ALGO

Ch 6 – Fichiers

I. Introduction

A. Notion de fichiers

Un **FICHIER** (anglais : file) est un REGROUPEMENT LOGIQUE DE DONNEES MEMORISEES SUR UN SUPPORT PERMANENT (disque dur, par exemple) afin de permettre une réutilisation ultérieure des informations qu'il contient.

B. Structuration des données dans un fichier

L'organisation des données dans un fichier correspond à la manière selon laquelle les données seront organisées à l'intérieur du fichier.

1. Fichiers NON structurés

On peut énumérer dans cette catégorie tous les fichiers de types documents, codes sources, etc.

Ils sont constitués d'un texte donc la structure n'est pas déterminée : on ne trouve pas de notion de données élémentaires. Le fichier doit être pris dans son ensemble.

Ces fichiers comportent néanmoins un format reconnu par un programme qui sera capable d'analyser son contenu et d'y trouver un sens. (les .doc sont lus par MS Word).

2. Fichiers structurés

Contrairement aux premiers, les fichiers structurés sont composés d'un ensemble de données élémentaires identifiables de manière précise au sein du fichier.

a) Structure déterminée par un balisage des données élémentaire

Dans cette catégorie, on trouvera les fichiers de type XML, HTML, Latex, etc. Chaque donnée élémentaire est encadrée par un balisage.

b) Structure organisée en enregistrements

Un enregistrement est un bloc de données élémentaires qui décrit une entité. Par exemple, au sein d'un fichier « Clients », un enregistrement correspond aux données relatives à un client.

Un champ est l'une des données élémentaires d'un enregistrement : le numéro de client est l'un des champs d'un enregistrement du fichier client (d'autres seront par exemple, la raison sociale, l'adresse, etc.).

Un enregistrement (*anglais : record*) est donc composé de plusieurs champs (*en anglais : field ou item*).

Il existe 2 manières d'isoler les champs dans un enregistrement :

- A l'aide d'un caractère séparateur de champs : les données sont codées en ASCII (ou autre format de caractères) ; un enregistrement correspond à une ligne d'un fichier texte
- A l'aide d'une définition précise de l'enregistrement et de ses champs grâce à un type structuré.

C. Résumé des caractéristiques des fichiers

Organisation	**Non structuré**	**Structuré en regroupement de données élémentaires**		
Délimitation des champs		**Balisage des données élémentaires**	**Enregistrements**	
			Séparateurs de champs	**Structure**
	.cpp, .doc	.xml, .html, structure définie dans une DTD	Fichiers de données au format texte (codification ASCII, par exemple) : .csv ou .tab (exportation des tableurs)	Fichiers de données au format binaire .bmp, fichiers de données

II. Les fichiers de données structurés en enregistrements

A. Détermination des champs de données

	Nombre de champs	***Longueur des champs***	***Délimitation des champs***	***commentaire***
Enregistrements avec séparateur de champs	fixe	variable	Un caractère séparateur est défini pour l'ensemble du fichier	On parle de « fichier plat » ou fichier texte avec séparateur
Enregistrements avec type structuré	fixe	fixe	Enregistrement sous forme de type structuré	Les données numériques sont codées en binaire et donc non lisibles

Exemple d'enregistrements d'un fichier texte avec caractère séparateur « ; » :

```
1,"Tim";"Burtley";"tb@free.fr"
2,"Max";"Ximum";"max.ximum@yahoo.fr"
```

Exemple d'enregistrements d'un fichier avec un type structuré :

```
000@Tim      Burtley    tb@free.fr          000çMax      Ximum
     max.ximum@yahoo.fr
```

B. Organisation du placement des données

L'organisation des données dans un fichier détermine comment seront placés chacun des enregistrements.

1. Organisation séquentielle :

Dans des fichiers séquentiels, les enregistrements sont mémorisés consécutivement dans l'ordre de leur entrée et peuvent seulement être lus dans cet ordre.

Si on a besoin d'un enregistrement précis dans un fichier séquentiel, il faut lire tous les enregistrements qui le précèdent, en commençant par le premier.

2. Organisation calculée

Dans des fichiers à placement calculé, les enregistrements sont placés à une position précise du fichier. Cette position est

- Soit donnée comme numéro d'ordre de l'enregistrement dans le fichier,
- Soit calculée selon un algorithme de placement appliqué à une clef (on parle de placement aléatoire, *en anglais : random*).

3. Organisation indexée

Dans des fichiers à organisation indexée, au moins 2 fichiers sont nécessaires pour chaque fichier géré :

- le premier fichier contient les données
- le second fichier contient une table d'index qui à une valeur d'une clef d'un enregistrement conserve la position du fichier à laquelle il se trouve.

