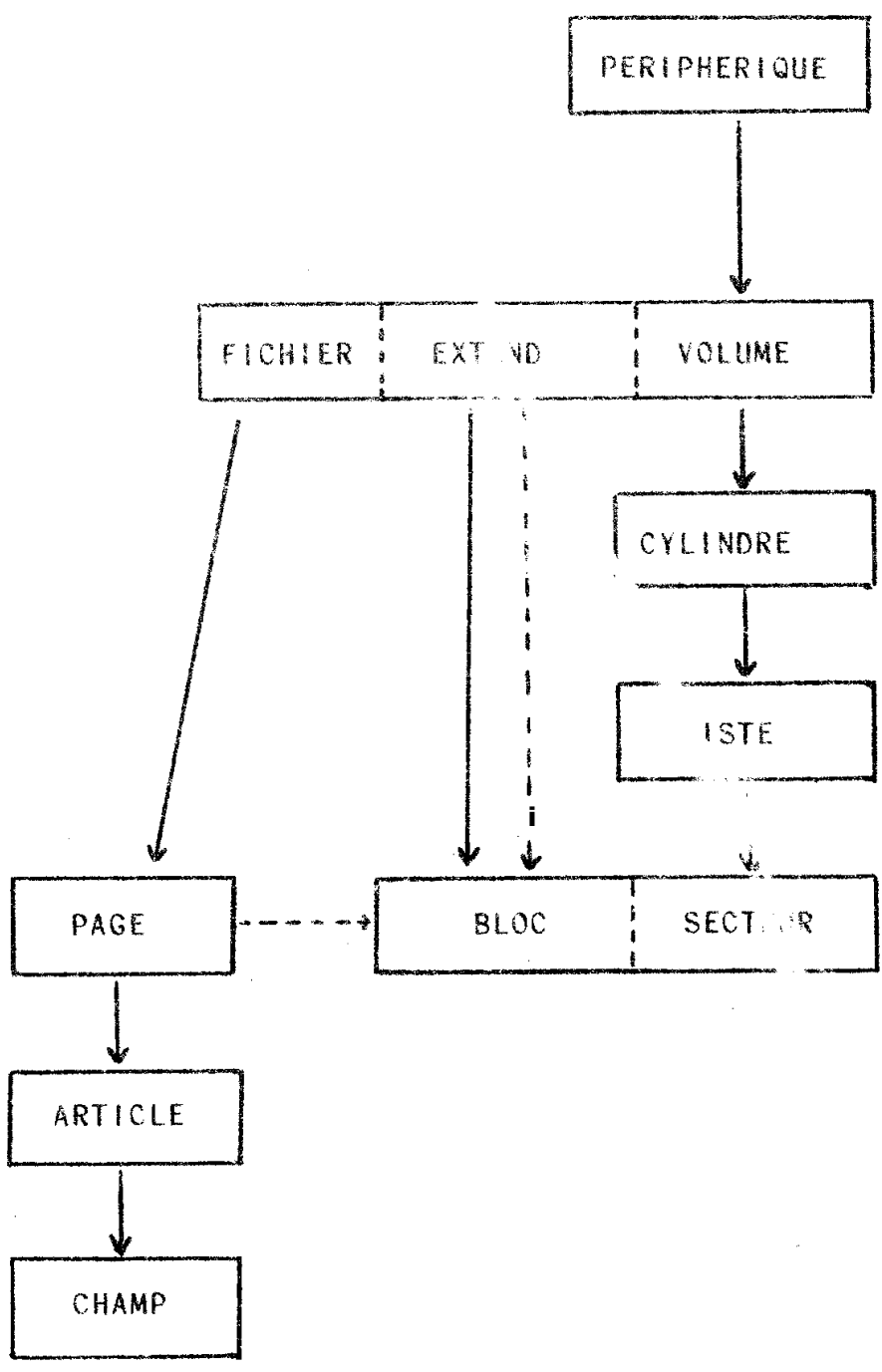


FIG. 1.8

STRUCTURE DE STOCKAGE : EVOLUTION DE LA TAILLE DE LA PAGE

PERIPHERIQUE  
PERIPHERIQUE

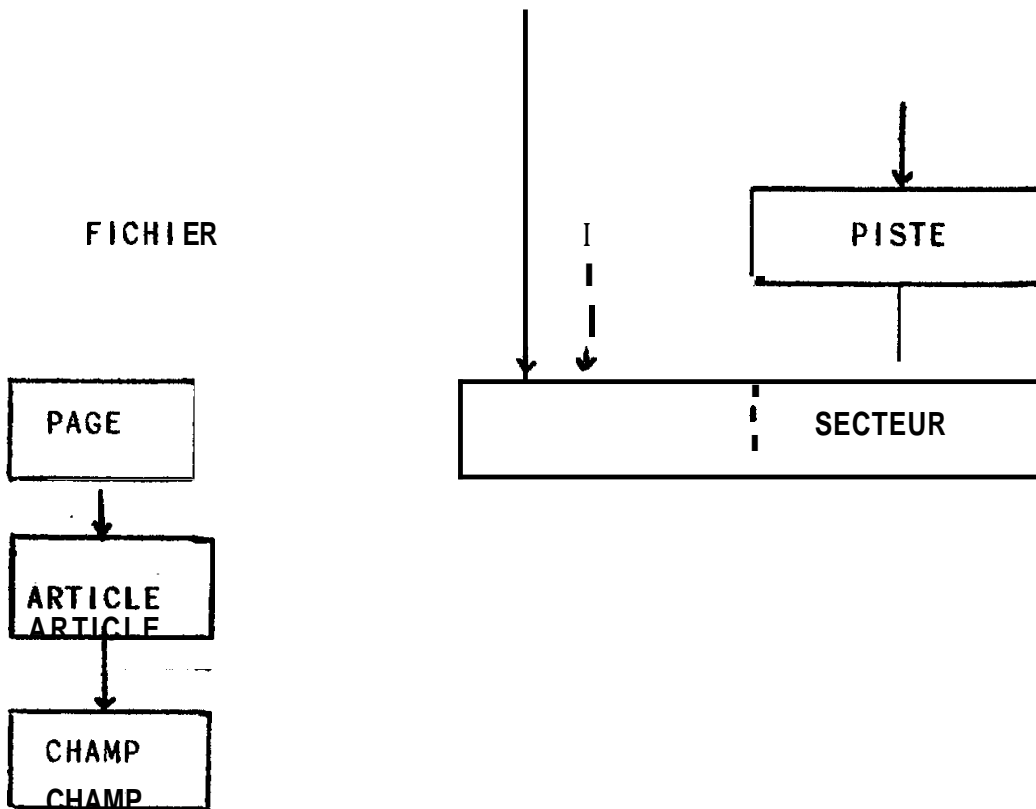


FIG. 1.9

STRUCTURE DE STOCKAGE : MODELE GENERAL AVEC ALLOCATION DYNAMIQUE

## CHAPITRE II

### STRUCTURES DE GESTION DE L'INFORMATION

L'information est l'une des plus importantes ressources d'un système d'exploitation, et le logiciel chargé de la gérer en constitue une partie non moins importante.

Ce logiciel, de conception modulaire, est désigné sous le nom de système de fichiers ou système de gestion de fichiers (SGF). Il est soit fourni en standard avec l'ensemble matériel et logiciel, soit développé pour répondre aux objectifs d'un mode d'exploitation de cet ensemble [ PIR-74 ], [ MUR-72 ], ou l'utilisation d'une technique particulière de gestion de la mémoire secondaire [ KAT-71 ] ou encore, il est développé pour une gamme d'ordinateurs [ STE-67 ], [ SEM-80b ] .

Enfin, certaines études ont abouti à des modèles de systèmes de fichiers qui se proposent d'être généraux [ DAL-65 ], [ MAD-74 ]. Cependant, ni la réalisation ni l'utilisation de ces systèmes n'est maîtrisée; c'est pourquoi, l'étude de leurs propriétés [ ROT-74 ], [ FRI-70 ], [ PAR-79 ], ou la proposition d'un modèle [ CAR-73 ], [ PRA-70 ], correspondent à un besoin toujours d'actualité.

Dans ce chapitre, nous présentons les différentes fonctions des systèmes de fichiers et certaines techniques qui sont utilisées pour les réaliser.

En se basant sur les fonctions d'un SGF, nous proposons un ensemble de propriétés qui permettent une définition fonctionnelle des systèmes de fichiers et de classer les systèmes les uns par rapport aux autres.

## I - LES PRINCIPALES FONCTIONS D'UN SYSTEME DE FICHIERS

### 1.1 - Définition du rôle d'un SGF

Le système de gestion de fichiers est un ensemble de programmes qui permettent à un utilisateur de **créer/** détruire, ouvrir/fermer, lire/écrire, un fichier qui est un 'réceptif' d'informations, identifié par un nom, doté d'une organisation interne et de méthodes d'accès. Ces programmes ddchargent l'utilisateur des problèmes de l'allocation de l'espace en mémoire secondaire, du stockage et de l'accès des informations.

Certains systbmes de fichiers assurent aussi les fonctions de partage et de contrôle d'accès des fichiers.

Un SGF n'est concerné que par la plus simple organisation logique des données **qui** constituent des collections non structurées et non interprktées au niveau **du** système d'exploitation. La structuration et l'interprétation des données sont du ressort des systkmes d'intdrogation de fichiers et des systèmes de gestion de base de données.

Un SGF prdsente deux aspects :

- aspect fonctionnel
- aspect interne

L'aspect interne d'un système concerne **la** structure des ses modules, les tables de gestion des fichiers ouverts, **la** syntaxe de son langage de commandes et l'appel de **ses** primitives.

L'aspect fonctionnel rassemble toutes les informations décrivant les services rendus par le système. C'est cet aspect que nous étudions dans ce chapitre, car il présente un intérêt pour l'utilisateur du fait qu'il permet de comparer les fonctions offertes par le système sans aucune considération de matériel utilisé ou d'environnement d'exploitation.

## 1.2- Fonctions de base d'un SGF

Dans un système de gestion de fichiers il existe quatre fonctions de base. Deux d'entre elles sont liées à la définition même du fichier; les deux autres tiennent de son exploitation.

Les fonctions de définition concernent l'identification des fichiers et leur organisation avec les méthodes d'accès associées.

Les fonctions d'exploitation sont les fonctions d'implantation des fichiers en mémoire secondaire et leurs contrôles.

### 1.2.1 - Identification d'un fichier

Cette fonction regroupe deux fonctions essentielles qui sont le catalogage et la protection des fichiers contre les accès non autorisés.

#### a - le catalogage

Les informations qui définissent un fichier sont maintenues dans une table en mémoire secondaire appelée fichier Catalogue ou Table des Contenus du Volume.

Les systbmes de fichiers présentent différents niveaux de catalogues. Les plus simples n'ont pas de catalogues (CII SIRIS 2, IBM DOS/360) et c'est à l'utilisateur de se souvenir sur quel support sont implantés ses fichiers; d'autres cataloguent un sous ensemble de fichiers (Burroughs MCP, CDC SCOPE 3.1 et 3.4) [KEA-76 ]; d'autres enfin, les plus complbxes cataloguent les fichiers quels que soient leurs status et supports (MULTICS, IBM OS/VS, UNIVAC EXEC 8 ...).

Le catalogage se fait automatiquement (MULTICS, ISM OS/VS ...), ou à la demande de l'utilisateur (IBM OS/360, CII SIRIS 7).

En général, les systbmes le font automatiquement pour les fichiers sur support magnétique à accès direct.

La structure du catalogue peut être simple ou hiérarchique. La hiérarchisation d'un catalogue introduit une classification des fichiers créés par la notion de sous-catalogues et catalogues privés. Cette structuration est utilisée par certains systèmes, tel celui dont le modèle a été propose par DALEY et NEWMANN [DAL-65] pour établir des liens de "synonymie" entre catalogues de fichiers, c'est-à-dire designer dans un catalogue un fichier décrit dans un autre catalogue. Ce qui facilite le partage de fichiers, décrits une seule fois, par plusieurs utilisateurs [FR1-70].

**b - La protection**

L'étude de **la** protection des fichiers contre **les accbs** non autorisés nécessite **la** connaissance des possibilités et techniques de partage des informations du système : Ces deux notions sont en effet indissociables [ FRI-70 ],

Dans un environnement monoprogrammé, **la** notion de partage peut exister uniquement sous une forme statique et la fonction de protection se réduit au contrôle des droits d'accbs de l'utilisateur.

Par contre dans un environnement multiprogrammé, conversationnel ou 'batch-processing', où un grand nombre d'utilisateurs opèrent en commun sur un ensemble de données et de programmes, **la** fonction de partage, si elle existe, peut offrir plusieurs techniques d'accès concurrents aux fichiers : partage statique ou dynamique d'un groupe de fichiers, d'un fichier ou partie de fichier ( une ou plusieurs pages de données); partage par synonymie ou par d'autres moyens [ MUR-72 J, [ PIR-74 ].

Dans un tel environnement **la** fonction de protection doit avoir des mécanismes variés et efficaces pour assurer l'intégrité et **la** confidentialité des informations du système.

Mais si le partage de fichiers pose le problème de leur protection, il offre néanmoins des avantages notables tels que :

- la souplesse d'utilisation des programmes et données du système

- l'amélioration de l'efficacité du système d'exploitation par la réduction du nombre d'entrées-sorties, du fait que les données peuvent être partagées en mémoire centrale

- la synchronisation des opérations sur les fichiers assurant ainsi la cohérence des données

Certains systèmes tels que le CDC SCOPE 3.1 ou l'IBM DOS/360 n'offrent aucune protection.

D'autres en assurent pour un sous-ensemble de fichiers (BURROUGHS MCP). D'autres enfin protègent l'ensemble des fichiers (MULTICS, ..).

Mais, l'efficacité de la protection dépend des mécanismes de protection utilisés.

Parmi les mécanismes nous avons :

- la "privatisation" des fichiers qui fait qu'ils ne sont accessibles que par les programmes de même numéro de compte.

- la limitation du mode d'accès.

- la réservation de l'accès d'un fichier à un ensemble d'utilisateurs associée à un mode d'ouverture (lecture seule, lecture-écriture, ..). Cette technique

peut être généralisée à l'aide d'une matrice de contrôle dont chaque élément (i,j) est le mode d'ouverture autorisé du fichier i pour l'utilisateur j.

- attribution de mot de passe par catalogue, sous catalogue ou fichier.
- coder les informations du fichier ou du chemin d'accès à ces informations (cryptographie).
- combinaison de deux ou plusieurs de ces techniques.

La protection d'un système est dite fine si elle utilise plus de deux techniques.

A titre d'exemple, les systèmes CII SIRIS 7, IBM OS/VS et SEMS MMT2 [ SEM-80b ] offrent deux techniques de protection, MULTICS utilise trois techniques et plus.

