Programming and Problem Solving through C Language
O Level / A Level

Chapter - 11 : File Processing

Concept of Files

- Many real-world problems handle large volume of data and in such situations external storage devices like the floppy disk and the hard disks are used.
- Data is stored in these devices using the concept of files.
- A file is a collection of related data stored on a particular area of the disk.

Filenames

- Every disk file has a name, and one must use filenames when dealing with disk files.
- Filenames are stored as strings, just like other text data.
- The rules as to what is acceptable for filenames and what is not differ from one operating system to another.

Types of Files

1. **Text files**
   - Text files are the normal **.txt** files. The text files are easily created using any simple text editors such as Notepad.
   - We can see all the contents within the file as plain text and the contents can be modified.
   - These files require minimum effort to maintain, are easily readable, and provide the least security and takes bigger storage space.

2. **Binary files**
   - Binary files are mostly the **.bin** files in the computer.
   - Instead of storing data in plain text, they store it in the binary form (0's and 1's).
   - These files can hold a higher amount of data, are not readable easily, and provides better security than text files.

File Operations

Operations allowed on files, either text or binary are:

1. Creating a new file
2. Opening an existing file
3. Closing a file
4. Reading from and writing information to a file

Opening a file

- The process of creating a stream linked to a disk file is called opening the file.
- When one opens a file, it becomes available for reading (meaning that data is input from the file to the program), writing (meaning that data from the program is saved in the file), or both.
- After working with the file, close the file.
- To open a file, use the **fopen**( ) library function.

```
FILE *fptr;
fptr = fopen("fileopen", "mode");
```

- **FILE** is special data type to handle the File Stream pointer.
- The prototype of fopen( ) is located in STDIO.H and it reads as follows:
- This prototype tells the programmer that fopen() returns a pointer to type FILE, which is a structure declared in STDIO.H.
- The members of the FILE structure are used by the program in the various file access operations.

Example – 1

```
#include<stdio.h>
#include<stdlib.h>

void main ( )
{       FILE *fp;
        fp=fopen("abc.txt","w");
        fprint(fp, "%s %d\n","Ajay Kumar",30);
        fclose(fp);
}
```

Example -2

```
#include<stdio.h>
#include<stdlib.h>

int main ( )
{       FILE *fp;
        fp=fopen("abc.txt","r");
        if( fp==NULL)
         { printf("Error : Can't open file");
           return 1;
         }
        else
         { printf("File Opened.. Sucessfully");
           fclose(fp);
           return 0;
         }
}
```

- For each file that has to be open, one must declare a pointer to type FILE.
- The fopen(), that function creates an instance of the FILE structure and returns a pointer to that structure.
- A programmer uses this pointer in all subsequent operations on the file.
- If fopen() fails, it returns NULL.
- The argument filename is the name of the file to be opened.
- As noted earlier, filename can—and should--contain a path specification.
- The filename argument can be a literal string enclosed in double quotation marks or a pointer to a string variable.
- The argument mode specifies the mode in which to open the file.
- In this context, mode controls whether the file is binary or text and whether it is for reading, writing, or both.
- The default file mode is text.

Modes of Opening File

Mode	Meaning of Mode	During Inexistence of file
r	Open for reading.	If the file does not exist, fopen() returns NULL.
rb	Open for reading in binary mode.	If the file does not exist, fopen() returns NULL.
w	Open for writing.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
wb	Open for writing in binary mode.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
a	Open for append. Data is added to the end of the file.	If the file does not exist, it will be created.
ab	Open for append in binary mode. Data is added to the end of the file.	If the file does not exist, it will be created.
r+	Open for both reading and writing.	If the file does not exist, fopen() returns NULL.
rb+	Open for both reading and writing in binary mode.	If the file does not exist, fopen() returns NULL.
w+	Open for both reading and writing.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
wb+	Open for both reading and writing in binary mode.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
a+	Open for both reading and appending.	If the file does not exist, it will be created.
ab+	Open for both reading and appending in binary mode.	If the file does not exist, it will be created.

Closing a File

- fclose() function in File Handling.
- When a programmer opens a file in read or write mode then he performs appropriate operations on file and when file is no longer needed then he closes the file.
- FILE close will flush out all the entries from buffer.
- Syntax

  fclose(FILE *fp);

Example

```
 #include<stdio.h>
#include<stdlib.h>

void main ( )
{       FILE *fp;
        fp=fopen("abc.txt","w");
        fprint(fp, "%s %d\n","Ajay Kumar",30);

        fclose(fp);
}
```