C Programming Files I/O

In C programming, file is a place on disk where a group of related data is stored.

Why files are needed?

When the program is terminated, the entire data is lost in C programming. If you want to keep large volume of data, it is time consuming to enter the entire data. But, if file is created, these information can be accessed using few commands.

There are large numbers of functions to handle file I/O in C language. In this tutorial, you will learn to handle standard I/O(High level file I/O functions) in C.

High level file I/O functions can be categorized as:

1. Text file
2. Binary file

File Operations

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

Working with file

While working with file, you need to declare a pointer of type file. This declaration is needed for communication between file and program.

FILE \*ptr;

Opening a file

Opening a file is performed using library function fopen(). The syntax for opening a file in standard I/O is:

ptr=fopen("fileopen","mode")

For Example:

fopen("E:\\cprogram\program.txt","w");

/\* --------------------------------------------------------- \*/

E:\\cprogram\program.txt is the location to create file.

"w" represents the mode for writing.

/\* --------------------------------------------------------- \*/

Here, the program.txt file is opened for writing mode.

| Opening Modes in Standard I/O | | |
| --- | --- | --- |
| File Mode | Meaning of Mode | During Inexistence of file |
| r | Open for reading. | 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. |
| a | Open for append. i.e, Data is added to end of file. | If the file does not exists, it will be created. |
| r+ | Open for both reading and writing. | 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. |
| a+ | Open for both reading and appending. | If the file does not exists, it will be created. The reading will start from the beginning but writing can only be appended. |

Closing a File

The file should be closed after reading/writing of a file. Closing a file is performed using library function fclose().

fclose(ptr); //ptr is the file pointer associated with file to be closed.

The Functions fprintf() and fscanf() functions.

The functions fprintf() and fscanf() are the file version of printf() and fscanf(). The only difference while using fprintf() and fscanf() is that, the first argument is a pointer to the structure FILE

Writing to a file

#include <stdio.h>

int main()

{

int n;

FILE \*fptr;

fptr=fopen("C:\\program.txt","w");

if(fptr==NULL){

printf("Error!");

exit(1);

}

printf("Enter n: ");

scanf("%d",&n);

fprintf(fptr,"%d",n);

fclose(fptr);

return 0;

}

This program takes the number from user and stores in file. After you compile and run this program, you can see a text file program.txt created in C drive of your computer. When you open that file, you can see the integer you entered.

Similarly, fscanf() can be used to read data from file.

Reading from file

#include <stdio.h>

int main()

{

int n;

FILE \*fptr;

if ((fptr=fopen("C:\\program.txt","r"))==NULL){

printf("Error! opening file");

exit(1); /\* Program exits if file pointer returns NULL. \*/

}

fscanf(fptr,"%d",&n);

printf("Value of n=%d",n);

fclose(fptr);

return 0;

}

If you have run program above to write in file successfully, you can get the integer back entered in that program using this program.

Other functions like fgetchar(), fputc() etc. can be used in similar way.

fopen

fopen( ) performs three important tasks when you open the file in “r” mode:

(a) Firstly it searches on the disk the file to be opened.

(b) Then it loads the file from the disk into a place in memory called buffer.

(c) It sets up a character pointer that points to the first character of the buffer

fclose

fclose( ) three operations would be performed:

(a) The characters in the buffer would be written to the file on the disk.

(b) At the end of file a character with ASCII value 26 would get written.

(c) The buffer would be eliminated from memory.

Reading from a file – fgetc()

Once the file has been opened for reading using fopen( ), as we have seen, the file’s contents are brought into buffer (partly or wholly) and a pointer is set up that points to the first character in the buffer. This pointer is one of the elements of the structure to which fp is pointing

To read the file’s contents from memory there exists a function called **fgetc( )**. This has been used in our program as,

ch = fgetc ( fp ) ;

fgetc( ) reads the character from the current pointer position, advances the pointer position so that it now points to the next character, and returns the character that is read, which we collected in the variable ch.

What is EOF

A special character, whose ASCII value is 26, signifies end of file. This character is inserted beyond the last character in the file, when it is created

Writing to a file – fputc()

| Opening Modes in Standard I/O | | |
| --- | --- | --- |
| 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. i.e, Data is added to end of file. | If the file does not exists, it will be created. |
| ab | Open for append in binary mode. i.e, Data is added to end of file. | If the file does not exists, 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 exists, it will be created. |
| ab+ | Open for both reading and appending in binary mode. | If the file does not exists, it will be created. |