

CHAPITRE9

LES FICHIERS

1.Introduction

Jusqu'ici les données utilisées dans nos programmes étaient stockées de manière temporaire dans la mémoire vive de l'ordinateur ; mais souvent on a besoin de stocker les données et les résultats des programmes de façon permanente ; parmi les moyens utilisés qui permettent de stocker les données de façon permanente sont les bases de données et les fichiers.

Dans ce chapitre on va s'intéresser aux fichiers.

2. Définition :

Un fichier structuré est une collection d'informations toutes de même type stockées les unes à la suite des autres sur un support physique (disque dur, cd, dvd, etc....) et terminé par une marque de fin de fichier.

On peut faire l'analogie avec un fichier manuel comportant des fiches chaque fiche correspond à une information et le fichier correspond à la boîte d'archive.

D'un point de vue informatique un fichier informatisé peut être vu comme un ruban sur lequel sont écrits des enregistrements et défilant devant une tête de lecture/écriture l'enregistrement placé devant cette tête est appelé enregistrement courant.

Exemples :

-  Un fichier d'étudiants
-  Un fichier d'articles
-  Un fichier d'employé

Caractéristiques d'un fichier :

- Un fichier un nom, un mode d'organisation et un mode d'accès.
- On ce qui concerne le nom d'un fichier on distingue le nom interne qui est le nom avec lequel un fichier est identifié dans un programme et un nom externe qui comporte le

chemin du fichier (nom du support physique), le nom du fichier ainsi que son extension.

- Pour le mode d'organisation d'un fichier on trouve le mode séquentiel, le mode séquentiel indexé, le mode relatif ou sélective.
- Pour le mode d'accès d'un fichier on trouve l'accès séquentiel et l'accès direct.
- Un fichier séquentiel permet uniquement d'accéder aux données dans l'ordre de leur écriture (accès séquentiel).
- Un fichier à accès direct (fichier à organisation relative) correspond à un tableau en mémoire : toutes ses composantes ont la même taille, on peut donc accéder directement au N^{ème} enregistrement. Comme pour un tableau, les insertions et les suppressions nécessitent des décalages des composantes du fichier.
- Pour les fichiers séquentiels indexés ou sélectifs Les composantes sont stockées dans le fichier dans l'ordre de leur création. On utilise alors un tableau d'index, donnant en première position le numéro de la première composante, puis de la seconde,....

3. Les fichiers séquentiels :

3.1 Déclaration d'un fichier séquentiel :

Syntaxe :

```
Type  
Nom_fichier= Fichier de type  
Variables  
F : Nom_fichier
```

Exemple :

```
Type  
Voiture=Struct  
    Marque : chaine  
    Prix: réel  
    Nserie : entier  
Fin Struct  
FiVoi=Fichier de voiture  
Variables  
V : Voiture  
F : FiVoi
```

3.2 Manipulation des fichiers séquentiels :

Ouverture d'un fichier :

Syntaxe :

Ouvrir (Nom_fichier, mode)

On peut ouvrir un fichier pour consultation on appelle ça mode lecture(L) comme on peut l'ouvrir pour modification on appelle ça mode écriture(E) après, ouverture du fichier on aura un pointeur qui pointe sur la 1^{ère} donnée du fichier.

Exemple :

Ouvrir (F, E)

On va ouvrir le fichier F qui est un fichier de voiture en mode ecriture afin d'ajouter, de modifier ou de supprimer des voitures.

Lecture d'un Fichier :

Syntaxe :

Lire(Nom_fichier, x)

Lors la lecture d'un fichier on commence toujours par la 1^{ère} donnée du fichier chaque donnée lue est récupérée dans une variable (tampon) qui a le même type que les éléments du fichier.

Si on veut lire une donnée particulière il faut commencer depuis le début jusqu'à atteindre l'information voulue.

