# Chapter2:Files

## 2.1 Introduction

All the data used until now was stored in central memory, that is to say in a volatile area of the machine (RAM). The life time of the variables and manipulated information was therefore equal to the execution time of the program.

In a company, information must have a life span longer than the execution of a program. They are therefore stored on non-volatile media (e.g. hard disks, flash disks, CD…).The algorithms can manipulate the information in these files usingspecial instructions.

## 2.2 Definition of a file

A computer file is, in the common sense, a collection or a set of digital data (binary form 0.1) gathered under the same name, saved on a permanent storage media, called mass memory, such as a hard disk, a CD -ROM, memoryflash.., and manipulated as a unit.

A file has a name that is used to designate and access the content. This name often includes a suffix called the extension, which provides information on the nature of the information contained in the file and therefore the software that can be used to manipulate it.

The essence of the file is the information it contains. File format is the convention by which information is scanned and sequenced in the file. The file format is proprietary when the convention is known only to its author and has never been published. The file format is open when the convention is made public in order to allow the interoperability of the software handling it. Depending on the nature and format of thecontent, files can be classified as executable, compressed, text, document, image, audio or video, the following table shows some file types:

| **Name** | **Natureofcontent** | **Extensions** |
|---|---|---|
| executable | Files that can be executed by the computer - in other words, programs. | .exe,.com,.sh, .bat,... |
| compressed | Files encoded using a process that makes them smaller than the original; not coded files. A program *decompressor* is required to perform a reverse coding And find Thus the original file. | .bow,.zip,.rar, .z, .arj, .sit, .gz,.7z,... |
| pictures | Files that contain images and sound in a computer-readable form. Such files may contain pictures, pictograms, graphics, | .gif, .jpg, .bmp, .png, .eps,.tif, ... |

| | | |
|---|---|---|
| audio | songs, music, of the broadcasts radio or movies. | .au, .wav, .mp3, oga, .ram,... |
| video | | .avi, .mpg, .mov,... |
| documents | written documents, intended to be printed and read. The file contains the text as well as typography information (fonts, colors). | .docx, .odt, .html,.doc,... |
| text | *raw* text files contain written texts, without typographical indications. These can be texts intended for users, such as instructions for use or drafts; or of texts intended has the computer such that of coded source or data for a program. | .txt,.html,.ini, .csv, .log, .conf, .cpp,... |

In algorithmic, a file is a set of data of the same type, recorded on a permanent storage medium. It is often to use text or record type (structure) to store this data.

## 2.3 Types of files

In algorithmic, we have mostly two types of files:

- **The text file:** They are made up of a series of characters forming a text (character string). They are used to record texts but also numerical values with a view to exchanging them with other software. They are readable by a simple text editor.
- **The binary file:** containing data in the form of bytes which therefore only have meaning for the software which uses them, this type of file is unreadable by a text editor, it is made up of a collection of records, each record containing a collection of logical units of information also called **fields**.

## 2.4 Handlingof thefiles

Most current programming languages have instructions for manipulating files. These instructions can be classified as follows:

- Opening and creation of a file,
- Closing of a file
- Reading And writing of recordings of file,
- Positioning in the file,
- Detection of the **E**ND **o**f **f**ile.

**Important notice:**

To use a physical file **F** in an algorithm, this algorithm had to include a file variable **f**. The association between **f** and **F** will therefore be carried out by means of a process called **assignment**, such that modifications made to **f** in the algorithm will directly affect **F** on its support.

Syntax: **ASSIGN** ( **f, 'path_access_F'** );

**a) Opening and creation of a file**

After assignment, you must always open a file to be able to use it. The file open instruction must indicate whether the file will be read, modified or created.

## Syntaxes:

- **OPEN (f);** // if **F** is already existing file on disk (opening for reading).
- **REWRITE(f);**// if we want to create a**file F**which does not yet exist on disk, or which already exists but whose we want to overwriteentire content (open for writing).
- **ADD (f);** // if we want to open a file **F** in order to add data (only at the end of the file).

**Noticed:**

- **Open()** makes **F** readable and positions the pointer in the beginning of the file.

- In the case where **F** does not exist on the disk, **REWRITE ( )** allows you to create a file whose name and access path are those specified during the assignment.

- when we open a file that contains any data, the begin of file will then be the end of file. The Boolean function **Eof (f)** (**E**nd **o**f **f**ile) allows you to check at any time whether the pointer has reached the end of the file or not.

**b) Reading And writing of recordings of file**

After opening the file, several possibilities are offered: read the information it contains, modify some of it, delete some of it, or add others.

**Syntaxes :**

**Read (f, p);** to read a record from **F** and insert the data it contains into the variable **p** (p is of type record).

**Write (f, p);** to insert into **F** a record having the data contained in the variable **p** (p is of record type).

**Noticed:**

After reading/writing a record, the pointer immediately positions itself on the next record. If there are no more, the pointer position then becomes the end of file.

**c) Closing of a file**

Closing is an essential operation in file manipulation. A good programmer must therefore never neglect it, because it helps avoid input/output errors,and preserve the integrity of the data in a file, and optimize an algorithm using a minimum of internal variables with a maximum of external files.

**Syntax: CLOSE** (f);

**Example:**

Write an algorithm who allows displaying all the values contained in a file of integers F.

**Algorithm** files

p: integer;

f : file of integer;

**Begin**

Assign ( **f**,' **c:\integers.txt** ');

Open ( **f** );

While (not( **Eof(f)** ))do

Read (f, p);

Write ('We are reading the number ',p);

end while

Close (f);

**END**.