Enjoy with files

When a program runs, the data is in the memory but when it ends or the computer shuts down, it gets lost. To keep data permanently, we need to write it in a file.

File is used to store data. In this topic, we will learn about reading data from a file and writing data in the file.

First of all, let's see how to declare a file.

File is declared using a pointer of type file as follows.

**FILE \*fr;**

We can perform many functions with files. Some of these functions are listed below.

|  |  |
| --- | --- |
| **Function** | **Description** |
| fopen() | create a new file or open a existing file |
| fclose() | close a file |
| fgetc() | read a character from a file |
| fputc() | write a character to a file |
| fscanf() | read a string from a file |
| fprintf() | write a string to a file |

Let’s see how to write data in, open, close or read a file.

Opening a file

We open a file or create a new file using **fopen()** function. Let's look at the **syntax** of fopen().

***#include*** *<stdio.h>*

***int*** *main()*

*{*

***FILE*** *\*fr;*

*fr = fopen("filename", "mode");*

***return*** *0;*

*}*

Here, **fr** is a pointer to file, **filename** is the name of the file which we want to open and **mode** is the reason for which we want to open the file.

Mode can take the following values.

|  |  |
| --- | --- |
| **Mode** | **Description** |
| r | opens a text file for reading |
| w | opens a text file for writing |
| a | opens a text file for appending. (appending means to add text at the end) |
| r+ | opens a text file for reading and writing |
| w+ | opens a text file for both reading and writing. (it first cuts the length of the file to zero if it exists, or create a file if it does not exist) |
| a+ | opens a text file in both reading and writing mode. (It creates a file if it does not exist. Reading will start from the beginning but writing can only be done at the end) |

These were the modes of the text file. If we have to deal with binary files, then the modes are as follows:

**rb**, **wb**, **ab**, **rb+**, **wb+**, **ab+**

However in our tutorial, we will just deal with text files.

When we use 'fopen', then it searches the file on the disk and then loads the file on memory called buffer. The changes we made before closing are made in buffer memory. Buffer memory saves from accessing the disk again and again while making changes.

Closing a file

We close an opened file using **fclose()** function.

The **syntax** of fclose() is as follows.

***#include*** *<stdio.h>*

***int*** *main()*

*{*

***FILE*** *\*fr;*

*fclose(fr);*

***return*** *0;*

*}*

fclose() function returns **zero** if a file is successfully closed. If there is some error in closing the file, **EOF** ( End of File ) is returned.

We have just seen how to open and close a file. Now, we will see how to read and write in a file.

Reading and Writing on a file

The functions which we use for reading and writing depends on whether we are dealing with characters or strings.

For example, functions fgetc() and fputc() are used for reading and writing characters in a file. Similarly, functions fscanf() and fprintf() are used for reading and writing strings in a file.

Let's see an example for each and then understand it line by line.

Using fgetc() and fputc()

**fgetc()** reads a character from a file opened in **read mode** using **fopen()**. It returns **EOF** on reaching the end of the file.

**fputc()** writes a character to a file opened in **write mode** using **fopen()**.

Now let's see an example of reading from file.

***#include*** *<stdio.h>*

***int*** *main()*

*{*

***FILE*** *\*fr;*

***char*** *c;*

*fr = fopen("prog.txt", "r");*

***while****( c != EOF)*

*{*

*c = fgetc(fr); /\* read from file\*/*

*printf("%c",c); /\*  display on screen\*/*

*}*

*fclose(fr);*

***return*** *0;*

*}*

**Output**

This is test   
These are content in prog.txt in my system.

In the above example, firstly we opened a file named 'prog.txt' in read mode so that we can read data from the file. For this, 'prog.txt' must exist in your computer.

In while loop, **fgetc(fr)** reads the value of c from the file 'prog.txt'. When the entire value of 'c' has been read from the file and **EOF** (End of file) is reached, **fclose(fr)** closes the file.

As simple as that !

We read the contents of the file by **fgetc()** using variable 'c'. In while, printf function prints the value of c on the screen. After this, we will close the file using **fclose**.

Using fscanf() and fprintf()

While getc and putc are used for characters, we use **fscanf()** and **fprintf** to read and write string to a file.

Let's understand with the help of an example.

***#include*** *<stdio.h>*

***int*** *main()*

*{*

***FILE*** *\*fr;*

***char*** *s[40];*

***int*** *n;*

*fr = fopen("welcome.txt", "w");*

*printf("Enter a string and a number");*

*scanf("%s%d", s, &n); /\*read from keyboard\*/*

*fprintf(fr, "%s %d", s, n); /\*  write to file\*/*

*fclose(fr);*

*fr = fopen("welcome.txt", "r");*

*fscanf(fr, "%s%d", s, &n); /\*read from file\*/*

*printf("%s %d", s, n); /\*display on screen\*/*

*fclose(fr);*

***return*** *0;*

*}*

**Output**

Here, we first created a file named 'welcome.txt' and opened it in write mode using **fopen** function. Then, we input a string and an integer value from the keyboard using **scanf** function and stored the values in variables 's' and 't' respectively.

Then, **fprintf** writes both the string and integer values to the file. After that, **fclose(fr)** closes the file.

Now, we will read the contents of the file and then print those on the screen. For that, we will first open the file in read mode using fopen function. After that, **fscanf** function will read the contents of the file and **printf**function will display the contents of the file on the screen.

Now see this program to copy one file to another.

***#include*** *<stdio.h>*

***int*** *main()*

*{*

***FILE*** *\*s,\*t;*

***char*** *c;*

*s = fopen("source.txt", "r");*

*t = fopen("target.txt", "w");*

***while****(1){*

*c = fgetc(s);*

***if****(c == EOF)*

*{*

***break****;*

*}*

***else***

*{*

*fputc(c,t);*

*}*

*}*

*fclose(s);*

*fclose(t);*

***return*** *0;*

*}*

**Output**

'fputc(c,t)' will put the character stored in 'c' to the file opened as 't'.

This is some impressive work, you have just written the code to copy two files. For this program to work, you must have a file named **source.txt**.