Chapitre Troisième : Les Fichiers et Enregistrement

Chapitre Quatrième

Les fichiers et Enregistrements

I. Introduction

Un tableau est une variable composée concernant un ensemble de données ayant **les mêmes caractéristiques (type)** et désigné par le même **nom_identificateur** et dont la taille est **fixée et non évolutive** (Statique) **,** et est **présente en mémoire** durant l'exécution du programme.

Exemple :

MOIS

JANVIER	FEVRIER	.....	NOVEMBRE	DECEMBRE

Var MOIS(12) = Caractère ;

- Tous les éléments du tableau sont de **type caractère,**
- Le tableau doit être **rempli au début de l'exécution du programme,**
- et **après l'exécution,** toutes les données **sont effacées et perdues**
- A chaque **exécution du programme**, il faut remplir le tableau

Les fichiers servent **à stocker** des informations **de manière permanente, entre deux exécutions** d'un programme. Car si les variables, qui sont des adresses de mémoire vive, disparaissent à chaque fin d'exécution, les fichiers, eux sont stockés sur des périphériques à mémoire de masse (disquette, disque dur, CD Rom, Carte mémoire, Clef USB,...).

NUMINSC	**NOM**	**PRENOM**	**ADRESSE**	**MOYENNE**
N(6)	A(20)	A(25)	AN(45)	N(5.2)

(**N** : Numérique ; **A** : Alphabétique ; **AN** : Alphanumérique)

Figure IV.1 : Dessin d'un enregistrement

II. Définitions

II.1. Champs

C'est une zone mémoire qui reçoit des données.

Exemple :

- **NUMINSC** pour le champ qui reçoit les numéros d'inscription
- **NOM** pour le champ qui reçoit les noms

II.2. Enregistrement

Un enregistrement est un groupe de champs **homogènes** (sur un même objet) et qui peuvent être de **type différent.** Un enregistrement doit être désigné par un nom.

Exemple :

On peut avoir un enregistrement désigné **ETUDIANT** qui aura pour champs :

- **NUMINSC** pour le numéro d'inscription
- **NOM** pour le nom
- **PREN** pour le prénom
- **DAT-NAIS** pour la date de naissance
- **ADRES** pour l'adresse
- …
-

Dans un algorithme, l'enregistrement doit être déclaré sous forme de **type**.

II.3. Fichier

Un Fichier est un ensemble d'enregistrements. Un fichier doit être désigné par un nom et est relatif à un type d'enregistrement.

Exemple :

On peut avoir le fichier **FETUDIANT** composé des enregistrements des étudiants de type **ETUDIANT**

III. Caractéristiques des fichiers

- Un fichier est stocké sur un support
- Les enregistrements d'un fichier sont rangés selon une organisation (Séquentielle, Directe) ;
- Pour travailler avec un fichier, il faut l'ouvrir
- Quand on ne travaille plus avec un fichier il faut le fermer.
- Un fichier est ouvert en Lecture ou en Ecriture (Jamais les 2 à la fois) ;
- La lecture d'un fichier consiste en une lecture d'un enregistrement **en entier** (pas d'un champ) ;
- La lecture d'un enregistrement consiste à placer un enregistrement dans la zone tampon.

IV. Déclaration d'un fichier

Un fichier est une structure de **donnée composée** et doit être déclarée dans la partie déclaration. La déclaration consiste à donner **la structure de son enregistrement** sous forme d'un **Type.**

```
Type ENRG = enregistrement
              Var1 : type1 ;
              Var2 :Type2 ;
              Var3 :Type3 ;
              ……..
              Varn :Typen ;
              Fin;
Var FichierF: fichier de ENRG;
    ….
    ….
```

Ainsi, quand on utilise un fichier, il faut d'abord déclarer :

- Le **type** de son **enregistrement** (sa structure)
- La **variable** fichier

Exemple en Pascal:

```
PROGRAM GESTION-PATIENT;
Type Patient Record of
            NUMDOS: char
            NOMP: String[15] ;
            PRENP: String[20];
            DAT-NAIS:Date;
            ADRES: String[45];
            ACT: Char;
            SEX: Char;
            ……..
            End;
Var FP-PATIENT: File of Patient;
…..
Begin
……..
 ……..
End.
```

Description de la structure de l'enregistrement

Déclaration du fichier

V. Organisation d'un fichier

V.1. Organisation séquentielle

Dans cette organisation, les enregistrements du fichier sont placés séquentiellement (les uns derrière les autres) sur le support.

V.2. Organisation Directe

Dans cette organisation, les enregistrements possède **une clé (identifiant)** et sont placés aléatoirement sur le support, cependant, leur adresse (l'emplacement où ils ont été stockés) est gardé dans une table d'index (gérée par le système) (qui sera utilisé pour **accéder directement** aux enregistrements).

VI. Création d'un fichier

Tout fichier doit avoir été créé. La création consiste à nommer le fichier, spécifier le support et son organisation. C'est la première opération qui doit être effectué avant la manipulation des fichiers.

VII. Accès à un enregistrement d'un fichier

Le type d'accès, indique la manière dont la machine va pouvoir aller rechercher les informations contenues dans le fichier.

