TutorialKart

How to Write to File in C?

C – Write to File

When a programs output or some of the variables has to be saved to a storage location on file system, the data has to be written to a file. In this tutorial, we shall learn how to write data to a file using some of the builtin functions of C programming language.

The following functions are used to write data into the file. They are:

- fprintf()
- fputs()
- fputc()
- fwrite()

fprintf()

The fprintf() is used to write formatted output to stream.It is used to write a set of characters into a file.

Syntax

The syntax of fprintf() function is

```
int fprintf(FILE *stream, const char *format);
```

`stream` ? This is the pointer to a FILE object that identifies the stream.

`format` ? This is the C string that contains the text to be written to the stream. It can optionally contain format tags,that are replaced by the values specified in subsequent additional arguments.prototype of format flag is: *% [flags][width][.precision][length]specifier.*

- *Flags* which specifies the output justification such as decimal point,trailing 0's,octal,hexadecimal prefixes.(- ,+,#,0).
- *Width* specifies the minimum number of characters to print after being padded with 0's or blank spaces.
- *Precision* specifies maximum number of characters to print.
- *Length* defines whether the argument ia a short,long or double long.
- *Specifier* is used to define the type and the interpretation of the value of the corresponding argument.

Example

C Program

```c
#include<stdio.h>

int main() {
    FILE *fp;
    char name[50];
    int roll_no,  i, n;
    float marks;

    fp = fopen("marks.txt", "w");

    if(fp == NULL) {
        printf("file can't be opened\n");
        exit(1);
    }

    printf("Enter the number of student details you want to enter: ");
    scanf("%d", &n);

    for(i = 0; i < n; i++) {
        fflush(stdin);
        printf("\nEnter the details of student %d \n\n", i +1);
        printf("Enter name of the student: ");
        gets(name);

        printf("Enter roll no: ");
        scanf("%d", &roll_no);

        printf("Enter marks: ");
        scanf("%f", &marks);

        fprintf(fp, "Name: %s\t Roll no: %d \tMarks: %f \n", name, roll_no, marks);

        printf("\n Details successfully written to the file\n\n");
    }

    fclose(fp);

    return 0;
}
```

Output

```
Enter the number of student details you want to enter: 1
Enter the details of student 1
Enter name of the student: aaa
Enter roll no: 1
Enter marks: 100
Details successfully written to the file
```

fputs()

fputs() is used to write a line to a file.

Syntax

The syntax of fputs() function is

```
int fputs(const char *str,FILE *stream);
```

The fputs() writes the string pointed to by *str* to the stream pointed to by stream.On successful completion ,it returns 0 else EOF.

Example

C Program

```
#include<stdio.h>

int main() {
    FILE *fp;
    char comment[50];

    fp = fopen("comments.txt", "w");

    if(fp == NULL) {
        printf("file couldn't be opened\n");
        exit(1);
    }

    printf(“comment on red color”);

    gets(comment);
    fflush(stdin);
    fputs(comment,fp);
    fclose(fp);
}
```

```
Comment on red color:good.
```

fputc()

fputc() is used to write a character to the stream.

Syntax

The syntax of fputc() function is

```
int fputc(int c,FILE *stream);
```

The `fgetc()` function will write byte specified by `c` to the output stream pointed to by `stream`.On successful completion, `fputc()` will return the value it has written, else `EOF` character will be returned.

Example

C Program

```
#include<stdio.h>

int main() {
    FILE *fp;
    char comment[50];

    fp = fopen("comments.txt", "w");

    if(fp == NULL) {
        printf("file couldn't be opened\n");
        exit(1);
    }

    printf(“provide feedback on a book”);

    gets(comment);

    for(i=0;i<comment[i];i++)
        fputc(comment[i],fp);

    fclose(fp);
}
```

Output

```
Provide feedback on a book:good
```

fwrite()

The fwrite() function is used to write data (can contain multiple characters and multiple lines) to a file.

Syntax

```
int fwrite(const void *str,size_t size,size_t count,FILE *stream);
```

- The *fwrite()* function will write objects of objects specified by *size* from an array pointed to by *ptr* to the stream pointed to by *stream*.
- On successful completion,the fwrite() function returns the number of objects successfully written else EOF will

be set.

Example

In the following example, we will writing a variable to a file.

C Program

```
float *f=200.14;
fwrite(&p,sizeof(f),1,fp);
```

Example

In this example, we will write an array to a file. The below code snippet writes an integer array to a file.

```
int arr[5]={100,320,4,678,220};
    fwrite(arr,sizeof(arr),1,fp);
```

Example

In this example, we will write structure to a file. To write a C structure to a file, use fwrite().

C Program

```
struct student {
    char name[10];
    int roll;
    float marks;
};

    struct student stud= {"rima", 12, 88.123};
    fwrite(&stud, sizeof(stud), 1, fp);
```

Example

In this example, we will write char array to a file.

C Program

```
#include<stdio.h>

int main() {
    FILE *fp;

    size_t count;
```

```
    Char str[] = "good morning";

    fp = fopen("intro.txt", "wb");

    if(fp == NULL) {
        printf("file couldn't be opened\n");
        exit(1);
    }

    count = fwrite(str,1,strlen(str),fp);

    printf(“\n bytes written into file are : ”,count);

    fclose(fp);

    return 0;
}
```

Output

```
bytes were written into file are : 13
```

Conclusion

In this C Tutorial, we have learnt how to write variables or char array or stream of data into a file using different function available in C programming with examples.