Pour améliorer la protection, FRIEDMAN a proposé dans son étude sur la résolution des problèmes de partage [ FRI-70 ] de protéger différemment les informations et les programmes qui assurent la protection et de différencier entre les droits d'accès d'un utilisateur et ceux des programmes qu'il utilise.

### 1.2.2 - Organisation et méthodes d'accès

Cette fonction présente un grand intérêt dans **la** conception et l'utilisation d'un SGF. Contrairement **aux** autres fonctions, elle a souvent fait l'objet d'études [ CHA-69 ], [ HUZ-79 ], [ CAR-73 ] .

Elle regroupe **deux** fonctions :

- l'organisation **de** base du fichier
- son organisation avec les méthodes d'accès associées.

#### 1.2.2.1 - Organisation de base d'un fichier

L'organisation de base comprend deux aspects **qui** sont exclusifs et qui déterminent le type du fichier.

Si l'unité de base dont est constitué le fichier est un caractère, un mot ou un double mot machine, le fichier est une extension de la mémoire centrale et il est dit fichier " plat " ou fichier en " ligne " .

Sinon le fichier est constitué d'articles et dans ce cas, on a le deuxième aspect qui **consiste en la** définition de **la** taille du bloc **de** données. Le fichier est dit " fichier article\* " .

### 1.2.2.2 - L'organisation d'accès

L'échange de données entre le SGF et l'utilisateur se fait en terme d'articles logiques ou de blocs d'articles. Et, c'est le rôle du SGF d'organiser ces blocs ou articles dans le fichier selon la méthode d'accès que l'utilisateur désire appliquer pour accéder à ses données.

En plus de l'organisation séquentielle des articles dans le fichier, les systèmes de fichiers proposent diverses organisations qui réellement se résument en trois catégories :

- L'organisation directe qui permet l'accès à l'article en fournissant son adresse au système. Elle est aussi appelée organisation relative lorsque le fichier n'est pas " plat ", et l'accès à l'article se fait par son numéro logique dans le fichier.
- L'organisation séquentielle indexée qui permet l'accès aux articles soit séquentiellement soit en fournissant au système leur clé d'accès ou identificateur. Pour cette organisation le système constitue à la création du fichier, une table dite table d'index qui contient les clés et les adresses dans le fichier de tous les articles introduits. L'accès séquentiel peut suivre soit la séquence physique des articles soit l'ordre lexicographique des clés dans la table d'index.

Pour améliorer le temps de recherche d'un article, certains systèmes introduisent des modifications sur l'organisation de la table d'index qui peut être à plusieurs niveaux. C'est le cas notamment des méthodes ISAM et VSAM chez IBM [ MAD-74 ], [ KEE-74 ] et de FMS (MMT2).

Enfin certains systèmes (tel que le H.B G 58 ) offrent la possibilité d'indexer un fichier par plusieurs clés. Mais cette méthode s'avère souvent coûteuse car il arrive que la taille de la table d'index dépasse celle du fichier.

- autres organisations : parmi les organisations, on peut citer l'organisation aléatoire où chaque article est identifié par clé et l'association clé-adresse est réalisée à l'aide d'une formule de calcul. Cette organisation est souvent utilisée dans un but de réduire la taille effective du fichier [ ROT-74 ]. Une autre organisation souvent utilisée pour conférer une structure 'bibliothèque' à un fichier, c'est l'organisation partitionnée [ SEM-80b ].

### 1.2.3 - Implantation sur mémoire secondaire

Cette fonction correspond soit à un choix de support, soit à un choix de support plus une allocation d'espace s'il s'agit d'un volume disque pourvu d'un catalogage de fichiers.

Le choix du support de stockage est fait selon la disponibilité **du** matériel, de l'option du système et de la demande de l'utilisateur. La part prise, dans ce choix, par le système peut **être** nulle, partielle ou totale.

L'allocation d'espace en mémoire secondaire à un fichier est l'opération qui consiste à réserver statiquement ou dynamiquement une zone mémoire non nécessairement continue sur un volume disque.

Les systèmes des fichiers offrent trois types d'allocation :

- l'allocation statique qui consiste à réserver une zone de taille fixe à **la** création du fichier.
- L'allocation statique avec increment permet d'allouer un espace de taille fixe appelé racine du fichier et un nombre fini d'incrémentations de même taille si la racine est saturée. Les incréments sont alloués un à la **fois**. Cette méthode est surtout utilisée par les SGF lorsque la taille du fichier n'évolue **pas** d'une manière imprévisible durant leur utilisation (MMT2, CII SIRIS 7, **IBM OS/360**).
- L'allocation dynamique qui s'effectue par bloc mémoire secondaire ou page au fur et **à** mesure des écritures. Cette méthode se trouve dans les systèmes qui gèrent des **fi-**chiers dont la taille ne peut pas être

prévue à l'avance (MULTICS, CTSS , CDC SCOPE 3.4).

#### 1.2.4 - Contrôle de fichiers

Cette fonction traite le contrôle des générations de fichiers (nombre de copies disponibles) et leur connection aux programmes qui les manipulent .

##### Archivage des fichiers

Dans le but de pouvoir récupérer les fichiers à la suite d'incidents qui peuvent survenir durant l'exploitation à cause d'une défaillance matérielle ou une erreur humaine, des copies des fichiers sont archivées sur **des** supports différents de celui supportant la copie originale du fichier.

Cette fonction se caractérise par l'automatisme d'archivage des copies de fichiers et le nombre que le système peut cataloguer .

##### Connection du fichier à un programme

Le contrôle de connection se fait *soit* par langage de commandes et nom interne de fichiers, *soit par* programme. *a* connection par langage de commande se fait *soit* avec gestion du format des articles et facteur de blocage ( CII SIRIS 7, IF OS/360 ), *soit* sans gestion du blocage d'articles (CII SIRIS 2).

La Connection dynamique (par programme) se fait avec un nom interne (BURROUGHS MCP, MMT2), ou un nom de fichier ou segment (MULTICS).

Certains systèmes offrent les deux possibilités de connection. La connection par langage de commandes permet de rendre les programmes totalement ou partiellement indépendants des fichiers exploités.

## 11 - CLASSIFICATION DES SYSTEMES DE GESTION DE FICHIERS

Sil existe des études de comparaison et de classification des matériels [ BEL-71 ], [ FAL -64 ] et des SGBD [ BAR-75 ], rares sont celles qui traitent des systèmes de gestion de fichiers [ PAR-77 ].

Mais ceci ne signifie pas que les logiciels sont équivalents ou que leur étude manquerait d'intérêt. Il suffit de comparer deux systèmes de fichiers tels que ceux de l'IBM OS/VS et de MULTICS ou encore celui du THE MULTIPROGRAMMING [ KEA-76 ], pour constater des différences fondamentales qui existent dans leur conception, leur fonctionnement et leurs objectifs.

Comme pour les matériels ou les SGBD, des descriptions fonctionnelles des systèmes de fichiers permettraient aux utilisateurs de connaître et comprendre les services rendus et aux réalisateurs d'utiliser la somme de ces expériences résultant des réalisations antérieures.

L'objet de ce paragraphe n'est pas de faire une telle étude, mais de décrire certaines qualités fonctionnelles qui permettent de comparer des SGF et d'en étudier les propriétés.

### 11.1 - Définition fonctionnelle d'un SGF

Une telle définition peut être établie en étudiant toutes **les** fonctions élémentaires d'un SGF. Mais cette méthode en **plus** du fait qu'elle soit peu aisée présente l'inconvénient de ne **pas** permettre de le situer par **rapport** aux autres systèmes.

**Les fonctions** élémentaires d'un système de fichiers concourent pour fournir un ensemble de services qui caractérisent ses qualités fonctionnelles.

Une solution consisterait donc à étudier certaines qualités fonctionnelles d'un SGF, tels que l'accès à ses services, ses potentialités d'extension ou d'adaptation, sa capacité et son orientation.

### 11.2 - Présentation de quelques qualités fonctionnelles d'un SGF

#### 11.2.1 - Accès aux services d'un système

Il y a deux moyens d'accès aux services d'un SGF :

- par langage de commandes
- par primitives ou routines utilisables par programme

La simplicité d'usage d'un système est caractérisée par la richesse et l'accessibilité du langage de commande et par **la** diversité des primitives de manipulation **de** fichiers.

Ces primitives sont soit des sous-programmes externes, soit des macro-instructions utilisables en macro-assembleur, soit intégrées dans le langage de programmation.

Selon que le SGF est de type en "ligne" ou "article", l'un ou l'autre de ces moyens est plus favorisé.

### 11.2.2 - Potentialité d'extention

C'est la faculté d'un SGF de s'adapter, dans les limites de son orientation, à des applications particulières et variées sans en changer sa conception. Ceci impose que le système soit de conception modulaire et au besoin paramétrable à sa génération.

### 11.2.3 - Capacité d'un SGF

C'est la mesure de l'intérêt apporté aux utilisateurs par l'ensemble de ses services.

La capacité d'un système n'est pas à confondre avec sa complexité interne qui pourrait être évaluée en nombre d'instructions machine; c'est une notion qui fait appel à l'utilisation de ses fonctions.

Etant donné qu'un SGF est utilisé par différentes applications, sa capacité peut être évaluée par rapport à l'intérêt maximum qu'il offre.

#### 11.2.4 - Orientation d'un SGF

*Les* applications informatiques se répartissent globalement en deux catégories : applications de gestion et applications scientifiques. De même, la conception des SGF reflète cette dichotomie et on rencontre des systèmes de type Gestion et d'autres de types Scientifiques ou en ligne ou plat.

Cela signifie que leurs utilisateurs respectifs ont des besoins différents et que certaines fonctions fondamentales sont spécifiques d'un type de système.

##### Les systèmes de fichiers de type 'gestion'

Les fichiers gérés par ces systèmes sont structurés en articles dont le nombre, pour un fichier, peut être très élevé (quelques milliers); l'accès se fait en séquentiel ou en sélectif sur numéro ou clé d'articles. Le nombre d'intervention sur ces fichiers est souvent très élevé.

Ces considérations font que les SGF laissent à l'utilisateur l'initiative dans l'organisation de ses fichiers afin d'être dans des conditions optimales d'exploitation : temps d'accès, espace mémoire secondaire...

##### Les systèmes de fichiers de type en 'ligne'

Les fichiers gérés par cette catégorie de

systemes, sont des fichiers 'plat', non structures en articles. Ils sont accessibles séquentiellement ou sélectivement sur adresse d'élément; ils ont une organisation identique à celle de la mémoire centrale dont ils sont souvent ('extension.

Leur taille est généralement plus petite que celle des fichiers de type gestion, mais leur nombre peut être élevé si l'environnement d'exploitation est conversationnel. Ce qui nécessite un catalogage et une protection plus sophistiqués que pour le premier type de système. Les utilisateurs, qui sont en général des non-informaticiens, requièrent de ces systèmes une grande simplicité d'usage et un minimum de connaissance sur leur fonctionnement.

Mais, du fait **que** ce sont les systèmes **eux-mêmes** qui choisissent les caractéristiques **des** fichiers et souvent leur implantation, les **programmes** d'application deviennent totalement **dépendants** des fichiers.

Pour ces systèmes de fichiers, le langage de commandes est très simplifié, sinon abandonné **si en plus ils** offrent la connexion dynamique, comme c'est le cas du système CTSS [ CRI-65 ], [ KEA-76 I.

#### Systemes généraux

Ce sont des systèmes qui ont une forte capacité

et: une faible orientation en offrant de nombreuses fonctions élémentaires que l'on rencontre sur les deux types de systèmes de fichiers précédents ( CDC SCOPE 3.4, BURROUGHS MCP, ...). La tendance actuelle est de ne pas réaliser **des** systèmes spécialisés comme on le constate sur des versions d'un même système : CDC SCOPE 3 et 4, IBM OS/360 et OS/VS.

### III - EXEMPLE D'UN SGF : LE FMSZ DU MMT2

Le SGF FMS2 (Files Manager System) assure l'ensemble des services permettant la création et l'exploitation des données sur la gamme de mini-ordinateurs MITRA 125-225-525 (SEMS) sous le moniteur MMT2 [ SEM-80a ], [ SEM-80b ].

FMS2 est l'interface standard des entrées-sorties physiques en mode utilisateur. Il gère tous les périphériques connectables à cette gamme d'ordinateurs ainsi que les fichiers disque et bande. Il permet un fonctionnement temps réel, multi-tâches et multi-fonctions.

Une donnée manipulée par **FMS** doit appartenir à un fichier, qui peut être labellé ou non (cas des fichiers cartes, bandes non labellées, ...).

#### III.1 - identification d'un fichier

Le FMS dispose d'un seul niveau de catalogue dans lequel sont rangés tous les descripteurs de fichiers créés sur disque labellé, c'est-à-dire pourvu d'un **nom** de volume.

En plus du nom, un fichier est identifié par le numéro de compte sur lequel il est créé et qui constitue la première protection, puisque seuls les programmes travaillant sous ce numéro de compte sont autorisés à l'utiliser. Le numéro de compte zéro est commun à l'ensemble des utilisateurs.

Le deuxième niveau de protection est matérialisé par le droit d'accès qu'on attribue au fichier à sa création. Un utilisateur (un programme) désirent utiliser un fichier doit avoir un droit au moins égal à celui de ce fichier.

Le troisième niveau de protection est assuré par le mode d'ouverture permis pour un fichier, soit pour limiter le mode d'accès, soit pour assurer la cohérence des données qu'il renferme dans le cas où il est ouvert par plus d'un utilisateur.

#### 111.2 - Organisation et modes d'accès

Le FMS2 offre quatre types d'organisation de fichiers: séquentielle, directe, partitionnée, séquentielle indexée.

L'organisation de base des fichiers est de type gestion.

L'organisation séquentielle permet un accès séquentiel ou direct sur numéros d'article.

L'organisation directe permet l'accès direct sur adresse de secteur (adresse virtuelle).

Un volume non labellé peut être considéré comme un fichier direct.

L'organisation partitionnée est utilisée sous **MMT2** pour constituer des **bibliothèques** de programmes et de sous-programmes. L'accès **se** fait en mode direct pour accéder à une partition et en séquentiel aux éléments de cette partition.

Dans le cas de l'organisation séquentielle indexée, la table d'index est éclatée en plusieurs niveaux, chaque niveau résumant le niveau inférieur. Le nombre de niveaux est limité à cinq. L'accès ~~se fait~~ soit par clé, soit en séquentiel physique ~~ou logique~~.

### 11.3 - Implantation des fichiers en mémoire secondaire

À la création du fichier, l'utilisateur spécifie le volume sur lequel il désire créer son fichier. Si le volume n'est pas spécifié, **FMS** le crée sur le volume supportant le système d'exploitation.

