**File Operations**

There are different operations that can be carried out on a file. These are:

* Creation of a new file
* Opening an existing file
* Reading from a file
* Writing to a file
* Moving to a specific location in a file (seeking)
* Closing a file

To perform this operation C provides set of file handling functions which are available is <stdio.h>.

**Function Use**

fopen Opens a file

Close Closes a file

fclose Closes a file

feof Detects end-of-file

fgetc Reads a character from a file

fputc Writes a character to a file

fgets Reads a string from a file

fputs Writes a string to a file

fputchar Writes a character to screen (function version)

fgetchar Reads a character from keyboard (function version)

fprintf Writes formatted data to a file

fscanf Reads formatted data from a file

fseek Repositions file pointer to given location

ftell Gets current file pointer position

getch Reads a character from the keyboard

getche Reads a character from keyboard and echoes it

getchar Reads a character from keyboard (macro version)

File Pointer: - File pointer is a pointer to a structure FILE, which keeps track of which file we are accessing. The type FILE is predefined structure in <stdio.h>. It is a data structure which holds the information the standard I/O library needs to keep track of the file for you.

You declare a variable to store a file pointer like this:

**FILE \*fp;**

 If we are reading from two files or reading from one file and writing to another then we use two file pointers:

FILE \*fp1, \*fp2;

**Opening a File**

Before we can read (or write) information from (to) a file on a disk we must open the file. To open the file we have called the function **fopen( )**. fopen() takes two arguments. First one is a string containing name of the file to be opened and second one is mode value.

**fopen( )** returns the address of the file, which we have collected in the structure pointer called **fp**. It returns NULL, if it is unable to open the file.

FILE \*fp ;

fp = fopen ( "file name", "mode" ) ;

example: - if we want to open file Sample.txt in read mode , then we have to write the following lines of code

FILE \*fp ;

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

Different modes of file

|  |  |
| --- | --- |
| Mode | Meaning |
| r | Opens a file for reading |
| w | Create a file for writing |
| a | Append to a file |
| r+ | Opens a file for reading/writing |
| w+ | Create a file for reading/writing |
| a+ | Append or create a file for reading/writing |

**Writing to a file: -**

To perform write operation we have to open the file in “w” mode. We can write a file using any of the following function fputc(), fputs() and fprintf().

**Reading a file: -**

To read from a file the function fgetc(), fgets() and fscanf() are used.

**Closing a File: -**

When we have finished reading from the file, we need to close it. This is done using the function **fclose( )** through the statement,

fclose ( fp ) ;

/\*Display contents of a file on screen. \*/

# include <stdio.h>

main( )

{

FILE \*fp ;

char ch ;

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

if(fp==NULL){

printf("cannot open");}

while ( 1 )

{

ch = fgetc ( fp ) ;

if ( ch == EOF )

break ;

printf ( "%c", ch ) ;

}

fclose ( fp ) ;

}

Write a program to copy the content of a file “Sample.txt” to another file “output.txt”.

Note:- we have to create Sample.txt before opening it.

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

int main() {

FILE \*fp1, \*fp2;

char ch;

clrscr();

fp1 = fopen("Sample.txt", "r");

if(fp==NULL){

printf("cannot open");

return(0);

}

fp2 = fopen("Output.txt", "w");

while (1) {

ch = fgetc(fp1);

if (ch == EOF)

break;

putc(ch, fp2);

}

printf("\nCOMPLETED");

getch();

}

\*\*EOF means end of file. Every terminated with EOF special character.

Write a program to copy the content of a file “Sample.txt” to another file “output.txt” using command line argument.

#include<stdio.h>

int main(int argc,char \*argv[])

{

FILE \*fp1,\*fp2;

char ch;

if(argc!=3)

{

printf("Invalide numbers of arguments.");

return(0);

}

fp1=fopen(argv[1],"r");

if(fp1==NULL)

{

printf("Can't find the source file.");

return(0);

}

fp2=fopen(argv[2],"w");

if(fp2==NULL)

{

printf("Can't open target file.");

return(0);

}

while(1)

{

ch=fgetc(fp1);

if (ch==EOF)

break;

fputc(ch,fp2);

}

}