On distingue :

- ***L'accès séquentiel*** : on ne peut accéder qu'à la donnée suivant celle qu'on vient de lire. On ne peut donc accéder à une information qu'en ayant au préalable examiné celle qui la précède. cela signifie qu'on lit le fichier enregistrement par enregistrement.
- ***L'accès direct*** *(ou* ***aléatoire****)* : on peut accéder directement à l'enregistrement de son choix, en précisant **le numéro** de cet enregistrement

VIII. Opérations sur les fichiers

VIII.1. Ouverture d'un fichier

Un fichier doit d'abord être ouvert avant toute opération.

Si l'on veut travailler sur un fichier, la première chose à faire est de l'**ouvrir**. Cela se fait en attribuant au fichier un **numéro de canal**. On ne peut ouvrir qu'un seul fichier par canal, mais quel que soit le langage, on dispose toujours de plusieurs canaux.

Un fichier est ouvert en Lecture ou en Ecriture.
L'important est que lorsque l'on ouvre un fichier, on stipule ce qu'on va en faire :
Lire, Ecrire ou Ajouter.

- Si on ouvre un fichier **en Lecture,** on pourra uniquement récupérer les informations qu'il contient, sans les modifier.
- Si on ouvre un fichier **en Ecriture,** on pourra mettre dedans toutes les informations que l'on veut. Mais les informations précédentes, si elles existent, seront **intégralement écrasées** et on ne pourra pas accéder aux informations qui existaient précédemment. (fichier séquentiels)
- Si on ouvre un fichier **en Ajout,** on ne peut ni lire, ni modifier les informations existantes. Mais on pourra **ajouter** de **nouveaux enregistrements**.

OuvrirFichier(F) en Ecriture en Séquentielle(Directe)

OuvrirFichier(F) en Lecture Séquentielle(Directe)

VIII.2. Lecture d'un enregistrement d'un fichier

La lecture d'un enregistrement consiste à placer l'enregistrement dans une zone mémoire tampon qui permet de manipuler les données (les champs) de l'enregistrement.

LireFichier(F, X)

VIII.3. La fin d'un fichier

La fin d'un fichier est indiquée par FinDeFichier (EOF)(End Of File) .
Pour tester la fin d'un fichier on utilise la structure :

```
TantQue NOT (EOF)  Faire
        Début
        ....
        ....
        Fin ;
```

VIII.4. Ecriture dans un fichier

L'écriture dans un fichier consiste à ajouter un enregistrement dans le fichier. Cette écriture dépend de l'organisation du fichier (Séquentielle ou Directe).

Ecrire(F, X)

VIII.5. Fermeture d'un fichier

Quand on n'a plus besoin du fichier, on doit le **fermer**, ainsi, tout accès au fichier (aux enregistrements) est impossible. Tout nouvel accès au fichier nécessite sa **réouverture.**

FermerFichier(F)

VIII.6. Principales opérations sur les fichiers

3 principales opérations sont définies sur les fichiers :
- L'Insertion d'un enregistrement (nouveau) dans le fichier
- La modification d'un champ (de l'enregistrement) dans un fichier

- La surpression d'un enregistrement d'un fichier.

Ces opérations sont désignées par **Opérations de Mise-à-Jour**. Il existe une autre opération qui est très fréquemment utilisée sur les fichiers : **La consultation** d'un enregistrement qui consiste juste à visualiser et afficher certains (ou tous) champs d'un enregistrement particulier.

Toute ces opération dépendent de l'organisation du fichier et le fichier doit avoir été ouvert et doit être fermé après avoir réalisé l'opération.

VIII.6.1. L'insertion d'un enregistrement

L'insertion d'un enregistrement consiste à ajouter un enregistrement au fichier.

VIII.6.2. La modification d'un enregistrement

La modification consiste à modifier une information contenue dans un champ parmi les champs de l'enregistrement

VIII.6.3. La surpression d'un enregistrement

La suppression d'un enregistrement consiste à supprimer un enregistrement du fichier.

IX. Application en Pascal

IX.1. Déroulement des opérations

1- Déclarer le fichier (la variable f)

```
f : file of qqchose ;
```

2- Assigner la variable f au fichier de nom nomf

```
Assign (f, nomf) ;
```

3- Ouvrir le fichier f pour pouvoir y lire ou y écrire les données

```
Reset (f) ; ou rewrite (f) ;
```

4- Lire ou écrire des données

```
Read (f, donnée) ; ou Write (f, donnée) ;
```

5- Quand on a fini, on ferme le fichier

```
Close (f) ;
```

IX.2. Lecture ou écriture

On ouvre un fichier soit en **Lecture**, soit en **Ecriture**. On ne peut pas faire les deux en même temps.

- ***En Lecture :*** on fait Reset(f); puis des Read(f, ...);

Si le fichier n'existe pas, il y a une erreur.

- ***En Ecriture :*** on fait Rewrite(f); puis des Write(f, ...);

Si le fichier n'existe pas, un rewrite le crée. Si il existe déjà, le rewrite l'écrase, c'est-à-dire que l'ancien contenu est définitivement perdu.

IX.3. La Fin du fichier

En lecture, avant de faire un **Read**, il faut tester s'il y a encore quelque chose à lire ;
on n'a pas le droit de faire un **Read** si la fin du fichier est atteinte.
La fonction **eof(f)** retourne **True** si la fin du fichier est atteinte.

IX.4. Exemple d'utilisation des fichiers dans un programme en Pascal

```
PROGRAM Exemple ;
.....
BEGIN
Assign (f, nomf);
Reset (f);
nb := 0;
While not eof(f) and (nb < vmax) do
Begin
nb := nb+1;
Read(f, vec[nb]);
End;
Close (f);
END;
```