L'allocation mémoire secondaire peut se faire en statique uniquement ou en statique et increments de taille inférieure à 256 granules et dont le nombre ne doit pas dépasser 62.

L'allocation statique avec increment n'est pas permise pour les fichiers partitionnés dont les partitions sont de taille fixe.

#### 111.4 - Contrôle de fichiers

FMS ne dispose pas de fonction d'archivage automatique. Un utilitaire permet la sauvegarde d'un ou plusieurs fichiers ou de tout le volume.

La connection fichier-programme se fait de deux manières :

- par langage de commandes
- par programme (en dynamique) mais uniquement en macro-assembleur.

Les deux types de connection utilisent un nom interne de fichier ou nom logique de **fichier**.

## CHAPITRE III

### LE SYSTEME DE GESTION DE FICHIERS

Dans un environnement conversationnel multi-utilisateurs, les besoins en mémoire secondaire sont considérables. Durant l'interaction avec le système, les supports d'information détachables ou ne permettant pas un accès sélectif, telles que la carte perforée et la bande magnétique, deviennent sans intérêt pour l'utilisateur.

Mais, si tous les utilisateurs d'un système conversationnel ont la possibilité de conserver en mémoire secondaire à accès direct autant d'informations qu'ils désirent, divers problèmes spécifiques à cet environnement seraient engendrés.

Ces problèmes de gestion concernent essentiellement l'allocation d'espace aux fichiers, leur protection contre les accès non autorisés, leur partage éventuel entre utilisateurs et la performance d'accès aux informations qu'ils renferment.

En général, les systèmes qui offrent un tel environnement d'exploitation ont une structure hiérarchique de fichiers [ RIT-78 ], [ TSM-76 ], [ MAD-69a ] permettant une diversité dans l'utilisation de ces fichiers et de palier aux problèmes engendrés par ce mode d'exploitation.

C'est le cas notamment des systèmes tels que le superviseur TITAN [ WIL-76 ] au le CTSS2, UNIX ou encore le système MULTICS dont la conception reflète le modèle général de système de fichier proposé par Daley et Newman [ DAL-65 ].

Le système d'édition de textes et de préparation de programmes (MISTEP) constitue une machine logique au sens MMT2 [ SEM-80a ], [ BAD-82 ] offrant un environnement conversationnel multi-utilisateurs. Son système de fichiers, que nous présentons dans ce chapitre, conçu selon un modèle de structure hiérarchique de fichiers, permet à des utilisateurs de conserver et de manipuler leurs données et programmes en mémoire dans des conditions d'optimalité d'accès et de stockage, de souplesse et de sécurité.

## I - DESCRIPTION GENERALE

### 1.1 - Objectifs du système de fichiers

Les fonctions essentielles du système de fichiers de MISTEP sont les suivantes :

- enregistrer l'information sur disque de manière qu'elle puisse être rapidement accédée,
- gérer efficacement l'espace disque en évitant les allocations arbitraires,
- protéger l'information contre les accès non autorisés,
- permettre le partage des fichiers,
- assurer la conversion de supports d'information,

- assurer la sécurité des fichiers durant leur ouverture,

Ces fonctions se traduisent par :

- tenir à jour des renseignements sur l'information dans le système, à travers une structure de tables en mémoire centrale et secondaire,
- allouer dynamiquement l'espace disque uniquement à la suite d'une écriture d'une page de données non encore créée,
- maintenir un cloisonnement entre les fichiers en instituant des numéros de compte par groupe d'utilisateurs; la privatisation de fichiers à l'intérieur d'un numéro de compte,
- assumer le prêt de fichiers entre utilisateurs de même numéro de compte ou non; le contrôle d'accès aux fichiers,
- déclarer des fichiers communs à un groupe d'utilisateurs ou à l'ensemble du système.

## 1.2 - Le support d'information

L'unité périphérique, support d'information du système MISTEP, est une unité disque à bandes mobiles, d'une capacité de 50 millions d'octets, connectée sur processeur d'entrées-sorties (IOP).

#### - III.4 -

Au lancement du système MISTEP, une unité disque de la configuration lui est affectée. Les volumes disques qui peuvent être montés sur cette unité doivent avoir subi un prémarquage logiciel par le système à leur première utilisation. Cette opération de prémarquage consiste à créer sur les premiers secteurs du disque un bloc de contrôle contenant le nom symbolique du volume, l'adresse physique du premier fichier des informations de gestion et de statistique sur le volume, ainsi qu'une table binaire appelée BITMAP destinée à gérer les granules du volume.

A l'exception du bloc de contrôle, toute information créée sur le volume disque appartient à un fichier.

Dans sa version actuelle, le système ne peut être utilisé qu'en mode conversationnel; les fichiers sont créés, détruits ou manipulés statiquement par commandes ou dynamiquement à l'aide de procédures, fournies en standard, appelables par les tâches qui s'exécutent sous le système MISTEP.

#### 1.3 - Les concepts de base du système

Dans le contexte du système MISTEP, le terme 'utilisateur' désigne une personne ou une tâche.

Le concept d'utilisateur est rigoureusement défini par un nombre fixe de composantes et toute synonymie ou homonymie est écartée.

Un fichier est une séquence ordonnée d'éléments appelés page; une page contient un nombre entier d'articles.

Tout fichier est décrit dans un catalogue. Les fichiers gérés par le système sont de type fichier-article et ils sont formatés.

#### 1.4 - Le catalogage

Un catalogue est un fichier particulier géré par le système de fichiers et contenant une liste de points d'entrée dont chacune correspond à la description d'un fichier ou d'un catalogue.

Du point de vue de l'utilisateur, un point d'entrée dans un catalogue est un fichier accessible par un nom symbolique.

Toutefois un point d'entrée peut pointer vers un autre point d'entrée dans le même catalogue ou dans un autre.

Une branche est une entrée d'un catalogue qui pointe directement un fichier; un ~~lien~~ est une entrée qui pointe vers une entrée de catalogue.

À l'exception des liaisons qui peuvent constituer des boucles, un lien peut éventuellement pointer une **branche** et donc un fichier. Cette liaison peut se faire directement ou à travers une chaîne de liens.

De ce fait, lien et branche peuvent effectivement pointer un même fichier.

En général, l'utilisateur est moins concerné par la nature de liaison que par la facilité d'accéder à son fichier.

Une branche contient la description du fichier qu'elle pointe. La seule information d'une entrée de type lien est le numéro composé de l'entrée pointée c'est-à-dire, le numéro du catalogue contenant l'entrée et le numéro de l'entrée dans ce catalogue.

## II - STRUCTURE HIERARCHIQUE DE FICHIERS

### 11.1 - Modèle général

D'une manière **générale**, une structure hiérarchique de fichiers peut être vue comme une arborescence de fichiers et de catalogues, munie d'une racine à laquelle arrive une branche fictive seule connue du système de fichiers.

La hiérarchie d'une telle structure est définie comme suit :

- un fichier pointé à partir d'un catalogue est immédiatement inférieur à ce catalogue (et le catalogue est immédiatement supérieur au fichier).
- un fichier immédiatement inférieur à un catalogue qui lui-même est inférieur à un autre catalogue, est inférieur à ce catalogue (d'une manière similaire, le second catalogue est supérieur au fichier).
- la racine de l'arborescence a le niveau zéro et les fichiers qui lui sont immédiatement inférieurs ont le niveau un. Par extension, 'l'infériorité' ou 'la supériorité' d'un fichier par

rapport à un catalogue est définie par le nombre de niveaux qui les séparent de la racine sur la même déviation (fichier et catalogue appartiennent au même chemin partant de la racine).

Dans une telle structure, il est donné à l'utilisateur de travailler sur plusieurs fichiers appartenant à des catalogues différents afin de lui éviter de passer continuellement d'un catalogue à l'autre. Ainsi une telle structure, permet aux utilisateurs de partager des fichiers écrits dans des catalogues différents.

Un exemple de structure est donné par la figure 11.1. Sur cette figure, les catalogues sont représentés par des cercles et les noeuds terminaux sont des fichiers et ils sont schématisés par des triangles. Chaque trait qui part d'un noeud non terminal est une entrée de catalogue. Ces entrées sont désignées par des lettres (A, B, C, D, E, F, X). On constate que la lettre X identifie plus d'une entrée. Cela signifie que plusieurs fichiers écrits dans des catalogues différents peuvent avoir le même nom symbolique. Et, pour désigner un fichier, il faut en plus de son nom, définir toute sa filiation en partant de la racine. Par exemple le nom symbolique indiqué par un 1 est: **A.B.C.**

A partir de ce catalogue on peut atteindre les deux fichiers écrits dans les entrées D et E et dont les noms composés sont respectivement **A.B.C.D** et **A.B.C.E**.

En pratique, l'utilisateur ne donne pas systématiquement et explicitement le nom composé du fichier qu'il réfère; celui-ci est obtenu par le système en concaténant logiquement les noms d'entrées qui constituent son chemin d'accès.



STRUCTURE DE FICHIERS EN SYSTEME MISTEP

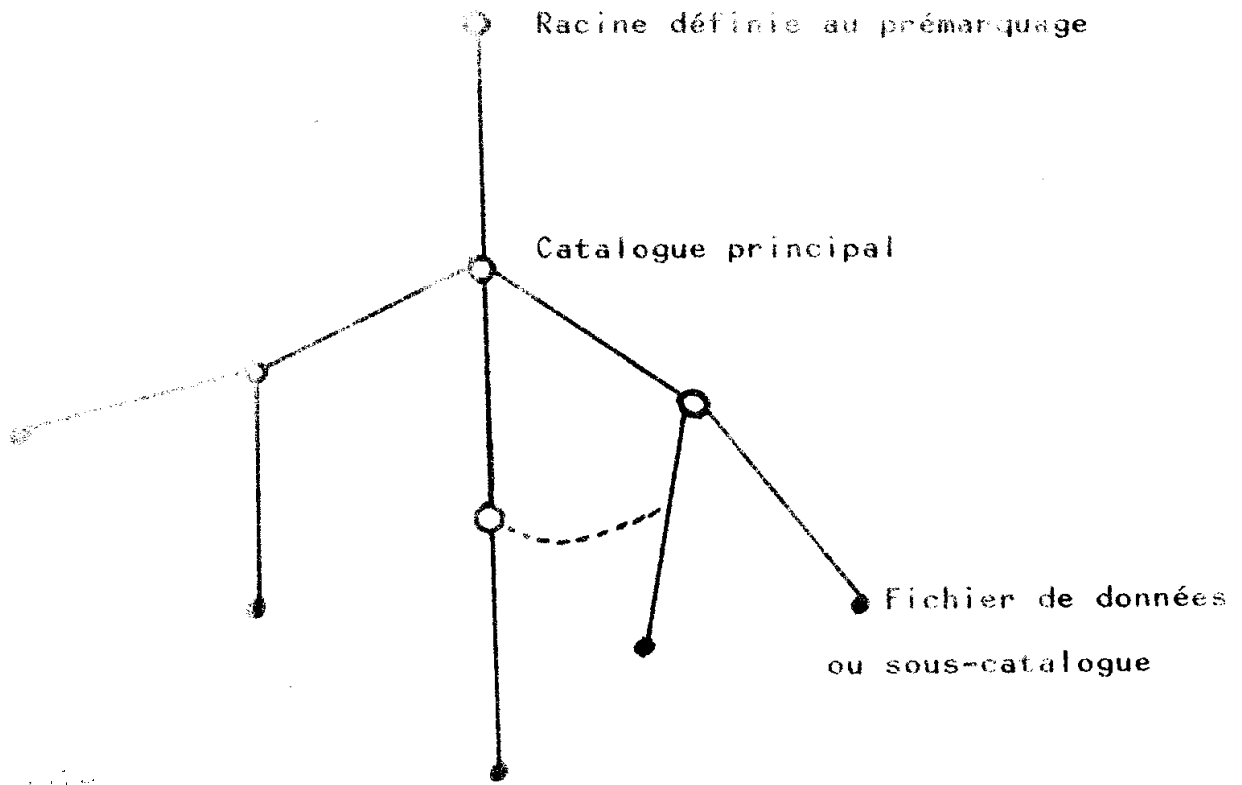


FIG. III.24

OU

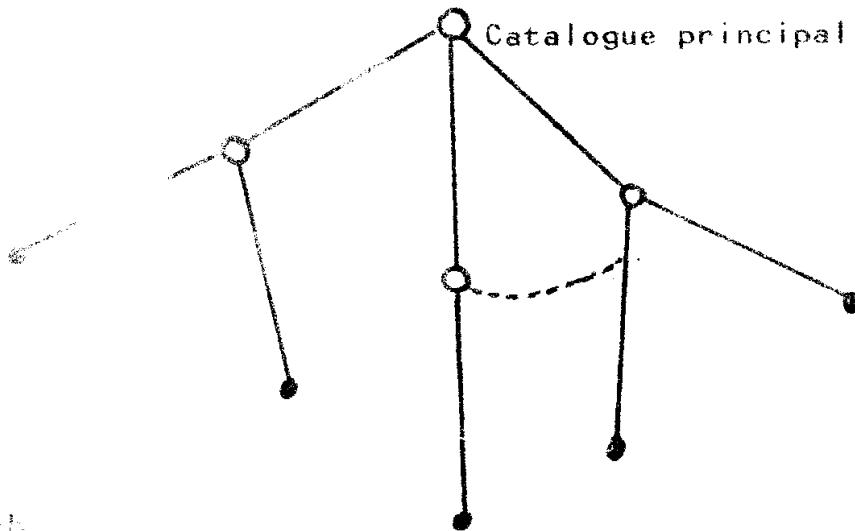


FIG. III.25

11.3.2 - Description des différents éléments de  
la structure

Le catalogue principal

C'est un fichier formé d'une racine appelée dictionnaire et d'un certain nombre de partitions.

A tout numéro de compte correspond une partition destinée à contenir les descripteurs des utilisateurs de ce numkro de compte.

**La** racine est formée d'un nombre d'entrées égal à celui des partitions et dont chacune décrit un numéro de compte selon **la** structure suivante :

- Numkro de compte                    4 octets
- Adresse de **la** partition  
correspondante                    3 octets
- Adresse du fichier ca-  
talogue secondaire as-  
socié au numéro de  
compte                                3 octets
- Nombre de pages allouées  
au numéro de compte            2 octets
- Nombre de pages utilisées        2 octets

Chaque point d'entrée d'une partition décrit un utilisateur selon **la** structure :

- Nom utilisateur	20 octets
- Mot de passe	04 octets
- Dernière date de modification ou de <b>consultation</b>	03 octets
- Numéro de l'entrée dans le catalogue secondaire correspondant au premier fichier de l'utilisateur	02 octets
- Droit utilisateur et qualification	01 octets

#### Le catalogue secondaire

C'est un fichier alloué à un numéro de compte et dont les articles sont des descripteurs de fichiers. La longueur d'un article élémentaire correspond à la description d'un fichier séquentiel, elle est égale à **24** octets. **Dans** le cas de fichier non séquentiel le descripteur s'étale sur deux articles élémentaires. Le premier octet de l'article indique **si** celui-ci est libre ou alloué.

