# C – File Handling

# ****WHAT IS FILE?****

File is a collection of bytes that is stored on secondary storage devices like disk. There are two kinds of files in a system. They are,

1. Text files (ASCII)
2. Binary files

* Text files contain ASCII codes of digits, alphabetic and symbols.
* Binary file contains collection of bytes (0’s and 1’s). Binary files are compiled version of text files.

# BASIC FILE OPERATIONS IN C PROGRAMMING:

There are 4 basic operations that can be performed on any files in C programming language. They are,

1. Opening/Creating a file
2. Closing a file
3. Reading a file
4. Writing in a file

Let us see the syntax for each of the above operations in a table:

|  |  |
| --- | --- |
| **File operation** | **Declaration & Description** |
| **fopen()** – To open a file | Declaration: FILE \***fopen**(const char \*filename, const char \*mode)  fopen() function is used to open a file to perform operations such as reading, writing etc. In a C program, we declare a file pointer and use fopen() as below. fopen() function creates a new file if the mentioned file name does not exist.  FILE \*fp; fp=**fopen**(“filename”, ”‘mode”);  Where, fp – file pointer to the data type “FILE”. filename – the actual file name with full path of the file. mode – refers to the operation that will be performed on the file. Example: r, w, a, r+, w+ and a+. Please refer below the description for these mode of operations. |
| **fclose()**– To close a file | Declaration: int **fclose**(FILE \*fp);  fclose() function closes the file that is being pointed by file pointer fp. In a C program, we close a file as below. **fclose**(fp); |
| **fgets()** – To read a file | Declaration: char \***fgets**(char \*string, int n, FILE \*fp)  fgets function is used to read a file line by line. In a C program, we use fgets function as below. **fgets** (buffer, size, fp);  where, buffer – buffer to  put the data in. size – size of the buffer fp – file pointer |
| **fprintf()**– To write into a file | Declaration: int **fprintf**(FILE \*fp, const char \*format, …);fprintf() function writes string into a file pointed by fp. In a C program, we write string into a file as below.fprintf (fp, “some data”); or fprintf (fp, “text %d”, variable\_name); |

# ****MODE OF OPERATIONS PERFORMED ON A FILE IN C LANGUAGE:****

There are many modes in opening a file. Based on the mode of file, it can be opened for reading or writing or appending the texts. They are listed below.

* r – Opens a file in read mode and sets pointer to the first character in the file. It returns null if file does not exist.
* w – Opens a file in write mode. It returns null if file could not be opened. If file exists, data are overwritten.
* a – Opens a file in append mode.  It returns null if file couldn’t be opened.
* r+ – Opens a file for read and write mode and sets pointer to the first character in the file.
* w+ – opens a file for read and write mode and sets pointer to the first character in the file.
* a+ – Opens a file for read and write mode and sets pointer to the first character in the file. But, it can’t modify existing contents.

# ****1. EXAMPLE PROGRAM FOR FILE OPEN, FILE WRITE AND FILE CLOSE IN C LANGUAGE:****

C



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | / \* Open, write and close a file : \*/  # include <stdio.h>  # include <string.h>    int main( )  {      FILE \*fp ;      char data[50];      // opening an existing file      printf( "Opening the file test.c in write mode" ) ;      fp = fopen("test.c", "w") ;      if ( fp == NULL )      {          printf( "Could not open file test.c" ) ;          return 1;      }      printf( "\n Enter some text from keyboard” \               “ to write in the file test.c" ) ;      // getting input from user      while ( strlen ( gets( data ) ) > 0 )      {          // writing in the file          fputs(data, fp) ;          fputs("\n", fp) ;      }      // closing the file      printf("Closing the file test.c") ;      fclose(fp) ;      return 0;  } |

#### ****OUTPUT:****

|  |
| --- |
| Opening the file test.c in write mode  Enter some text from keyboard to write in the file test.c Hai, How are you? Closing the file test.c |

Let’s do this with the help of a text file. We are going to write a set of data file1.txt and then copy it to file2.txt