Ecriture dans d'un Fichier :

Syntaxe :

```
Ecrire(Nom_fichier, x)
```

Exemple :

```
V.Marque ← ford  
V.Prix ← 20000  
V.Nserie ← 136  
Ecrire(F, V)
```

On a copié le contenu de l'enregistrement V dans le fichier F.

Remarque :

Dans un fichier séquentiel l'insertion se fait toujours à la fin du fichier.

Fermeture d'un fichier :

Syntaxe :

```
Fermer(Nom_fichier)
```

Après toute utilisation du fichier on doit obligatoirement le fermer.

Exemple :

```
Fermer(F)
```

Test de la fin d'un fichier :

Syntaxe :

```
X ← FDF(Nom_fichier)
```

Pour tester si on a atteint la fin du fichier ou non utilise cette fonction qui retourne vrai si c'est le cas sinon elle retourne faux.

Exercice d'application 1 :

Écrire une procédure qui permet de remplir un fichier de 10 article un article et caractérisé par son libellé, son prix et sa quantité.

Correction :

```
Type
Article= Struct
    Libelle : chaine
    Prix : réel
    Quantite : réel
Fin Struct
Fich= Fichier d'article
Procédure Remplissage(var F :Fich)
Variables
x : Article
i : entier
Début
    Ouvrir(F, E)
    Pour i de 1 à 10 Faire
        Ecrire("donner le nom du produit")
        Lire(x.Libelle)
        Ecrire("donner le prix du produit")
        Lire(x.Prix)
        Ecrire("donner la quantité du produit")
        Lire(x.Quantite)
        Ecrire(F, x)
    Fin Pour
    Fermer(F)
Fin
```

Exercice d'application 2

Écrire une procédure qui permet d'afficher le contenu d'un fichier d'un article.

Correction

```
ProcédureAffichage(F :Fich)
Variables
x : Article
Début
    Ouvrir(F, L)
    Lire(F, x)
    Tant que (FDF(F)=faux) Faire
        Ecrire("le nom du produit est", x.Libelle)
        Ecrire("le prix du produit est", x.Prix)
        Ecrire("la quantité du produit est", x.Quantite)
        Lire(F, x)
    Fin Tant que
    Fermer(F)
Fin
```

4. Les fichiers Texte :

4.1 Définition :

Les fichiers de texte se sont des fichiers séquentiels: ils en fait souvent des fichiers à accès direct aux caractères (on peut accéder directement au N^{ème} caractère d'un ligne), mais en fait l'accès est séquentiel au niveau des lignes : pour accéder à la N^{ème} ligne, il faut lire le fichier depuis le début, jusqu'à compter N-1 signes de fin de ligne.

Les fichiers de texte peuvent également être utilisés pour stocker des valeurs numériques : les nombres sont alors formatés pour être écrits en décimal à l'aide de caractères ASCII.

Remarques :

✚ Le retour chariot a 13 comme code ASCII.

✚ Le saut de ligne a 10 comme code ASCII.

Syntaxe :

```
Type
Nom_fichier= Fichier de caractères
Variables
F : Nom_fichier
```

Donc un fichier texte est une suite de caractère organisée en ligne.

Remarque :

- ✚ Un fichier teste peut être traité caractère par caractère ou ligne par ligne.
- ✚ Pour lire une ligne on utilise la procédure suivante Lire_lig(Nom_fichier, x) permet de lire une ligne du fichier et copier son contenu dans la variable x qui est de type chaine.

Exercice d'application 3 :

Écrire une procédure qui permet d'afficher le contenu d'un fichier texte.

Correction :

```
Type
Fichtxt=Fichier de caractère
Procédure Affichage( F : Fichtxt)
Variables
X :chaine
  Début
    Ouvrir(F, L)
    Lire_lig(F,X)
    Tant que (FDF(F)=faux) Faire
      Ecrire(X)
      Lire(F,X)
    Fin Tant que
    Fermer(F)
  Fin
```