Le descripteur de fichier se compose de cinq catégories d'informations qui sont :

- identification du fichier
- organisation
- implantation en mémoire secondaire
- contrôle
- informations supplémentaires

La description de chaque catégorie est donnée en annexe.

#### Le sous-catalogue

Un sous-catalogue a une structure de catalogue secondaire. Il est destiné à décrire des fichiers d'un seul utilisateur ou des fichiers communs à tous les utilisateurs du numéro de compte auquel il appartient.

Ainsi, un utilisateur peut décrire les fichiers d'une application dans un sous-catalogue du fait qu'ils sont tous ouverts en même temps.

Un sous catalogue est décrit comme un fichier ordinaire par un article élémentaire du catalogue secondaire associé au numéro de compte.

#### II.4 - Manipulation de catalogues

La création d'un numéro de compte est faite par le gestionnaire ou administrateur du système auquel il associe un catalogue et un administrateur. Le rôle de l'administrateur de numéro de compte, consiste à créer ou à supprimer un utilisateur. Les administrateurs du système et des numéros de compte ont des qualifications différentes de celles des utilisateurs ordinaires.

#### Catalogue particulier

Pour tout volume, il existe un catalogue secondaire

particulier dit catalogue public associé au numéro de compte :%PUB. Ce numéro de compte n'a pas d'administrateur, et aucune partition utilisée sur du catalogue principal ne lui est affectée. Il est commun à l'ensemble des utilisateurs du système, avec la particularité que les fichiers créés sous ce numéro de compte ne sont pas protégés et ils peuvent être détruits par le système *si* ils ne sont pas ouverts durant plus de quinze jours (cette durée est fixée selon la période de sauvegarde de l'ensemble des fichiers qui est faite chaque semaine).

#### Sous-catalogue commun

Il est décrit dans un catalogue secondaire non public. Tout fichier décrit dans ce sous-catalogue est commun à l'ensemble des utilisateurs du numéro de compte.

Tout fichier qui n'appartient pas à un catalogue public ou sous-catalogue commun est dit privé et il ne peut être accédé que par son propriétaire ou par des utilisateurs autorisés par ce dernier.

La figure 111.3 donne un modèle de structure de fichiers du système.

### 111 - IDENTIFICATION ET CONTROLE D'ACCES

#### 111.1 - Qualification d'un utilisateur

La qualification d'un utilisateur est le droit qu'il est donné pour accéder aux différents fichiers et catalogues.

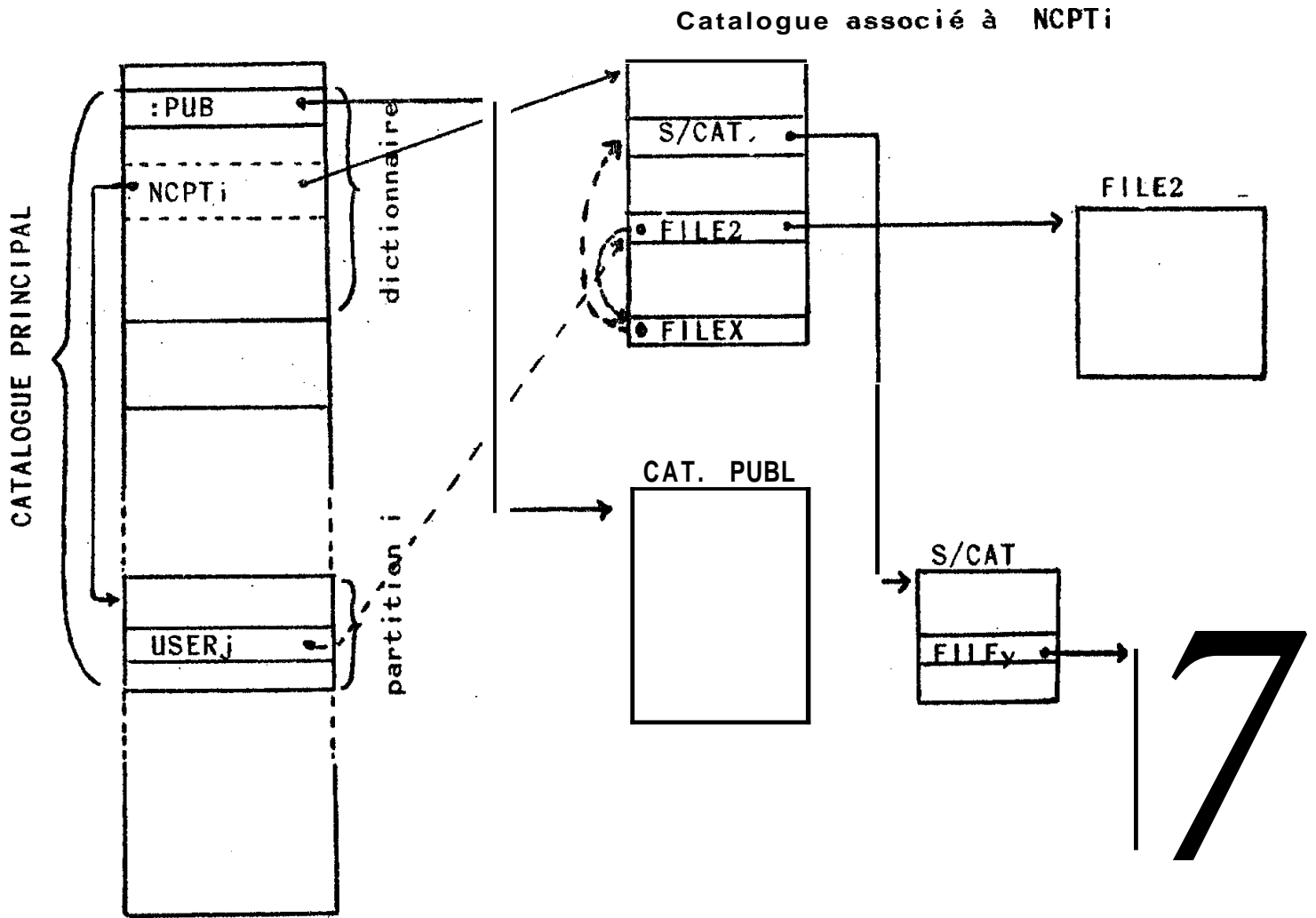


FIG. III.3

UN MODELE DE STRUCTURE DE FICHIERS SOUS MISTEP

Il y a quatre qualifications:

#### Qualification 0

Elle est implicitement attribuée au VISITEUR qui ne peut travailler que sur le numéro de compte : **PUB**.

A la fin de la session, ses fichiers sont détruits.

#### Qualification 1

Un utilisateur ordinaire a cette qualification, qui lui donne le droit de créer des fichiers sous son numéro de compte ou dans le catalogue commun. Il ne peut accéder qu'à ses fichiers et à ceux pour lesquels il est autorisé.

Un utilisateur de qualification supérieure ou égale à 1 appartient nécessairement à un numéro de compte.

#### Qualification 2

C'est la qualification des utilisateurs administrateurs des numéros de compte. En plus des droits d'un utilisateur ordinaire, elle leur donne le droit de créer ou détruire un utilisateur de leur numéro de compte.

#### Qualification 3

Un utilisateur muni de cette qualification est appelé administrateur de la base de données. Il est créé au prémarquage d'un volume et il a tous les droits.

### III.2 - Identification d'un utilisateur

Au lancement d'une session de travail, l'utilisateur s'identifie en fournissant son numéro de compte et son **nom** au système, qui procède à la vérification de leur validité. Si ces deux informations figurent dans le catalogue principal (ils sont respectivement dans le dictionnaire et dans la partition associée au numéro de compte), le système demande à l'utilisateur le mot de passe, qu'il vérifie avec celui **du descripteur-utilisateur**; si les deux mots de passe sont différents, l'accès au système lui est refusé.

À l'admission de l'utilisateur, son descripteur est chargé en mémoire centrale où il est maintenu durant toute la session. Ainsi, une partie du chemin d'accès à ses fichiers est constituée.

#### Cas particulier

Si un utilisateur n'appartient à aucun numéro de compte, il est identifié par le nom commun VISITEUR avec des droits très restreints, et il ne peut accéder qu'aux fichiers de numéro de compte public.

### III.3 - Identification d'un fichier

Un fichier est identifié à l'aide d'un chemin d'accès formé de branches et de liens partant de la racine et aboutissant à un noeud terminal de la structure.

**Les** différentes informations qui permettent de constituer ce chemin sont :

- le numéro de compte de l'utilisateur,
- **son** nom,
- éventuellement, le nom d'un sous-catalogue,
- le nom symbolique du fichier.

Le numéro de compte et le nom sont fournis à l'initialisation de la session, l'utilisateur ne **spécifie** que le nom du fichier et, si celui-ci appartient à un sous-catalogue, le nom de ce sous-catalogue.

Exemple : DEL FICHER1

Ce qui **signifie** : détruire le fichier de nom FICHER1, décrit dans le catalogue courant et appartenant à l'utilisateur qui **a** initié **la** session.

#### Cas particulier

Si un fichier n'appartient pas au catalogue courant, l'utilisateur doit spécifier le **nom** du sous-catalogue ou du catalogue commun ou privé.

Exemple :

- 1° DEL FICHER1
- 2° DEL FICHER1 :PUB
- 3° DEL FICHER1 :COM
- 4° DEL FICHER1 :CATPRIV

Chacune de ces commandes signifie :

- 1° - Détruire le fichier de nom FICHIER1 décrit dans le catalogue courant (c'est-à-dire celui associé au numéro de compte de l'utilisateur).
- 2° - Détruire le fichier de nom FICHIER1 décrit dans le catalogue public.
- 3° - Meme action que la précédente sur le fichier FICHIER1 appartenant au sous-catalogue commun du numéro de compte courant.
- 4° - Détruire le fichier FICHIER1 du sous-catalogue :CATPRIV du catalogue courant.

#### III.4 - Partage de fichiers

La modèlè de structure, en plus des fichiers publics et communs, permet à des utilisateurs de se partager des fichiers prives en accordant des prêts de fichiers entre eux.

Le prêt de fichier est matérialisé au niveau de la structure, par la creation dans le catalogue du bénéficiaire d'un point d'entrée particulier. Ce point d'entrée est chaîné aux autres fichiers de l'utilisateur et pointe, par une liaison de type lien, le descripteur effectif du fichier prêté.

Pour éviter que le fichier emprunté ait le même nom symbolique qu'un autre fichier de l'emprunteur, celui-ci peut le désigner par un autre nom que celui sous lequel il est désigné dans le catalogue du prêteur.

## - III.20 -

L'opération de prêt est initiée par le propriétaire du fichier à l'aide de la commande PERMIT suivie des paramètres suivants :

- numéro de compte et nom du bénéficiaire
- nom sous lequel le fichier est désigné
- nom et catalogue du fichier prêté
- mode d'ouverture permis

Exemple

```
PERMIT :UST USER1 FICHER3 FICHER5 :CATPRIV
Le fichier de nom FICHER5 décrit dans son catalogue
:CATRIV est prêté à l'utilisateur de numéro de compte
:UST et de nom USER1 sous le nom FICHER3.
```

### III.5 - Contrôle d'accès

A l'initialisation d'une session de travail, l'identification de l'utilisateur est nécessaire pour deux raisons :

- comptabilisation des services rendus à l'utilisateur,
- vérification des droits d'accès au système, et établissement dans certaines structures de fichiers, d'une portion du chemin d'accès au fichier.

Le mode de structure de fichiers adopté pour notre système de fichiers permet deux approches du contrôle d'accès.

La première approche considère la structure essentiellement ouverte avec un mode d'accès initialement sans restriction et, tout accès subséquent aux catalogues et fichiers de la structure, est permis à moins que cela ne soit spécialement refusé.

La deuxième approche considère la structure initialement fermée et l'accès est restreint aux catalogues et fichiers privés après vérification de certaines clés d'accès, et toute ouverture de catalogue ou de fichier est refusée à moins qu'il y ait une permission du propriétaire du fichier ou catalogue.

Ces deux approches constituent deux vues extrêmes de la protection et sont applicables dans deux environnements différents.

La première approche peut être efficace dans un environnement où la structure de fichiers schématise une base de données commune à l'ensemble des utilisateurs du système, et seules certaines de ses parties sont réservées.

Par contre, la seconde approche intéresse un contexte dans lequel la base de données est partitionnée et chaque partie appartient à un utilisateur qui est seul habilité à autoriser l'accès aux informations de sa partition.

Le principe du contrôle d'accès du système MISTEP est basé essentiellement sur la seconde approche tout en permettant qu'il y ait des catalogues et fichiers communs.

### Les différents niveaux du contrôle d'accès

La structure des fichiers est conçue de manière à assurer un cloisonnement entre les utilisateurs du système tout en leur permettant l'accès à des fichiers communs et se partager des fichiers.

Ce cloisonnement est matérialisé à trois niveaux de la structure :

- au niveau du catalogue principal : seuls les utilisateurs d'un numéro de compte peuvent ouvrir le catalogue correspondant.
- au niveau de l'utilisateur : un utilisateur ne peut accéder qu'à ses fichiers et à ceux pour lesquels il est explicitement autorisé.
- au niveau du catalogue d'un numéro de compte : la notion de sous-catalogue permet d'affiner le contrôle puisqu'elle ajoute une protection supplémentaire.

En plus de cette protection passive assurée par le cloisonnement, il existe deux autres mécanismes de protection permettant de valider le chemin d'accès à un fichier et de contrôler le mode d'ouverture permis. Ce sont le mot de passe et le mode d'accès autorisé pour un fichier,

Le contrôle du mot de passe est fait à l'identification de l'utilisateur,

Pour permettre à un utilisateur de savoir si son mot de passe a été 'découvert' le système actualise la date de sa dernière consultation qui est rangée dans le descripteur de l'utilisateur.

Le mode d'accès à un fichier est défini à sa création ou la modification de ses paramètres. Mais cette mesure de protection ne concerne que le propriétaire du fichier, qui peut s'imposer un mode d'accès et cela par mesure de sécurité.

Au prêt d'un fichier, le propriétaire définit le mode d'ouverture permis au bénéficiaire.

