

File Handling in C Language

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

C provides a number of functions that helps to perform basic file operations. Following are the functions,

Function	description
fopen()	create a new file or open a existing file
fclose()	closes a file
getc()	reads a character from a file
putc()	writes a character to a file
fscanf()	reads a set of data from a file
fprintf()	writes a set of data to a file
getw()	reads a integer from a file
putw()	writes a integer to a file
fseek()	set the position to desire point

File Handling in C Language

Opening a File or Creating a File

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;

General Syntax : *fp = FILE *fopen(const char *filename, const char *mode);

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.

Closing a file

fclose() function closes the file that is being pointed by file pointer fp. In a C program, we close a file as below.

General Syntax : int fclose(FILE *fp);
fclose (fp);

fprintf()

fprintf() function writes string into a file pointed by fp. In a C program, we write string into a file as below.

General Syntax : int fprintf(FILE *fp, const char *format, ...);

e.g. fprintf (fp, "some data");

Mode of operations performed on a file

mode	description
r	opens a text file in reading mode
w	opens or create a text file in writing mode.
a	opens a text file in append mode
r+	opens a text file in both reading and writing mode

File Handling in C Language

w+	opens a text file in both reading and writing mode
a+	opens a text file in both reading and writing mode
rb	opens a binary file in reading mode
wb	opens or create a binary file in writing mode
ab	opens a binary file in append mode
rb+	opens a binary file in both reading and writing mode
wb+	opens a binary file in both reading and writing mode
ab+	opens a binary file in both reading and writing mode

Input/Output operation on File

// Program to write 5 names in File and read from file and display on screen.

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

void main()
{
    FILE *fp;
    char name[30];
    int i;
    clrscr();

    fp=fopen("test.txt","a+");
    for(i=0;i<5;i++)
    {
        printf("\n\t Name :-");
        scanf("%s",&name);
        fprintf(fp,"%s",name);
        fprintf(fp,"\n"," ");
    }
    fp=fopen("test.txt","r");

    while(!feof(fp))
    {
        fscanf(fp,"%s",name);
        printf("\n %s",name);
    }
    getch();
}
```

File Handling in C Language

Following program reads a character from user and write in to file one.txt.

```
#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;
```

File Handling in C Language

```
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();
}
```

In this program, we have create two FILE pointers and both are referring to the same file but in different modes.

fprintf() function directly writes into the file, while **fscanf()** reads from the file, which can then be printed on console usinf standard **printf()** function.

```
//Program to copy data from one file to another file
//read data from file one.txt and copy into file one.txt
```

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

void main()
{
    FILE *fp1,*fp2;
    char name[30];
    clrscr();
    fp1=fopen("test.txt","r");
    fp2=fopen("one.txt","w");
    while(!feof(fp1))
    {
        fscanf(fp1,"%s",name);
        fprintf(fp2,"%s",name);
    }
}
```

:- For o/p open file one.txt in TC