

Cours N°8

Les Fichiers de Type Texte

1. Introduction

- Le type fichier est un type structuré au même titre que le type tableau, à la différence que le tableau a une taille fixe et que le fichier est a priori illimité.
- Les types de variables étudiés précédemment sont utilisés pour décrire des informations stockées en mémoire centrale. La durée de vie de ce type d'information est égale au temps d'exécution du programme.
- Le type fichier permet la manipulation des informations stockées en une mémoire secondaire telle que le disque dur, clé USB ...

Définition d'un Fichier:

Un fichier est une suite de composantes de même type qui contient de l'information codée en binaires (bits), définis par un identificateur au même titre qu'une variable et enregistrée sur un support de mémoire non volatil.

- Les fichiers sont utilisés pour la communication entre un programme et son environnement, ou entre des programmes différents ; ils permettent aussi de conserver temporairement de grands volumes de données.
- Il existe différents types de fichiers. En ce qui nous concerne nous allons nous limiter aux **Fichiers de Type Texte**.

2. Fichiers Texte

- Un fichier de type texte est constitué d'une suite de caractères affichables et de caractères de contrôle groupés en lignes, comme dans un texte. Chaque ligne est terminée par une marque de fin de ligne qui est une séquence.
- Ces fichiers sont manipulés d'une manière séquentielle c-à-d l'accès à ce type de fichiers peut s'effectuer d'un élément à l'autre, en partant du premier.
- Les fichiers de type texte sont appelés aussi « fichiers ASCII » parce que leur contenu peut être visualisé à l'aide de n'importe quel éditeur de texte.
- Nous déclarons les fichiers de type texte, comme suit:

```
Var <nom_Logique_Fichier> : Text;
```

3. Opérations sur les Fichiers Texte

Les opérations sur les fichiers de type texte se font dans une séquence précise :

- 1) **Assignment du fichier**
- 2) **Ouverture du fichier en lecture, écriture ou ajout**
- 3) **Traitement du fichier – opérations de lecture/écriture**
- 4) **Fermeture du fichier**

1) **Assignment du fichier : (Assign) :**

- Un fichier doit avoir deux noms: un **nom logique** (Interne: en mémoire centrale) et un **nom physique** (Externe: sur le disque).
- L'assignation est l'association du nom logique avec le nom physique de ce fichier.

- Toutes les opérations effectuées sur un fichier, concernent le fichier situé sur le disque.

Syntaxe:

`Assign(<Nom Logique>, <Nom Physique>);`

Exemple:

`Assign(Fich, 'D:\LMD\Resulat.dat');`

 Nom logique

 Nom Physique

 Chemin d'accès

2) Ouverture du Fichier en Lecture, Écriture ou Ajout :

■ **Ouverture du fichier en Lecture: (Reset)**

- La procédure `Reset` permet d'**ouvrir un fichier existant**, sans écraser son contenu, et positionne le pointeur de fichier au début de ce dernier.
- Cette procédure permet d'ouvrir le fichier en lecture seulement.

Syntaxe: `Reset(<Nom Logique>);`

Exemple: `Reset(Fich);`

■ **Ouverture du fichier en Écriture: (Rewrite)**

- La procédure `Rewrite` permet la **création d'un nouveau fichier** ou réécrire complètement un fichier existant.
- Le pointeur de fichier se positionne automatiquement au début de ce dernier.

Attention: Si un fichier disque de même nom existe déjà, il est détruit et remplacé par un nouveau fichier vide.

Syntaxe: `Rewrite(<Nom Logique>);`

Exemple: `Rewrite(Fich);`

■ **Ouverture du fichier en mode Ajout: (Append)**

- C'est une procédure qui permet l'**ouverture d'un fichier en mode ajout**, c-à-d qu'on peut ajouter des lignes de texte à la fin du fichier.
- Après un appel à la procédure `Append`, le pointeur de fichier se positionne à la fin de ce dernier et seule l'écriture est autorisée.

Syntaxe: `Append(<Nom Logique>);`

Exemple: `Append(Fich);`

3) Lecture / Écriture dans un fichier :

■ Lecture dans un fichier:

– Les procédures `Read` ou `Readln` permettent de lire les valeurs de variables à partir d'un fichier.

Syntaxe:

```
Read(<Nom Logique>,<Liste de variables>);
```

■ Écriture dans un fichier:

– Les procédures `Write` ou `Writeln` permettent d'écrire les valeurs de variables dans d'un fichier.

Syntaxe:

```
Write(<Nom Logique>,<Liste de variables>);
```

4) Fermeture d'un fichier : (Close)

- `Close` est une procédure standard qui ferme un fichier ouvert.
- Un fichier fermé ne peut faire l'objet de transferts de données.
- Cette procédure permet la fermeture d'un fichier préalablement ouvert par `Reset`, `Rewrite` ou `Append`.
- Tout programme qui utilise des fichiers doit se termine par la fermeture des fichiers.

Syntaxe: `Close(<nom logique>);`

Exemple: `Close(Fich);`

Exemple:

```
Program Fichier;
uses wincrt;
var binf,bsup,i,N1,N2:integer;
    x,f:real;
    Fich1,Fich2:text;
BEGIN
  Assign(Fich1,'Don.dat');
  Reset(Fich1);
  Read(Fich1,binf,bsup,N1,N2);
  close(Fich1);
  Assign(Fich2,'Res.dat');
  Rewrite(Fich2);
  For i:=binf to bsup do
  Begin
    x:=i/10;
    f:=sqr(x);
    writeln(Fich2,x:7:2,' ',f:15:8);
  end;
  Close(Fich2);
  Assign(Fich2,'Res.dat');
  Append(Fich2);
  For i:=N1 to N2 do
  Begin
    x:=i/10;
    f:=sqr(x);
    writeln(Fich2,x:7:2,' ',f:15:8);
  end;
  Close(Fich2);
end.
```