Les différents modes d'accès sont :

- lecture seule (L)
- écriture (mise-à-jour) (E)
- modification des paramètres du fichier (M)
- tout mode (A)

#### IV - ALLOCATION ET GESTION DE L'ESPACE DISQUE

##### IV.1 - Allocation de l'espace disque

L'allocation de l'espace disque se fait de trois façons :

- 1° - allocation statique de tout l'espace disque spécifié à la création du fichier (cas de l'IBM OS/360).
- 2° - allocation d'une racine en statique et d'un nombre d'incréments à la demande. Le nombre et la taille des incréments sont fixés à la création du fichier (15M OS/360, MMT2 FMS).
- 3° - allocation dynamique à l'écriture ou à la demande (MULTICS, IBM TSS/360, MTS ) I PIR-75 I.

Chacune de ces méthodes d'allocation a ses avantages et ses inconvénients, et le choix de l'une d'elles dépend essentiellement de l'environnement d'exploitation et du matériel utilisé.

Avantages et inconvénients de deux premières méthodes

La taille d'un fichier est souvent choisie d'une manière arbitraire. Ceci soit parce que l'utilisateur est un débutant ou un non initié (non informaticien), il ne sait donc pas évaluer la taille de son fichier, soit que l'utilisateur ne peut pas évaluer à priori l'espace disque nécessaire à son fichier.

La première méthode d'allocation peut engendrer l'une des deux situations suivantes :

- la taille du fichier choisie s'avère insuffisante à l'exploitation et elle doit être redéfinie.
- la taille du fichier est surévaluée et l'espace disque alloué n'est pas entièrement occupé, ce qui fait baisser le taux d'utilisation de tout l'espace **disque**.

L'allocation statique avec incrément permet dans certains cas d'éviter ces deux situations. Mais son efficacité reste relative à des applications pour lesquelles on peut prévoir toutes les éventualités de demande d'espace,

L'avantage de ces deux méthodes d'allocation est qu'elles nécessitent que très peu d'informations de gestion de l'espace alloué à un fichier. Seules l'adresse de début et la taille du fichier sont mémorisées dans le cas de l'allocation statique, et l'adresse et la taille de la racine et de chaque incrément dans le cas de la deuxième méthode.

L'espace alloué aux fichiers est continu sur disque, il est donc situé sur des pistes et cylindres contigus. Ce qui permet un accès rapide aux blocs de données lorsque le fichier est exploité en séquentiel.

Un autre inconvénient de ces deux stratégies d'allocation est la fragmentation de l'espace disque, surtout si les opérations d'allocation et de désallocation d'espace sont trop **fréquentes**, créant un nombre élevé de trous dont les tailles ne permettent pas souvent d'y ranger d'autres fichiers, et entraînant ainsi des demandes d'allocation qui ne sont pas satisfaites alors que l'espace disque est loin d'être saturé.

La fragmentation de l'espace disque est accentuée si l'environnement d'exploitation est conversationnel **multiutilisateurs** destiné à la mise au point de programmes et d'applications non opérationnelles.

Les systèmes qui utilisent cette stratégie d'allocation ont des techniques de gestion des zones libres, soit pour les allouer ou pour les compacter afin de satisfaire une demande d'espace qui n'a pu l'être par l'allocation d'une zone ou plus.

L'une de ces techniques consiste à chaîner les zones libres entre elles en mentionnant leur taille en son début. Ainsi, pour satisfaire une demande d'allocation, il suffit de parcourir la chaîne de zones et voir si l'une d'elles satisfait à la demande.

Comme pour la mémoire centrale, il existe des algorithmes d'optimisation du temps de recherche et de l'espace à allouer [CRO-75] .

Mais il arrive qu'une demande ne puisse être satisfaite par l'allocation de la plus grande zone libre. Dans ce cas le système a recours au compactage pour avoir un espace libre continu et satisfaire ainsi la demande d'allocation.

Mais cette solution est coûteuse en temps de traitement et d'entrées-sorties du fait que chaque déplacement d'un bloc se solde par une lecture et une écriture en mémoire secondaire, ainsi qu'une mise-à-jour des tables de gestion des fichiers implantés entre les zones libres à concaténer.

#### IV.1.2 - Avantages et inconvénients de l'allocation dynamique

Le principe de base de l'allocation dynamique est de permettre l'évolution de la taille de fichier durant l'exécution du programme qui l'utilise en lui allouant des zones mémoire secondaire de même taille appelées blocs physiques (ou pages selon la vue logique Cf. chapitre I ),

L'allocation est faite à chaque écriture d'article dont la page qui le contient n'est pas encore créée dans le fichier.

Contrairement aux deux précédentes méthodes d'allocation, l'allocation dynamique résout efficacement les problèmes de déficit et de surplus de mémoire secondaire allouée au fichier : la taille du fichier n'est **pas** fixée à sa création.

Les blocs alloués ne sont pas nécessairement contigus et leur taille dépasse rarement **4096** octets **IBM-76**, entraînant un nombre élevé de 'trous'. Mais comme la taille du bloc, exprimée en nombre de granules, varie selon les fichiers, tout l'espace disque est utilisable et on n'arrive pratiquement jamais à une situation de saturation alors que l'espace disque n'est pas totalement et effectivement occupé.

La première conclusion que l'on peut tirer **sur** cette méthode d'allocation est qu'elle permet une utilisation rationnelle de l'espace disque sans nécessiter des opérations de compactage, ou de recherche de **la** zone libre qui permet l'allocation la **plus** optimale.

Toutefois, cette stratégie d'allocation présente deux inconvénients qui sont :

- la taille de **la** table de gestion est relativement importante.
- les blocs physiques d'un fichier sont éparpillés sur disque.



Fig. 111.4

Table des blocs ou table des pages

#### IV.1.3 - Gestion de l'espace alloué à un fichier

Etant donné que la **mdmoire** secondaire est **al-**  
louée au fichier par 'morceau' **ii faut** pouvoir  
garder trace de leur adresse et de leur ordre  
logique dans le fichier.

Deux techniques **sont** souvent utilisées :

- chainage des blocs entre eux : l'adresse  
d **blocs**  $i-1$  et  $i+1$  sont rangées dans le  
bloc  $i$ .
- ranger les adresses de **blocs** dans une table  
constituant ainsi un chainage logique entre  
eux : l'élément  $i$  de la **table** contient l'ad-  
resse physique du bloc de numéro logique  $i$ .  
( Fig. 111.4)

La première méthode de **chainage** est avantageuse  
si le fichier **est** destiné à être utilisé en  
accès séquentiel mais est inefficace si l'accès  
**se** fait en mode direct ( par index , numéro ou  
calcul d'adresse), **car** l'accès à un bloc donné  
nécessite la lecture de tous les blocs **qui le**  
précèdent logiquement pour pouvoir récupérer son

adresse physique.

La technique de gestion par table des pages permet d'obtenir l'adresse physique d'un bloc donné, en consultant uniquement cette table qui est en plus gardée partiellement ou totalement, selon sa taille, en mémoire centrale.

## IV.2 - Stratégie d'allocation du système de fichier de MISTEP

### IV.2.1 - Organisation de l'espace disque

L'espace disque est subdivisé en granule de **512** octets correspondant à un bloc ou page de données élémentaire.

L'adresse d'un granule est une adresse physique paire d'un secteur de 256 octets qui constitue la subdivision physique du disque.

Les granules ou pages élémentaires sont **gérés à** l'aide d'une table binaire, appelée BITMAP, dont chaque élément définit l'état libre ou occupé d'une page. Cette table est maintenue, partiellement en mémoire centrale.

A chaque allocation d'un bloc, la **BITMAP** est consultée par bloc de 16 bits pour y chercher un nombre de granules **libres** et contigus pour **satisfaire** la demande.

Afin d'améliorer l'efficacité de cette méthode de gestion des granules du disque, on utilise une technique dite 'technique du packing' qui consiste à indexer des groupes de bits de la **BITMAP**.

Chaque groupe de bits est indexé par un bit qui définit son **état** 'totalelement' libre ou non.

Si bit = 0 alors tous les bits du groupe **sont** libres

Sinon il existe au moins un bit occupé.

Les index sont organisés en une table de niveau supérieur à la BITMAP. Et, de manière récursive, on peut constituer d'autres tables sur lesquelles est appliquée **la même** technique.

Dans notre étude des stratégies **d'allocation**, nous avons imaginé une amélioration de la 'technique du packing' en instituant **la** notion de classe de fichiers; et à chaque classe on fait correspondre des groupes de bits. Ces classes de fichiers sont caractérisées par **la** taille, exprimée en granules, des blocs physiques de données.

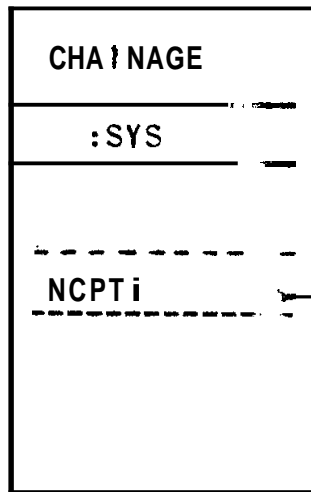
Ainsi, les groupes de bits associés à une classe **de** fichiers doivent être formés d'un nombre de bits permettant de satisfaire une demande d'allocation provenant d'un fichier de cette classe.

#### IV.2.2 - L'allocation d'espace

L'allocation d'espace à un fichier est dynamique. **Elle** est faite par bloc physique formé d'un nombre entier de granules **contigus**. Ces blocs sont **gérés** dans une table dite **table** des pages.

Cette table de taille dynamique, est constituée de pages élémentaires chaînées entre elles dans les deux sens.

PAGE 1



Catalogue secondaire

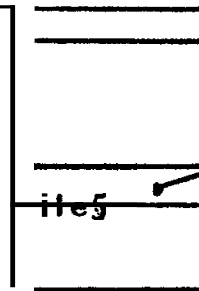
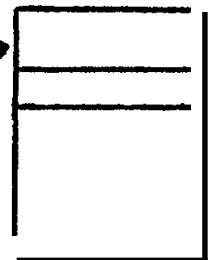


TABLE DES PAGES



PAGE 2

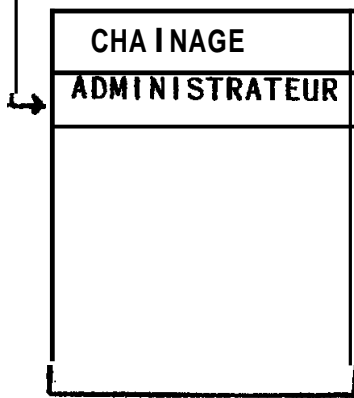


FIG. 111.5

STRUCTURE DU CATALOGUE PRINCIPAL

La taille **maximale** d'un bloc est fixée à 4096 octets. Au delà cette limite, les granules alloués au bloc sont groupés en sous bloc de taille inférieure ou égale à **4096** octets. Dans ce cas particulier le bloc est appelé segment et on dit qu'il est formé de blocs(ou sous-bloc).

#### IV.2.3 - Creation de fichier

La création de fichier se fait de deux manières selon que le fichier est de type catalogue ou de type données.

##### Fichier catalogue

La creation d'un fichier catalogue consiste en l'allocation d'un point d'entrée dans le catalogue, qui lui est immédiatement **supérieur**, et d'une page Blkmentaire.

Les pages d'un catalogue sont chaînées entre elles.

Le catalogue principal qui est créé **au** prémarquage du volume, est formé de pages de 1024 octets.

**A sa** création, deux pages lui sont **allouées**, l'une pour la partie dictionnaire, et qui contient le numéro de compte ;SYS, la seconde pour la partition liste des utilisateurs de ce numéro de compte et **qui** contient le descripteur de l'administrateur du système (Fig. III.5).

##### Fichier données

La creation d'un fichier données consiste à lui attribuer un point d'entrée dans le catalogue ou

sous-catalogue de son propriétaire s'il existe, et lui allouer une **page** élémentaire qui constitue la première page de la table des pages (Fig.III.6).

## V - ORGANISATION ET METHODE D'ACCES

Les fichiers gérés par le SGF du système MISTEP sont de type fichiers articles.

Les différentes organisations de fichiers possibles sont :

- L'organisation séquentielle
  - \* simple
  - \* partitionnée
  
- L'organisation séquentielle indexée
  - \* clé unique
  - \* **clés multiples**
  
- L'organisation partitionnée
  
- L'organisation libre ou device

A chacune de ces organisations correspond un ou plusieurs modes d'accès.

### V.1 - L'organisation séquentielle simple

Les articles d'un fichier séquentiel sont enregistrés consécutivement **selon** leur ordre d'arrivée ou selon leur ordre logique du numéro indépendamment de l'ordre chronologique.

**De même,** l'accès à ces articles peut se faire en mode séquentiel ou direct.

CATALOGUE SECONDAIRE

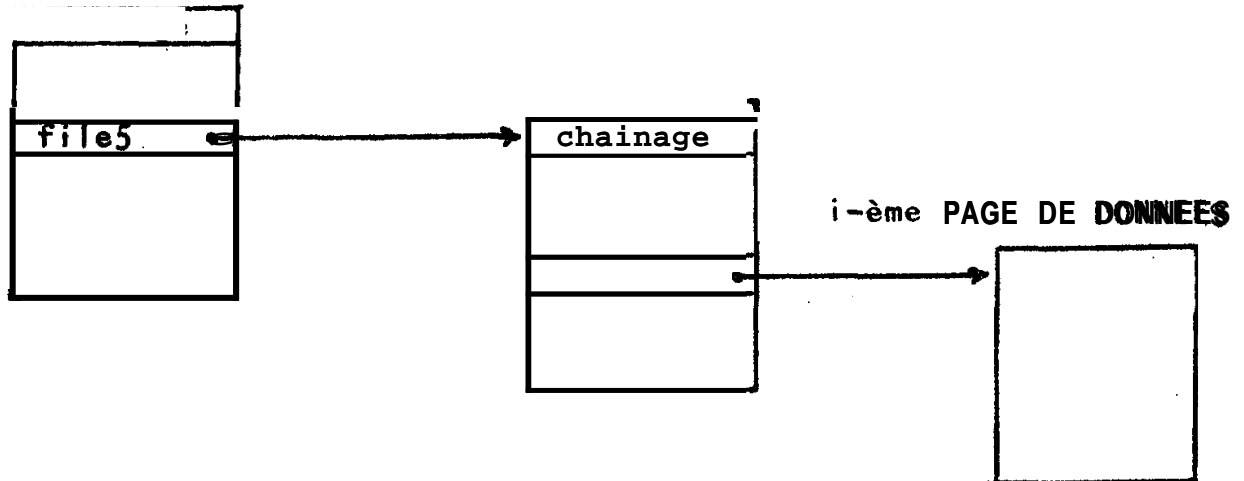


FIG. 111.6

LIAISONS CATALOGUE-FICHER,  
TABLE'DES PAGES-PAGES DE DONNEES

La structure et la longueur des éléments de la table des pages dans le cas des fichiers séquentiels, dépend du format d'article.

