**fputc**

**int fputc(int char, FILE \*pointer)**

**char:**  character to be written.

This is passed as its int promotion.

**pointer:** pointer to a FILE object that identifies the

stream where the character is to be written.

**fgetc**

**int fgetc(FILE \*pointer)**

**pointer:** pointer to a FILE object that identifies

the stream on which the operation is to be performed.

fgets()

char \*fgets(char \*str, int n, FILE \*stream)

Parameters

* **str** − This is the pointer to an array of chars where the string read is stored.
* **n** − This is the maximum number of characters to be read (including the final null-character). Usually, the length of the array passed as str is used.
* **stream** − This is the pointer to a FILE object that identifies the stream where characters are read from.

Writing File : fputs() function

The fputs() function writes a line of characters into file. It outputs string to a stream.

**Syntax:**

**int** fputs(**const** **char** \*s, **FILE** \*stream)

**Example:**

#include<stdio.h>

#include<conio.h>

**void** main(){

**FILE** \*fp;

clrscr();

fp=fopen("myfile2.txt","w");

fputs("hello c programming",fp);

fclose(fp);

getch();

}

**The fwrite() function**

The fwrite() function is used to write records (sequence of bytes) to the file. A record may be an array or a structure.

**Syntax of fwrite() function**

fwrite( ptr, int size, int n, FILE \*fp );

The fwrite() function takes four arguments.  
ptr : ptr is the reference of an array or a structure stored in memory.  
size : size is the total number of bytes to be written.  
n : n is number of times a record will be written.  
FILE\* : FILE\* is a file where the records will be written in binary mode.

**Example of fwrite() function**

#include<stdio.h>

struct Student

{

int roll;

char name[25];

float marks;

};

void main()

{

FILE \*fp;

char ch;

struct Student Stu;

fp = fopen("Student.dat","w"); //Statement 1

if(fp == NULL)

{

printf("\nCan't open file or file doesn't exist.");

exit(0);

}

do

{

printf("\nEnter Roll : ");

scanf("%d",&Stu.roll);

printf("Enter Name : ");

scanf("%s",Stu.name);

printf("Enter Marks : ");

scanf("%f",&Stu.marks);

fwrite(&Stu,sizeof(Stu),1,fp);

printf("\nDo you want to add another data (y/n) : ");

ch = getche();

}while(ch=='y' || ch=='Y');

printf("\nData written successfully...");

fclose(fp);

}

Output :

Enter Roll : 1

Enter Name : Ashish

Enter Marks : 78.53

Do you want to add another data (y/n) : y

Enter Roll : 2

Enter Name : Kaushal

Enter Marks : 72.65

Do you want to add another data (y/n) : y

Enter Roll : 3

Enter Name : Vishwas

Enter Marks : 82.65

Do you want to add another data (y/n) : n

Data written successfully...

**The fread() function**

The fread() function is used to read bytes form the file.

**Syntax of fread() function**

fread( ptr, int size, int n, FILE \*fp );

The fread() function takes four arguments.  
ptr : ptr is the reference of an array or a structure where data will be stored after reading.  
size : size is the total number of bytes to be read from file.  
n : n is number of times a record will be read.  
FILE\* : FILE\* is a file where the records will be read.

**Example of fread() function**

#include<stdio.h>

struct Student

{

int roll;

char name[25];

float marks;

};

void main()

{

FILE \*fp;

char ch;

struct Student Stu;

fp = fopen("Student.dat","r"); //Statement 1

if(fp == NULL)

{

printf("\nCan't open file or file doesn't exist.");

exit(0);

}

printf("\n\tRoll\tName\tMarks\n");

while(fread(&Stu,sizeof(Stu),1,fp)>0)

printf("\n\t%d\t%s\t%f",Stu.roll,Stu.name,Stu.marks);

fclose(fp);

}

Output :

Roll Name Marks

1 Ashish 78.53

2 Kaushal 72.65

3 Vishwas 82.65

Random Access To File

The C library function **int fseek(FILE \*stream, long int offset, int whence)** sets the file position of the **stream** to the given **offset**.

Declaration

Following is the declaration for fseek() function.

int fseek(FILE \*stream, long int offset, int whence)

Parameters

* **stream** − This is the pointer to a FILE object that identifies the stream.
* **offset** − This is the number of bytes to offset from whence.
* **whence** − This is the position from where offset is added. It is specified by one of the following constants −

|  |  |
| --- | --- |
| **Sr.No.** | **Constant & Description** |
| 1 | **SEEK\_SET**  Beginning of file |
| 2 | **SEEK\_CUR**  Current position of the file pointer |
| 3 | **SEEK\_END**  End of file |

Example program for fseek():  
**Write a program to read last ‘n’ characters of the file using appropriate file functions(Here we need fseek() and fgetc()).**

|  |  |
| --- | --- |
| 01 | #include<stdio.h> |
| 02 | #include<conio.h> |

|  |  |  |
| --- | --- | --- |
| 03 | void main() | |
| 04 | { |

|  |  |  |
| --- | --- | --- |
| 05 | FILE \*fp; | |
| 06 | char ch; |

|  |  |
| --- | --- |
| 07 | clrscr(); |
| 08 | fp=fopen("file1.c", "r"); | |

|  |  |
| --- | --- |
| 09 | if(fp==NULL) |
| 10 | printf("file cannot be opened"); | |

|  |  |  |
| --- | --- | --- |
| 11 | else | |
| 12 | { |

|  |  |  |
| --- | --- | --- |
| 13 | printf("Enter value of n  to read last ‘n’ characters"); | |
| 14 | scanf("%d",&n); |

|  |  |
| --- | --- |
| 15 | fseek(fp,-n,2); |
| 16 | while((ch=fgetc(fp))!=EOF) | |

|  |  |
| --- | --- |
| 17 | { |
| 18 | printf("%c\t",ch); | |

|  |  |  |
| --- | --- | --- |
| 19 | } | |
| 20 | } |

|  |  |  |
| --- | --- | --- |
| 21 | fclose(fp); | |
| 22 | getch(); |

|  |  |
| --- | --- |
| 23 | } |