Ministry of Higher Education and Scientific Research
Djilali BOUNAAMA University - Khemis Miliana(UDBKM)
Faculty of Science and Technology
Department of Mathematics and Computer Science

Chapter 1

Files

MI-L1-UEF121 : Algorithms and Data Structures II

Noureddine AZZOUZA

n.azzouza@univ-dbkm.dz

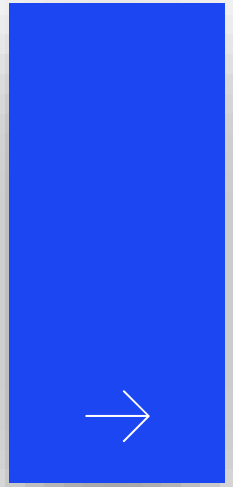
Course Topics

1. Files
    - 1.1 Definition
    - 1.2 Data Files and Text Files
    - 1.3 Operations on files
    - 1.4 Sequential access and direct access

Files

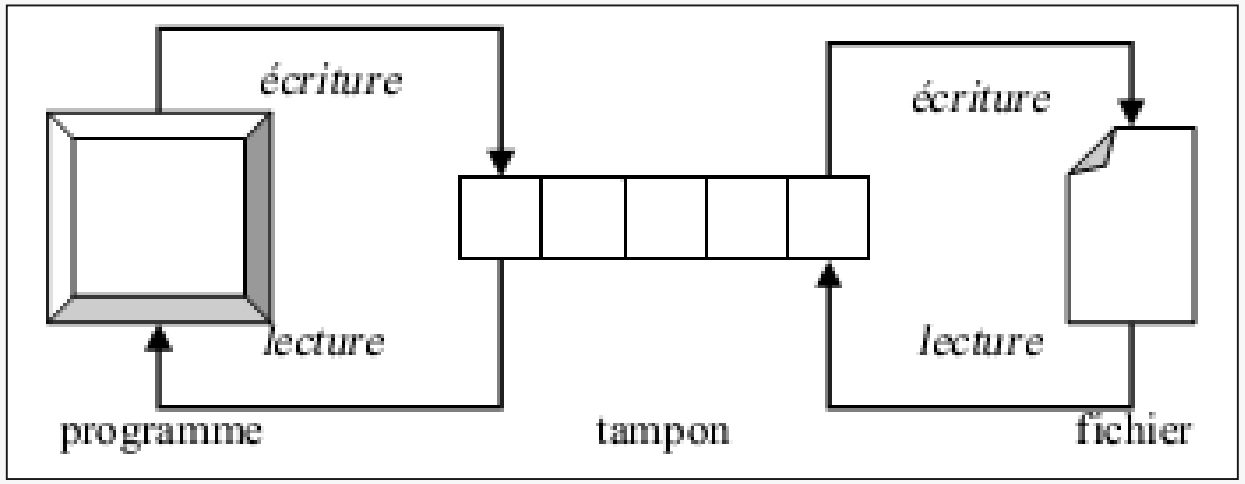
Definition

- ✓ A **file** is a set of data associated with a name and a location on a **storage device** (hard disk, USB key, optical disk, etc.).
- ✓ a file is not structured by nature, it is seen as a sequence of bytes.
- ✓ The information that we want to store can take two forms:
  - ➢ data in the form of components or articles
  - ➢ or text.



Access to files

- ✓ Access to files is buffered, that is to say, a buffer is used during writes and reads.
- ✓ buffer: area of memory used to manipulate data temporarily, before placing it elsewhere.



Notes

1. As we have already said, a file must be saved on external media, so this file must have a name and preferably an extension. This name is called the external name (or physical name)

2. The name of the object declared in the object table as file name is the internal name of the file (or also the logical name). This is the name used in the program instructions.

3. Data files have a .dat, .fch extension

Definition

- ✓ It is a set of components or articles dealing with the same subject and grouped under the same name.

F File

End Of File

Declaration

✓ The declaration of a data file is done by specifying the word "**FILE**" followed by the type of components. The latter can have any type (**elementary** or **structured**).

```
Type    type_name = File of type;
```

```
Type
    T : Array [1..20] of integer;
    F: File of T;

    contact= record;
        nom: string [25];
        Notel: integer;
      End
    r: File of contact;
```

```
Var
    F1: File of integer;
    Enr : F;
    Repertoire : r
```

Declaration : *Implementation*

PASCAL

```
Var  nom_var =  File of type;
```

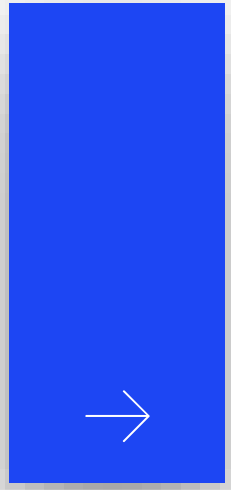
C

```
FILE      * nom_var ;
```

Definition

1. It is a sequence of characters from the ASCII game divided into lines, which can be of different lengths.

2. Each line ends with a "carriage return" possibly followed by a line feed.



Declaration

1. declaring a text file is simply done by specifying the “**TEXT**” type.

```
Var      var_name : Text;
```

```
Var
        book: text;
```

Declaration : *Implementation*

PASCAL

```
Var  nom_var =  Text;
```

C

```
FILE     * nom_var ;
```

1. Connection

- ✓ The link between internal (logical) file name and external (physical) file name.
- ✓ This connection will be made through an assignment procedure.

ASSIGN (Internal_Variable_file , External_file_name) ;

- ➢ *Internal_Variable_file* : Fichier
- ➢ *External_file_name* : String

2. Opening

After the assignment step, a file should be opened. Two cases are possible:

a) ***The file does not exist***: we will create it. So this is a new file. In this case the opening is done using the procedure

REWRITE (f)

b) ***The file already exists***: we want to consult or update it (creation of new components, deletion of components, modification of components) then the opening will be done by the procedure

RESET (f)

3. Closing

- When the processing of a file is finished, you must close the file:

CLOSE (f)

4. End of File

- We can detect the end of file using the following function:

EOF (f)

5. Seeking / Placing

- allows you to position the read/write head at a specific position in the file:

SEEK (f , pos)

5. Reading / Writing

- Read or write a component of a file using the WRITE and WRITELN, and READ and READLN procedures.
- to read or write to a file, simply specify its name.

Read (f , parameter1 , parameter2 ,……., parametern)

Write (f , parameter1 , parameter2 ,……., parametern)

1. Fichiers à accès séquentiel

1. Dans Files à accès séquentiel, les composants ou articles sont rangés les uns à la suite des autres de telle manière à ce que si l'on veut accéder à un composant nous sommes obligés de lire tous ceux qui se trouvent avant.

2. Lors de la lecture dans un fichier à accès séquentiel après les étapes d'assignation et d'ouverture, le pointeur se place automatiquement sur le premier composant (position 0) , et aussitôt que vous faites une lecture le pointeur se place automatiquement sur le composant suivant.

1. Fichiers à accès direct

1. Dans Files à accès direct, il est possible d'accéder directement à un composant. Dans ce cas la répartition des composants, lors de le création du fichier ou de leur recherche dans des opérations de consultation ou de mise à jour, se fait grâce à l'utilisation de formule de transformation de la forme ***F(indicatif) = Adresse physique*** ou adresse relative ou grâce à des tables d'index dans lesquelles on aura des couples (indicatif , position).

2. Dans Files à accès direct, une fois que vous avez la position du composant (qui est fournie par votre formule ou votre table index) vous pouvez y accéder directement en utilisant la procédure ***SEEK (f , POS).***