#### Format fixe

Un élément décrivant un bloc de données est constitué de 3 octets tels que :

bits (0 - 16) contiennent l'adresse physique du bloc

bits (17 - 21) nombre de granules par bloc

bit 22 Btat du bloc (alloué ou non)

bit 23 indique le début d'une partition

#### Format variable

L'une des particularités du système de fichiers que nous présentons est l'accès direct à des articles de longueur variable.

Pour réaliser ce mode d'accès, des informations supplémentaires sont rajoutées dans la table des pages.

En supposant que les blocs ont une structure de segment (c'est-à-dire le cas le plus défavorable), ces informations permettent de localiser le segment où se trouve le bloc (ou sous-bloc) contenant l'article puis localiser ce bloc. Après cela, l'article est recherché séquentiellement dans le bloc.

Ces informations sont :

- nombre d'articles par bloc 2 octets

- adresse du début du premier

article dans le bloc 2 octets

Si le segment est formé d'un seul bloc (donc de taille inférieure ou égale à 4096 octets), ces deux octets ne sont pas réservés.

A fin de pouvoir localiser la page de la table des pages qui décrit le segment recherché, on réserve dans chaque page de cette table :

- 2 octets pour le nombre d'articles contenus dans les segments décrits dans les pages précédentes.

- 2 octets pour le nombre d'articles contenus dans les segments décrits dans la page.

V.2 - L'organisation séquentielle partitionnée

Les articles d'un fichier séquentiel partitionné sont de longueur fixe. Chaque partition commence au début d'un bloc ou segment. Le bit 23 de l'élément qui décrit le bloc ou segment indique si c'est un début de partition ou non. La fin d'une partition est repérée par une marque "fin de partition". La méthode d'accès à ce type de fichier est l'accès séquentiel avec possibilité de sélection de partitions.

V.3 - L'organisation séquentielle indexée

Le système de fichiers offre deux types d'organisation séquentielle indexée :

- clé unique
- clés multiples

En plus de l'avantage d'accéder à un article selon sa clé, l'organisation séquentielle indexée de ce système présente l'intérêt de pouvoir rajouter des articles dans le fichier sans aucune contrainte en bénéficiant de l'avantage qu'offre l'allocation dynamique d'espace disque.

### V.3.1 - L'organisation séquentielle indexée à clé unique

Les articles sont rangés successivement dans leur ordre d'arrivée et la table d'indexe est constituée. Pour pouvoir appliquer un algorithme de recherche dichotomique des clés dans la table d'index, celle-ci est triée. Ce qui évite de trier au préalable les articles pour constituer le fichier, comme l'imposent les systèmes tels que le MMT2 (FMS) ou le CII - SIRIS - 7 .

Pour accélérer la recherche de clés, la table d'index est éclatée en trois niveaux, chaque niveau résumant le niveau inférieur [ KEE-74 ].

Mais, vu le caractère évolutif des fichiers, le tri des clés peut engendrer beaucoup de déplacements dans les pages d'index, entraînant des réorganisations de tous les niveaux de la table. Ceci nous a amené à développer une méthode de gestion des clés afin d'éviter ces opérations.

Cette méthode consiste à gérer au début de chaque page de la table d'index un répertoire qui contient

les rangs de toutes les clés de la page. Les clés occupent des places fixes dans la page, leur adresse est déterminée par un simple calcul à partir de leur rang dans la page.

Le rôle du répertoire est de permettre le tri logique des clés et d'indiquer les éléments d'index libres.

#### 4.3.2 - Organisation séquentielle indexée à clés multiples

Le fichier est structuré suivant la clé principale. Les clés secondaires ne modifient pas cette structure et nécessitent des tables d'index de même structure que celles de l'organisation séquentielle à clé unique,

#### V.4 - L'organisation partitionnée

Le fichier partitionné est formé d'un dictionnaire de partitions et d'un ensemble de partitions de longueur variable.

Le dictionnaire est paginé et il est organisé en articles de même longueur.

A une partition, est associé un élément en dictionnaire qui contient son nom et l'adresse physique de la table de pages qui lui est associée.

Les partitions ont la même organisation qui peut être séquentielle simple ou partitionnée. L'accès aux articles de la partition peut se faire en mode séquentiel ou direct.

#### V.5 - L'organisation libre ou directe

Un fichier device est un espace disque non structuré en articles, formé d'une suite de pages élémentaires. Cette espece, alloué par granule, est géré par l'utilisateur qui y accède en mode séquentiel ou direct par adresse de page relative au début des fichiers (adresse virtuelle).

#### VI - Organisation du système de gestion de fichiers

L'organisation du SGF reflète les différentes étapes à franchir dans la réalisation d'une entrée-sortie et ce partant d'un nom symbolique de fichier.

A chaque étape correspond un module du système. Les modules du système sont organisés en une arborescence dont les niveaux représentent les fonctions principales du système.

Chaque module de cette structure ne dépend que des modules de niveaux supérieurs; ce qui facilite notamment la maintenance et la modification de celui-ci **sans** altérer l'ensemble des modules.

Les différents modules sont répartis en 4 groupes :

- les modules d'identification et de contrôle
- les modules de gestion logique et physique du fichier
- le module d'allocation d'espace
- les modules de base qui réalisent l'entrée-sortie physique.

EDITION DE TEXTES

Les systkmes conversationnels disposent d'un ensemble de programmes qui permettent aux utilisateurs de résoudre des problèmes qui leurs sont souvent spécifiques.

Parmi ces programmes, il existe un appelé Editeur de texte que l'on rencontre sur la plupart de ces systkmes et permet de constituer, examiner et modifier un fichier à l'aide de commandes.

Ce programme essentiellement orienté vers le traitement de caractères est souvent adapté à la composition automatique de texte et l'édition de documents et, constitue ainsi un nouveau service appelé Editeur de document. On rencontre des programmes qui sont spécialisés dans l'un des deux services, ou généraux.

Dans ce chapitre, après présentation des caractéristiques des programmes d'édition, nous décrivons les modifications apportées au modèle de base de l'éditeur de documents EDIDOC [ PER-82 ] et les principes de base de l'éditeur de texte du système MISTEP.

I - EDITEUR DE TEXTES

1.1 - Les différents types d'éditeurs

L'éditeur de texte met à la disposition de l'utilisateur un jeu de commandes qui lui permettent de manipuler un fichier.

Un fichier traité par un éditeur de texte est constitué d'éléments rangés dans un certain ordre et caractérisés par leur format fixe ou variable et leur longueur.

Ces éléments sont plus souvent des lignes sur lesquelles où à partir desquelles porte l'action symbolisée par la commande de l'éditeur.

Un éditeur est utilisable soit en mode mise-à-jour automatique (sous batch processing); cas du programme MEDIT [ HUR-76 ], ou en conversationnel; cas de l'éditeur ED du système UNIX [ KER-74 ], [ THO-74 ].

Dans le premier cas, la ligne est désignée par son rang dans le fichier. Dans le second cas, cette méthode de désignation devient peut souple du fait que le numéro de ligne évolue à la suite d'insertions ou de suppressions de lignes; comme elle ne permet pas une véritable interaction.

C'est ainsi qu'un nouveau type d'éditeur est apparu permettant non seulement de désigner la ligne par son rang mais aussi par tout ou une partie de son contenu : ce sont les éditeurs par contexte [ ADI-71 ], tels que les éditeurs QED [ DEU-67 ], TSM [ TSM-76 ], TED12 [ SEM-80c ]. Ainsi, on peut rechercher une ligne contenant une chaîne de caractères donnée .

## 1.2 - Différences entre éditeurs

Les éditeurs de textes se différencient notamment par les moyens de communication utilisés, leur langage de commandes et par le type de fichier traité. Ce qui se

traduit par des différences en leur conception au niveau de l'unité de transfert entre le terminal et l'ordinateur, par la syntaxe des commandes et l'organisation de base du fichier.

Certains éditeurs tel que celui du CMS (Cambridge Monitor System) pour lesquels la ligne est la plus petite unité d'information qu'on peut accéder dans le fichier, sont dits Orientés-ligne.

Ce type d'éditeur est le seul approprié pour les terminaux à papier et pour les systèmes d'exploitation où la ligne est l'unité de transfert de données. Cette caractéristique dépend surtout de la technologie de l'unité de contrôle liée au terminal.

L'autre facteur qui fait qu'un éditeur travaille en mode 'ligne' est l'organisation de base du fichier traité? (voir chapitre II, § 1.2.2.1).

Pour cette catégorie d'éditeurs, l'utilisateur est moins concerné par l'édition des lignes que par leur insertion et suppression. Le plus souvent, le contenu des fichiers est soit un programme soit des données ou des résultats que l'utilisateur désire examiner.

Cependant, dans le cas général de l'édition de texte, il est plus commode de pouvoir accéder directement et individuellement aux caractères surtout si le contenu du fichier est un texte conventionnel dont on désire corriger les fautes d'orthographe et de ponctuation.

Ce genre d'édition est commodément offert par les éditeurs utilisables à travers des terminaux à clavier alphanumérique et écran de visualisation avec contrôle du curseur.

Le curseur peut donc être positionné sur la partie du texte, préalablement visualisée, qu'on veut modifier.

Les éditeurs qui admettent ce mode de traitement sont souvent implantés sur des ordinateurs pour lesquels le transfert se fait par caractère plutôt que par ligne. Ces éditeurs sont dits Orientés-caractères.

A titre d'exemple nous citons :

- L'éditeur GODOT [ MAC-77 ] implanté sur PDP 11/45 sous le système UNIX. A l'origine, il a été conçu pour l'édition de texte en langage naturel. Mais à cause de la souplesse de modification qu'il offre à l'aide du contrôle du curseur , il a été adopté pour la mise au point de programmes-source à la place de l'éditeur standard ED du système UNIX.
- L'éditeur OCCAM [ SNE-78 ] est aussi intéressant et il travaille sur un écran à 4 couleurs. Chacune de ces couleurs correspond à un type de message : texte, commandes, erreurs ou commande et texte.

Le texte manipulé est visualisé par partie. Le caractère à modifier est désigné par le curseur qui est positionné à l'aide de touches de fonction ou d'un 'lightpen'.

## II - EDITEUR DE DOCUMENT

Un éditeur de document assure l'impression de texte selon le format désiré. Le texte est créé et modifié par un éditeur de texte ou par l'éditeur de document lui même.

Ces programmes sont souvent spécialisés dans des applications particulières telle que la composition automatique de revues scientifiques [ CHE-75 ], [ LES-77 ], ou dans des tâches de secrétariat [ PRO-74 ].

D'autres sont moins spécialisés, mais ils sont conçus pour des mini-systèmes, et sont destinés à la confection de rapports, thèses et poycopiés (l'éditeur QUIDS) [ COU-76 ] .

Enfin, les éditeurs de documents tels que AURORE [ MAN-78 ] et TEXT 360 sont généraux.

Les éditeurs de documents se caractérisent par le format du texte en entrée, sa modification et sa mise en page.

Le texte en entrée est le plus souvent présenté sous forme d'enregistrement, incluant des commandes de mise en forme finale du document:

La modification se fait en différé ou en conversationnel. Seul TEXT 360 permet de modifier le texte édité.

### III - L'ÉDITEUR DE DOCUMENT EDIDOC

EDIDOC considère le texte traité comme un seul long enregistrement contenant des commandes et instructions de format.

Il dispose de commandes d'état (saisie, modification, sortie), d'action sur chapitres et paragraphes, et un ensemble de directives globales au document ou locales à un chapitre.

#### Modifications apportées au modèle de base

Le système AURORE impose une structure particulière du texte en entrée; c'est l'utilisateur qui doit se conformer à une

„macro-syntaxe" (saut de lignes, cadrage gauche...). Cette structuration du texte, si elle permet une manipulation aisée et sans restreindre la diversité d'impression, impose des contraintes à la saisie.

Pour bénéficier des avantages de la structuration du texte à traiter et sans imposer de contraintes à l'utilisateur, l'organisation interne du texte a été modifiée de manière qu'il soit structuré en chapitres, paragraphes et dérivées de paragraphe.

Ainsi, le texte est examiné et modifié non pas globalement ou par chapitre, mais par paragraphes et sous-paragraphes ... De plus, cette structuration du texte permet de le manipuler en toute sécurité puisque toute partie sélectionnée est recopiée sur un fichier de travail, comme elle offre l'avantage de pouvoir rééditer que les chapitres modifiés\*.

#### IV - L'EDITEUR DE TEXTE DU SYSTEME MISTEP

La notion d'éditeur de texte est toujours liée à l'existence de deux fichiers : le fichier dit Origine et le fichier dit de Travail [ SNE-78 ], [ TSM-76 ].

Afin d'éviter la perte accidentelle ou par erreur du contenu du fichier qu'on désire manipuler celui-ci est copié sur le fichier de travail qui va contenir toutes les modifications. A la fin du traitement l'utilisateur a le choix de reconstituer ce fichier origine ou non, à partir du fichier de travail.

La constitution du fichier de travail ne se limite pas à une simple copie de fichier, mais elle lui confère une organisation particulière en ajoutant des informations de gestion afin de permettre un accès rapide aux articles [ MAC-77 ] .

---

\* une étude sur la mise au point de texte édité est en cours.

Mais cette opération est d'autant plus coûteuse en nombre d'entrées-sorties, en temps de traitement et en espace disque que la taille du fichier origine est plus grande.

Etant donné que le système MISTEP offre un environnement conversationnel multi-utilisateurs, l'opération de constitution et de reconstitution de fichiers peut entraîner une saturation de l'espace disque et une détérioration du temps de réponse, surtout que tous les fichiers de MISTEP sont sur une même unité disque (problème de partage du processus d'entrée-sortie).

La conception de l'éditeur de texte que nous présentons prend en considération les mesures de sécurité, d'optimalité de gestion de la mémoire secondaire et du temps de réponse.

#### IV.1 - Le langage de commande

Les commandes de l'éditeur suivent la syntaxe imposée par le langage de commande de MISTEP [ BAD-82 ] qui en plus de sa facilité d'assimilation offre trois possibilités de formuler une requête :

- par paramètres optionnels
- par texte de contrôle
- par 'questions-réponses'

L'analyseur de commandes valide syntaxiquement et sémantiquement une commande avant de la transmettre au module chargé de son exécution.

#### IV.2 - Etat d'édition

L'appel du sous-système éditeur de texte se fait par la commande EDIT suivie d'un nom de fichier et d'une option. La fin de l'édition se fait par la commande END suivie d'une option.

On appelle Btat d'édition, l'état du sous-système, qui autorise l'utilisation des commandes de manipulation de texte.

A la commande EDIT, le sous-système prépare un environnement d'édition : il crée le fichier de travail et constitue la table de gestion de ce fichier appelée table des pages Bditables (T.P.E).