La clef correspond à l'un des champs de l'enregistrement permettant l'identification d'un enregistrement unique au sein du fichier.

C. Modes d'accès aux enregistrements d'un fichier

L'accès aux enregistrements d'un fichier est déterminé par l'organisation de ce fichier.

1. Accès séquentiel :

L'accès séquentiel consiste à parcourir, dans l'ordre dans lequel ils sont stockés, les enregistrements d'un fichier. Pour accéder à un enregistrement, il faut avoir lu tous les enregistrements qui le précèdent.

L'accès séquentiel est effectué dans un seul « sens » : on parcourt les enregistrements du début à la fin, sans retour arrière possible.

2. Accès direct :

L'accès direct permet l'accès individuel à chacun des enregistrements d'un fichier, en y accédant directement :

- Soit grâce à un numéro d'ordre de placement ;
- Soit grâce à une clef.

L'accès direct permet d'accéder à plusieurs enregistrements directement, quelque soit l'emplacement de l'enregistrement dans le fichier.

D. Modes d'ouverture d'un fichier

Le mode d'ouverture d'un fichier va déterminer les actions que l'on sera autorisé à effectuer sur ce fichier.

Mode d'ouverture	***Actions autorisée***
lecture	En mode lecture, seules seront autorisés l'accès aux données, sans modification possible du contenu
écriture	En mode écriture, le fichier est vidé de son contenu, et ne sera possible que l'écriture de nouveaux enregistrements
ajout	En mode ajout, les données présentes dans le fichier seront préservées, et il sera possible d'ajouter de nouveaux enregistrements
lecture/écriture	*En mode lecture/écriture, les données présentes dans le fichier seront préservées, et il sera possible de modifier le contenu de certains enregistrements.*

E. Résumé

		Organisation - placement		
		séquentielle	**calculée**	**indexée**
Accès	**séquentiel**	Seul possible	Dans une boucle de calcul des clefs	Par parcours du fichier d'index
	direct		*naturel*	*Naturel*
Structure des enregistrements		Fichiers textes avec séparateurs ou structurés	Fichiers à type de donnée structuré	
Modes d'ouverture du fichier	**Lecture**	OUI	OUI	
	Ecriture	OUI	OUI	
	Ajout	OUI	OUI	
	modification	-	OUI	

III. Algorithmique – fichiers séquentiels

Le langage algorithmique (*et la plupart des langages de programmation*) met à disposition du programmeur un ensemble d'instructions permettant la manipulation des fichiers.

A. Déclaration d'une variable de type FICHIER

```
VAR fichier : FICHIER
```

B. Ouvrir un fichier séquentiel

```
fichier ← OUVRIR(nom_du_fichier , mode_d_Ouverture)
```

Avec mode_d_Ouverture parmi : LECTURE, ECRITURE, AJOUT

C. Lire un fichier et tester la fin du fichier

```
LIRE_FICHIER(fichier, liste_de_variables)
```

```
FF(fichier)
```

retourne VRAI si la fin du fichier a été atteinte (il n'y a plus d'enregistrements à lire)

D. Ecrire dans un fichier

```
ECRIRE_FICHIER(fichier, liste de variables)
```

Les données sont ajoutées soit à partir du début du fichier soit à partir de la fin (selon le mode ouverture)

E. Fermer un fichier

```
FERMER(fichier)
```

F. Exemple de programme

Exemple de programme:

```
. . .
VARIABLES
      f1, f2 : FICHIER
      num, nom, prenom, email : CHAINE
DEBUT
  f1 ← OUVRIR("fichier1.txt" , LECTURE)
  f2 ← OUVRIR("fichier2.txt" , ECRITURE)

  // première lecture
  LIRE_FICHIER(f1, num, nom, prenom, email)
  TANTQUE (NON FF(f1))
      ECRIRE_FICHIER(f2, num, nom, prenom, email)
      // autres lectures
      LIRE_FICHIER( f1, num, nom, prenom, email)
  FINTANTQUE

  FERMER(f2)
  FERMER(f1)
FIN
```

Exemple de programme 2 :

```
. . .
TYPE
  STRUCTURE enreg
    num : ENTIER
    nom : CHAINE
    email : CHAINE
  FIN STRUCTURE
VARIABLE
  f3 : FICHIER
  ligne : enreg
DEBUT
  f3 ← OUVRIR("fichier3.dat" , ECRITURE)
  ECRIRE("entrez le numéro, le nom et l'adresse email")
  LIRE(ligne.num, ligne.nom, ligne.email)
  ECRIRE_FICHIER(f3, ligne)
  FERMER(f3)
FIN
```