1. #include <stdio.h>
2. **void** fn(){
3. FILE \*f1, \*f2;
4. **char** s[10], s2[10];
5. f1=fopen("file1.txt","w");
6. **for**(**int** i=0;i<3;i++){
7. gets(s);
8. fprintf(f,"%s",s);
9. }
10. fclose(f1);
12. f1=fopen("file.txt","r");
13. f2=fopen("file2.txt", "w");
15. **while**(s1!=EOF){
16. fscanf(f1,"%s",s2);
17. fprintf(f2,"%s",s2);
18. }
19. fclose(f1); fclose(f2);
20. }
22. **int** main(){
23. fn();
25. **return** 0;
26. }

Closing a File The fclose() function is used to close an already opened file. General Syntax : int fclose( FILE \*fp ); Here fclose() function closes the file and returns zero on success, or EOF if there is an error in closing the file. This EOF is a constant defined in the header file stdio.h.

#### Input/Output operation on File

In the above table we have discussed about various file I/O functions to perform reading and writing on file.getc() and putc() are simplest functions used to read and write individual characters to a file.

#include<stdio.h>

#include<conio.h>

main()

{

FILE \*fp;

char ch;

fp = **fopen**("*one.txt*", "*w*");

printf("Enter data");

while( (ch = getchar()) != EOF) {

**putc**(ch,fp);

}

fclose(fp);

fp = **fopen**("*one.txt*", "*r*");

while( (ch = **getc**(fp)! = EOF)

printf("%c",ch);

fclose(fp);

}

#### Reading and Writing from File using fprintf() and fscanf()

#include<stdio.h>

#include<conio.h>

struct emp

{

char name[10];

int age;

};

void main()

{

struct emp e;

FILE \*p,\*q;

p = **fopen**("*one.txt*", "*a*");

q = **fopen**("*one.txt*", "*r*");

printf("Enter Name and Age");

scanf("%s %d", e.name, &e.age);

**fprintf**(p,"%s %d", e.name, e.age);

fclose(p);

do

{

**fscanf**(q,"%s %d", e.name, e.age);

printf("%s %d", e.name, e.age);

}

while( !feof(q) );

getch();

}

**fseek(), ftell() and rewind() functions**

* **fseek()** - It is used to move the reading control to different positions using fseek function.
* **ftell()** - It tells the byte location of current position of cursor in file pointer.
* **rewind()** - It moves the control to beginning of the file.

## Program to Find Size of a File

We will be using fseek() and ftell() functions to find the size of the file. There are others ways to find the file size, like looping on the whole content of file and finding out the size, but using File Handling functions makes it easier.

#include<stdio.h>

#include<conio.h>

void main()

{

FILE \*fp;

char ch;

int size = 0;

fp = fopen("MyFile.txt", "r");

if (fp == NULL)

{

printf("\nFile unable to open ");

}

else

{

printf("\nFile opened ");

}

fseek(fp, 0, 2); /\* file pointer at the end of file \*/

size = ftell(fp); /\* take a position of file pointer un size variable \*/

printf("The size of given file is : %d\n", size);

fclose(fp);

}

### Program to Copy Content of One File into Another File

We already know how to open a file, read contents of a file and write into a file. So in this program, we will read from one file and simultaneously write into the other file, till we reach end of first file.

#include<stdio.h>

void main()

{

*/\**

*File\_1.txt is the file with content and,*

*File\_2.txt is the file in which content of File\_1*

*will be copied.*

*\*/*

FILE \*fp1, \*fp2;

char ch;

int pos;

if ((fp1 = fopen("File\_1.txt","r")) == NULL)

{

printf("\nFile cannot be opened");

return;

}

else

{

printf("\nFile opened for copy...\n ");

}

fp2 = fopen("File\_2.txt", "w");

**fseek**(fp1, 0L, SEEK\_END); *// file pointer at end of file*

pos = **ftell**(fp1);

fseek(fp1, 0L, SEEK\_SET); *// file pointer set at start*

while (pos--)

{

ch = **fgetc**(fp1); *// copying file character by character*

**fputc**(ch, fp2);

}

fcloseall();

}