#### IV.3 - Organisation des fichiers

Le fichier dont le nom est spécifié en paramètre de la commande EDIT peut exister ou non dans le catalogue du numéro de compte de l'utilisateur. Celui-ci peut rajouter à la suite du nom de fichier l'une des deux options OLD et NEW.

Selon que le fichier est nouveau ou ancien, la constitution de la table de gestion est faite différemment.

Le fichier origine est un fichier séquentiel, format variable et admettant l'accès direct (voir Chap. 111).

Le fichier de travail est de type 'device' dont la taille des pages est égale à 1024 octets. En plus des pages de données du fichier origine qui y sont copiées, il contient les lignes insérées et les pages qui constituent la T.P.E.

#### IV.4 . Particularité par rapport aux autres éditeurs

Un fichier ouvert par un éditeur n'est pas forcément modifié ou il l'est partiellement. Donc sa recopie sur le fichier de travail n'est pas toujours justifiée.

Dans le cas de cet éditeur SEULE UNE PAGE MODIFIÉE EST RECOPIÉE SUR UNE PAGE DU FICHIER DE TRAVAIL.

#### IV.5 - Organisation de la T.P.E.

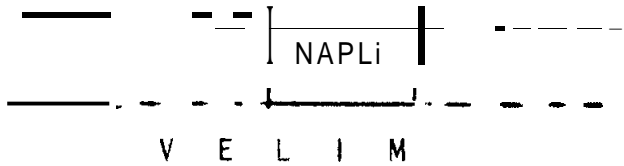
##### IV.5.1 - Description

Elle est formée de pages chaînées entre elles dans les deux sens et permet de gérer l'ensemble des pages de données des deux fichiers origine et de travail.

Chacun de ses éléments est sur 6 octets et décrit une page de données selon la structure suivante :

- numéro de la première ligne dans la page de données,
- numéro logique de la page dans le fichier origine,
- numéro logique de la page dans le fichier de travail.

Les pages de données décrites dans une même page de la T.P.E. constituent un groupe, et les lignes qu'elles contiennent ont des numéros relatifs au groupe (Fig. IV.1).



page i de la T.P.E

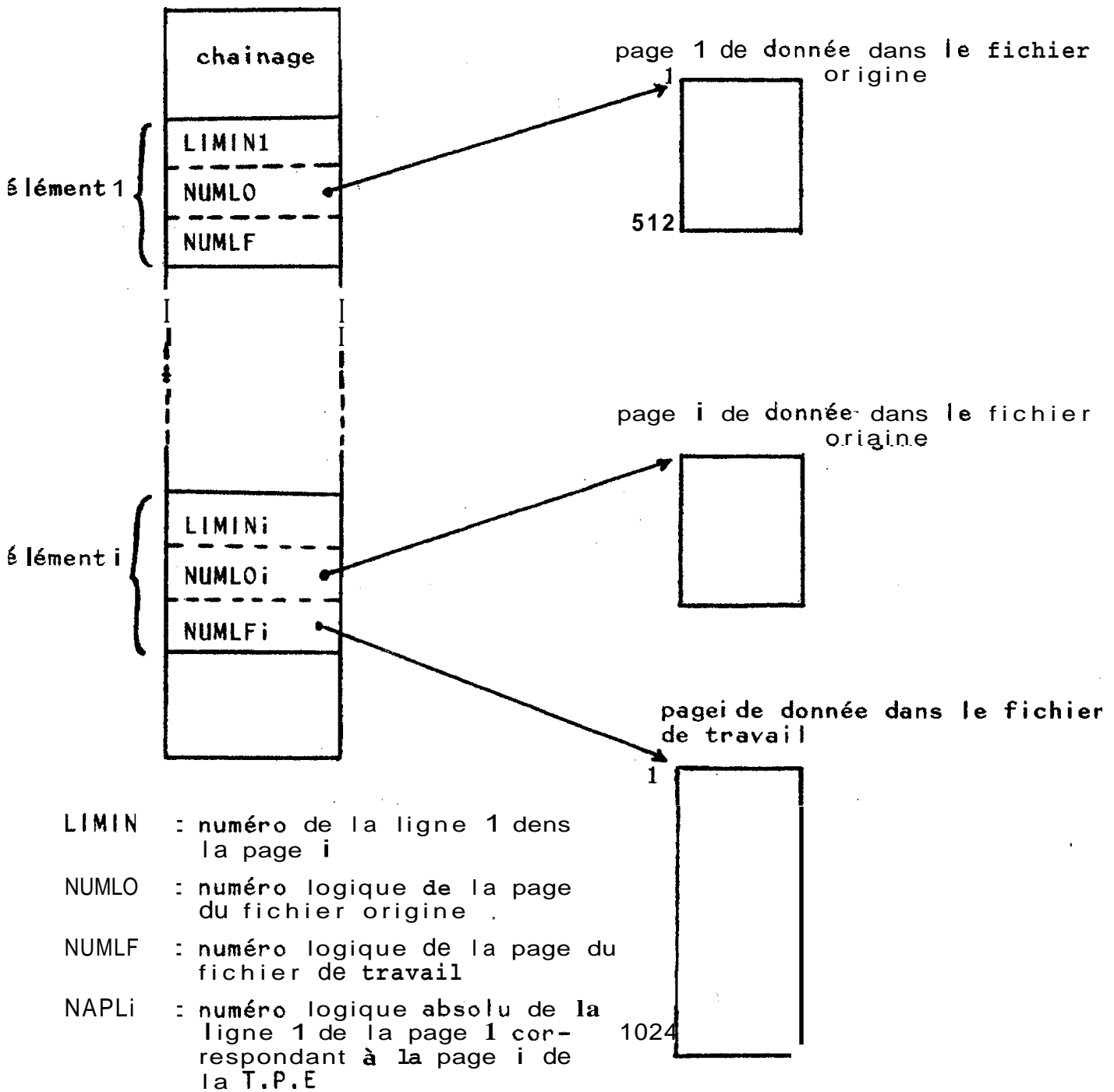


Fig. IV.1

#### IV.5.2 - Initialisation

Lorsque le fichier est 'NEW', l'élément T.P.E. est initialisé par le triplet (0,0,0), et une seule page est allouée à la T.P.E. qui est aussi la première page allouée au fichier de travail.

Par contre lorsque le fichier est 'OLD' un élément de la T.P.E. est initialisé par le triplet:

- numéro calculé de la première ligne dans la page de données décrite,
- numéro logique de la page dans le fichier origine,
- zéro.

Le premier élément de la T.P.E. dont la valeur est (0,0,0) indique la fin de fichier.

En plus de la T.P.E., l'environnement d'édition comporte un vecteur appelé 'vecteur des numéros (logiques) absolus minimaux' (VELIM) rangé en mémoire centrale et dont chaque élément contient à tout instant le numéro logique de la première ligne de la première page d'un groupe de pages.

#### IV.6 - Organisation logique des articles

##### IV.6.1 - Numérotation

Les lignes ont des numéros logiques attribués selon une numérotation à deux niveaux :

N 1 - numéro logique par rapport au groupe de pages.

N 2 - numéro logique par rapport au fichier

Il s'agit de la page contenant la ligne appartient au fichier de travail, sinon il est fictif.

N 2 est un numéro fictif selon lequel la ligne est symboliquement désignée.

#### IV.6. - Principe de numérotation des lignes

on distingue deux sortes de lignes :

- la ligne dite origine qui est issue du fichier origine ou d'une saisie dans le fichier de travail; elle a un numéro multiple de 10.
- la ligne dite insérée qui n'appartient pas au fichier origine; elle a un numéro compris strictement entre  $n$  (modulo 10) et  $n + 10$ .

L'utilisateur désigne une ligne origine par son numéro réel multiplié par 10 et une ligne insérée, par le numéro qui lui a été attribué lors de son insertion.

#### IV.7 - Organisation d'une page de données éditables

Une page du fichier origine devient éditables si l'un de ses articles est modifié ou créé durant la session d'édition en cours; elle appartient au fichier de travail et contient en plus des lignes copiées ou créées

durant **la** session d'édition en cours; elle appartient au fichier de travail et contient en plus des lignes copiées ou **créées**, un vecteur dont chaque élément de **4** octets est formé d'un numéro de ligne, de son adresse dans la page et de son btat. Ce vecteur est construit au moment de la recopie d'une page du fichier origine sur le fichier de travail, ou dynamiquement à chaque création ou modification de ligne (Fig. IV.2), et permet d'accéder directement aux lignes de **la** page.

L'espace libre dans la page du fichier de travail sert à contenir les lignes insérées et permet l'extension du vecteur-index. C'est ainsi que lors de **la** première constitution d'une page de données, les lignes **créées** n'occupent pas plus de **512** octets afin de permettre d'éventuelles insertions entre ces lignes.

#### IV.8 - Exemple de recherche d'une ligne

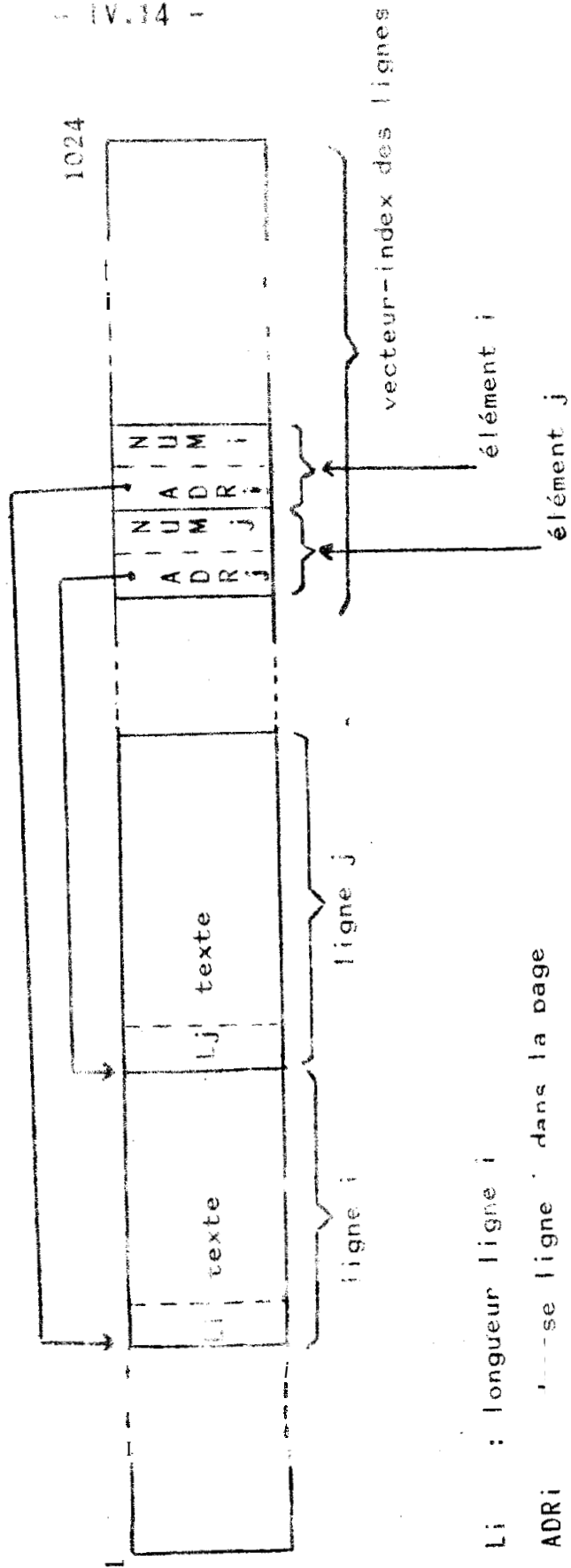
La **T.P.E.** permet d'avoir une vue d'ensemble sur **les** deux fichiers, et donc de savoir à quel fichier appartient **la** page contenant **la** ligne recherchée (Fig. IV.3).

La ligne dont le numéro est spécifié par l'utilisateur, est recherchée de **la** manière suivante :

- recherche du groupe **de** pages auquel appartient **la** page **la** contenant. Le rang  $i$  du groupe recherché est donné par l'inégalité  $VELIM(i) \leqslant NUMERO < VELIM(i+1)$
- calcul du numéro de la ligne par rapport au groupe,
- recherche de l'élément de **la** **T.P.E.** décrivant **la** page contenant la ligne.

Fig. IV.2

ORGANISATION D'UNE PAGE DE DONNÉES ED TABEL



$L_i$  : longueur ligne  $i$

ADRI : adresse ligne dans la page

NUM  $i$  : n° de la ligne  $i$  rapport au groupe de pages

$i = 1, 2, \dots, p$

Il y a deux possibilités

- la page n'est pas citée : la ligne est recherchée dans une page originale.
- la page est citée : le numéro calculé est recherché dans le vecteur-index qui donne la position de la ligne dans la page.

#### IV.9 - Reconstitution du fichier origine

A la commande de fin de session 'END', l'utilisateur peut spécifier s'il faut ou non reconstituer le fichier origine.

Si la fin de session est commandée avec l'option de sauvegarde, le système procède à la reconstitution du fichier origine, par groupe de pages, en utilisant la T.P.E. Et, seules les pages modifiées sont recopiées dans le fichier origine.

#### IV.10 - Les différents états du système

A l'appel de l'éditeur, il peut être dans l'un des deux états, selon que le fichier édité existe ou non :

- état mise-à-jour
- état saisie

Dans l'état mise-à-jour, l'éditeur accepte toutes les commandes dont l'action porte sur les lignes : listage, recherche, insertion, remplacement de lignes, ou modification, remplacement d'une chaîne de caractères dans la ligne en cours.

fichier origine

DESCRIPTION LOGIQUE DE L'ENSEMBLE

FICHER ORIGINE - FICHER DE TRAVAIL

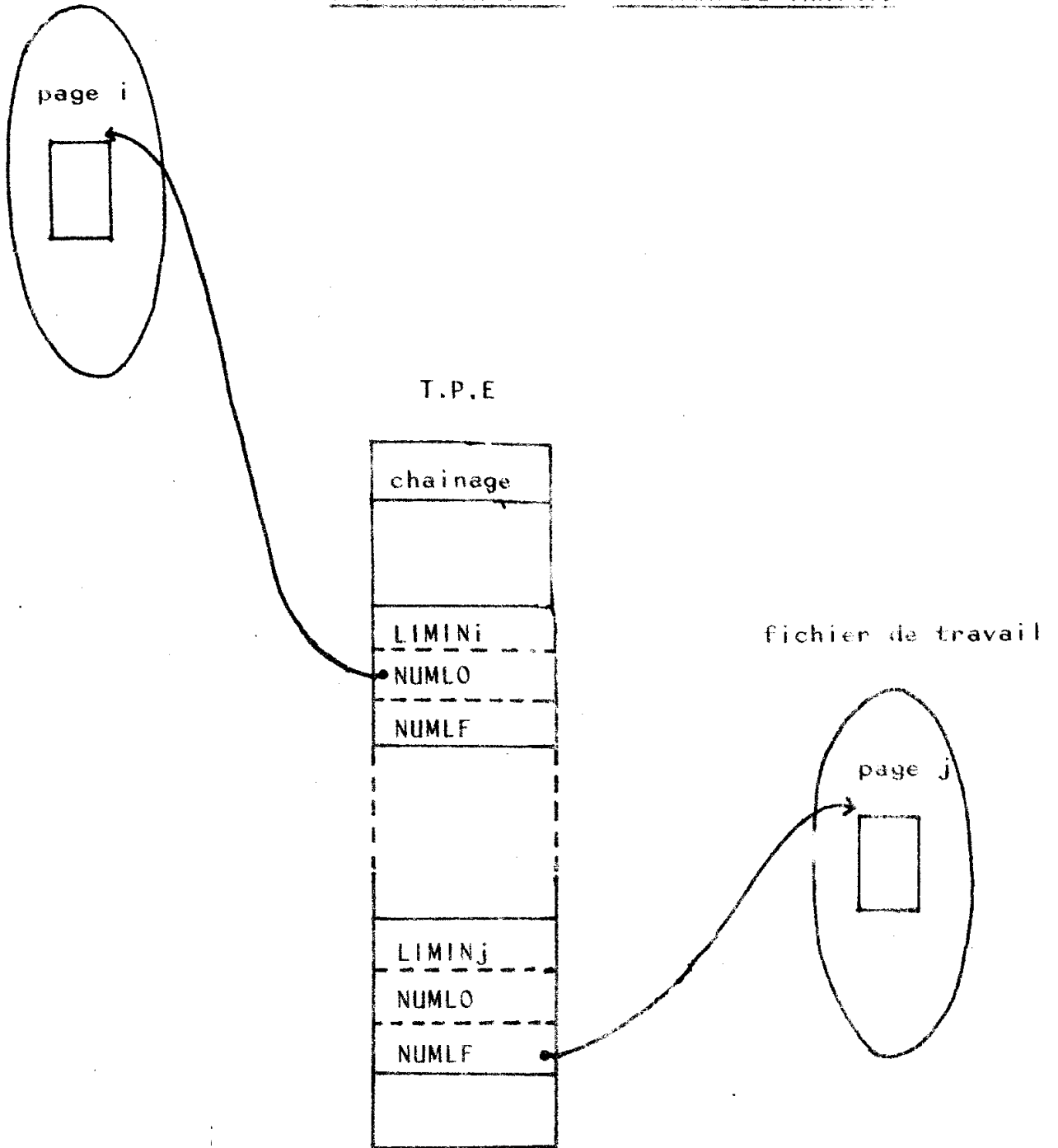


Fig. IV.3

Lorsque le systbme est dans l'état saisie, toute donnée rentrée est une ligne du texte.

Le passage de l'état saisie à l'Etat mise-à-jour se fait par l'émission d'une ligne vide.

#### IV.11 - Environnement associé à un utilisateur

En plus du descripteur d'un poste de travail (voir implé-  
mentation de MISTEP), l'éditeur associe à chaque utili-  
sateur un espace mémoire dans lequel sont gérées la pa-  
ge courante de la T.P.E. et deux pages de données Bdita-  
bles.

## (C) O N C L U S I O N

---=o0\$Co=---

Le système de gestion de fichiers du système MISTEP, bien que conçu pour répondre à divers **types** d'application en offrant différents modes d'organisation des données (organisation séquentielle format fixe, organisation sdquentielle format variable, organisation sdquentielle indexée à clé simple et multiple etc...) est actuellement utilisé uniquement en organisation sdquentielle à accès direct & des articles de format variable et en organisation "device" (ou libre) pour **la** création et **la** gestion des fichiers utilisateurs et des fichiers catalogues du SGF.

Notons par ailleurs que le SGF dans sa version actuelle est monovolume (une unite disque de 50 Mo) et monogroupe. Comme nous l'avons signalé dans la première partie ddcrivant l'implémentation de MISTEP, une version multigroupe du système (création de plusieurs copies du groupe avec partage du volume entre les différents groupes permettrait un accroissement du nombre d'utilisateurs (chaque copie supportant jusqu'à sept postes de travail) et par consdquent entraînerait forcèment un plus grand besoin en espace mémoire secondaire pour **le** stocksye des informations des utilisateurs et ndcessiterait alors une extension du SGF à une gestion multivolume.

Quant à l'éditeur, il se compose d'un ensemble de commandes simples fournissant **des** réponses rapides mais ne permet pas de regrouper **des** requêtes.

Pour effectuer un travail **spécialisé**, l'utilisateur est souvent **amené** à effectuer plusieurs fois les mêmes **requêtes**. Pour rendre ces opérations plus **agréables** et éviter une perte de temps au programmeur expérimenté, il serait avantageux d'ajouter la possibilité de créer, d'archiver et d'appeler des **macro-requêtes** regroupant plusieurs commandes **élémentaires**  
I ADI-71 I . I MAC-77 I.

D'autre part, une projection de la structure interne des **fichiers** (paginée) sur le mode de **visualisation** en permettant une édition par page et l'utilisation de consoles **graphiques** conférerait au **système** une plus grande souplesse **d'utilisation**, particulièrement pour **l'édition** de documents  
I ADI-71 I.

L'utilisation de **MISTEP** permet à un ensemble d'utilisateurs qui, habituellement saisissent leurs programmes sur cartes **perforées**, de réduire le temps consacré à **l'écriture** et la correction des erreurs de programmation d'une part et de soumission au batch d'autre part.

Cette phase de correction d'un programme ne constitue **qu'une** première étape dans la mise au point définitive permettant d'obtenir les **résultats escomptés**. La seconde **étape** consiste à trouver la **marche** à suivre pour mettre en évidence une anomalie dans l'exécution d'un programme, cause d'une fin **anormale** ou de l'obtention de résultats **erronés**. Cette **dernière** étape est certainement plus difficile à franchir et nettement plus **coûteuse** que la **première**.

Des outils de **debugging** sous batch sont offerts par le **système MITRA/125**, mais leur utilisation manque de souplesse. **L'écriture** d'un sous-système de **debugging** utilisant l'aspect conversationnel de **MISTEP**, permettrait aux utilisateurs de mettre au point plus rapidement leurs programmes et **augmenterait** leur motivation,

Dans sa version actuelle, **MISTEP** constitue une plateforme de départ pour la conception et la réalisation d'un système d'aide à l'enseignement et contribuera certainement à améliorer l'environnement de travail du programmeur d'une manière générale et de l'étudiant en particulier; surtout que la plupart des universités algériennes sont équipées de mini-ordinateurs de la gamme MITRA.



# BIBLIOGRAPHIE

---o0\$0o---

- I ADI-71 I ADIBA M.  
Editeur par contexte sous systemes conversationnels partage de temps  
Thèse 3<sup>e</sup> Cycle, Grenoble, (1971)
- I BAC-69a I BACHMAN C.W.  
Data structure diagrams  
Data base 1,2. Quarterly news letter of ACM-SIGBDP, (Summer 1969)
- I BAC-72b I ECHWAN C.W.  
the evolution of storage structures  
CACM Vol. 15, n°7, p: 628-634, (July 1972)
- I BAD--82 I BADACHE N.  
MISTEP : implémentation et langage de commande  
Thèse de Magister, U.S.T.H.B., (1982)
- I BAR-75 I BARBOZA R.  
Contribution à l'étude sémantique des bases de données : application au système SOCRATE  
Thèse Docteur Ingénieur, Grenoble (1975)
- I BAT-78 I RARNETT J.K.R.,  
A highly reliable file system which supports multiprocessing  
Software practice and expérience,  
Vol. 8, P: 645-672 (1978)

- I BEL-71 I BELL C.G., NEWELL A.  
Computers structures : readings and examples  
MC GRAW-HILL, BOOK CO. New-York (1971)
- I CAR-73 I CARDENAS A.F.  
Evaluation and selection of file organization :  
A model and a system  
CACY Vol. 16,N°9, P:243-25 , (1973)
- I CHA-69 I CHAPIN N.  
A comparison of the organization technics  
Proceedings of the ACM 24 th  
National conference (1969)
- I CHE-75 I CHERRY L.L., KERNIGHAN B.S .  
A system for typesetting mathematics  
CACM, Vol. 18, P: 151-157 (1975)
- I CLA-66 I CLARK W.A .  
The functional structure of OS/360  
part III : data management  
IBM systems Journal Vol. 5, N°1, (1966)
- I COU-76 I COOPER G.F., DREHMAN J., HUTCHINSON J.R.,  
PATEL M.H., REEVES T., WUNDERBANK D.G.  
The design and implementation of an interactive  
document editor  
Software Practice and Experience,  
Vol. 6, P: 271-279, (1976)
- I CRI-65 I CRISMAN P.A  
The compatible Time Sharing System. A programmer's  
guide.  
The MIT Computation center (1965)

- I CRO-75 I CROCUS  
Systemes d'exploitation des ordinateurs.  
DUNOD, (1975)
- I DAL-65 DALEY R.C., NEWMANN P.G.  
A general purpose of file system for secondary  
storage.  
Proceedings AFIPS SJCC, part I, P:213-229, (1965)
- I ELI-79 I EL-IRAKI A., BENKIRANE M.A.  
Système interactif d'édition de textes : concep-  
tion-réalisation-implantation  
Thèse 3" Cycle, Toulouse (1979)
- I FAL-64 I FALKOFF A.O., IVERON K.E., SUSSENGUTH E.H.  
A formal description of system 360  
IBM Systems Journal, Vol. 3, N° 3, (1964)
- I FAR-71 I FARR W.W., PEISEL W.E.  
An optimized disc organization for a virtual  
memory system  
computer design, Vol. 10, Part. 6, P:49-54, (1971)
- I FRA-79 I FRASER C.W.  
A compact, portable CRT-based text editor  
Software Practice and Experience,  
Vol. 9, P:121-125, (1979)
- I FRI-70 I FRIEDMAN T.D.  
The authorization problem in shared files  
IBM Systems Journal, Vol. 9, N° 4, P:258-280  
(1970)

- I HIL-77 I HILLMAN M.L., SCHOFIELD D.  
EDIT - an interactive network service : design  
and implementation  
Software Practice and Experience, Vol. 7,  
P: 595-611, (1977)
- I HUZ-79 I HUZAN E.  
Modelling of indexed sequential files monoto-  
ring disc transfert  
The computer Journal, Vol. 22, N° 1, P:22-27,  
(February, 1979)
- I KAT-71 I KATZAN H. Jh.  
Storage hierarchy systems  
Proceedings, AFIPS, SJCC, Vol. 38, P:325-336,  
(1971)
- I KEA-76 I KEALING MC., WILSON  
Studies in operating system  
MC GRAW-HILL, (1976)
- I KEE-74 I KEEHN D.T., LACY J.O.  
VSAM data set parameters  
IBM Systems Journal, Vol. 13, N° 3, P:24-29,  
(1974)
- I KER-74 I KERNIGHAN B.W.  
A Tutorial introduction to the UNIX text editor  
Bell Laboratories, Murray HILL, New Jersey, (1974)
- I LES-77 I LESK M.E., KERNIGHAN, B.W.  
Computer Typesetting of technical Journals on  
UNIX  
National computer conference, P:879-888, (1977)



- I PIR-75 I      PIRKOLA GARY C.  
A file organization for a general purpose  
time-sharing environment  
Proceedings of IEEE, Vol. 63, N° 6,  
P:918-924, (1975)
- I PRA-70 I      PRADEL P.M., ROCHER M.  
Un système hiérarchisé de fichiers simulta-  
nément partageables  
AFCET, Congrès Informatique, Paris (1970)
- I PRO-74 I      PROFIZI J.C.  
Contribution aux techniques de mise en page  
de l'édition par ordinateur. Conception et  
implantation de LAMPE : un langage spécialisé  
Thèse 3<sup>e</sup> Cycle, Grenoble (1974)
- I RIT-78 I      RITCHIE D. A.  
UNIX : a retrospective  
The Bell Journal System, Vol. 57, N° 6,  
Part. 2, P:1947-'969, (July-August 1978)
- I ROT-74 I      ROTHNIE J.J., LOZANO T.  
Attribute based file organization in a paged  
memory environment  
CACM Vol. 17, N° 2, P: 63-69 , (1974)
- I SCH-77 I      SCHNEIDER B.R.J., REID M.W.  
SITAR : An interactive text processing system  
for small computers  
CACM Vol. 20, N° 6, P:495-595, (1977)

- I SEM-80a I SEMS/MITRA 125  
Moniteur MMT2, manuel de référence  
Louveciennes Paris, (1980)
- I SEM-80b I SEMS/MITRA 125  
Système de gestion de fichiers, manuel de  
référence  
Louveciennes Paris, (1980)
- I SEM-80c I SEMS/MITRA 125  
Editeur de texte TEDI2, manuel d'utilisation  
Louveciennes Paris, (1980)
- I SNE-78 I SNEERINGER J.  
User-interface design for text editing : a case  
study  
Software Practice and Experience,  
Vol. 8, P:543-557, (1978)
- I STE-67 I STEIG O.J.-  
File management on a small computer. The C10  
system  
Proceedings AFIPS SJCC, Vol. 30, P:199-212, (1967)
- I TSM-76 I I.R.E.A.C.S - CNRS  
Système T.S.M :  
→ manuel de référence  
→ système de gestion de fichiers  
→ éditeur de texte  
→ langage de commande  
Lyon , (1976)

I WIL-76 I

WILSON R., Mc KEAG R.M.

Studies in operating systems

ACADEMIC PRESS, LONDON, (1976)

E R R A T A

---oO&Oo---

Page 1.8

Ligne 14 lire... ne doit pas dépasser... au lieu de... ne doit dépasser

Page 11.1 6

Ligne 8 lire... en ligne (ou plat)... au lieu de... en ligne ou plat

Page III.14

Ligne 24 lire... qu'il lui est... au lieu de... qu'il est...

Page III.17

Ligne 4 lire... qui procède... au lieu de... qui pocède

Page III.26

Ligne 2 lire... en leur... au lieu de... en son

Page 111.28

Ligne 16 lire... mais elle est... au lieu de... mais est

Page IV.1

Ligne 11 lire... édition de document... au lieu de... éditeur de document

Page IV.4

Ligne 15 lire... interactif... au lieu de... intératif

Page IV.7

Ligne 9 lire... processeur... au lieu de... processuer

Page IV.13

Ligne 1 supprimer de... elle appartient... à ... créés.