

UNIT-6 (FILE HANDLING)

(RAM GOPAL GUPTA- <http://ramgopalgupta.com/>)

PART-3

File Handling:

After discussing “how to open/ create file” and “how to close file”; now we will move over to the other topics:

3: Writing to a File:

We can write content to a file using C program.

: Method 1:

Using fputc(..) function:

The fputc() function is used to write a single character into file. It outputs a character to a stream.

Syntax:

```
int fputc(int c, FILE *fptr)
```

Example1:

```
//example1.c
#include <stdio.h>
main(){
    FILE *fp;
    fp = fopen("test.txt", "w");           //opening file in write mode
    fputc('a',fp);                         //writing single character 'a' into file
    fclose(fp);                             //closing file
}
```

Explanation:

When you will run the above program a file “test.txt” will create at the same location where “example1.c” is saved and when you will open “test.txt” file character ‘a’ will be written there.

test.txt

a

Example2:

```
// example2.c
Line 1- #include<stdio.h>
Line 2- void main(){
Line 3- FILE *fp;
Line 4- char ch;
Line 5- fp=fopen("a2.txt","w");
Line 6- ch=getchar();
Line 7- while(ch!='\n'){
Line 8- putc(ch,fp);
Line 9- ch=getchar();
Line 10- }
Line 11- fclose(fp);
Line 12- }
```

Explanation:

Line 3- declares a file pointer “fp”

Line 4- declares a character variable “ch”

Line 5- creates a new file “a2.txt” at the same location where your program “example2.c” is saved and open “a2.txt” file in write mode.

Line 6- using getchar() function, we are accepting a character from the user and storing it into character variable “ch”.

Line 7- While loop is checking where the value of “ch” is not ‘\n’ (enter). If “ch” has other than enter key value (‘\n’), program flow will move to the body of the loop.

Line 8- putchar(ch,fp) function will write the value of variable “ch” into the file “a2.txt” which is referred by pointer “fp”.

Line 9- Again using getchar() function, we are accepting a character from the user and storing it into character variable “ch”.

Line 10- Line 8 to 9 will execute repeatedly till the condition written at line 8 will be true. Once the user press “enter key” while loop will terminate.

Line 11- fclose(fp) close function will close the file which is referenced by pointer “fp” from the memory.

Important:

When you will open the file “a2.txt” from the same location where you have saved “example2.c” file, all the content which are written at the program execution will be shown here.

: Method 2:

Using fprintf(..) function:

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

Syntax:

```
int fprintf(FILE *stream, const char *format [, argument, ...])
```

Example3:

```
//example3.c
```

```
Line 1- #include <stdio.h>
```

```
Line 2- int main(){
```

```
Line 3- FILE *fp;
```

```
Line 4- fp = fopen("data.txt", "w"); //opening file
```

```
Line 5- fprintf(fp, "Hello I am using fprintf...\n"); //writing data into file
```

```
Line 6- fclose(fp); //closing file
```

```
Line 7- return 0;
```

```
Line 8- }
```

Explanation:

When you will run the above program a file “data.txt” will create at the same location where “example2.c” is saved and when you will open “data.txt” file character it will display as shown below:

data.txt

```
Hello I am using fprintf...
```

```

// example4.c
Line 1- #include <stdio.h>
Line 2- void main()
Line 3- {
Line 4- FILE *fptr;
Line 5- int id;
Line 6- char name[30];
Line 7- float salary;
Line 8- fptr = fopen("emp.txt", "w");/* open for writing */
Line 9- if (fptr == NULL)
Line 10- {
Line 11- printf("File does not exists \n");
Line 12- return;
Line 13- }
Line 14- printf("Enter the id\n");
Line 15- scanf("%d", &id);
Line 16- fprintf(fptr, "Id= %d\n", id);
Line 17- printf("Enter the name \n");
Line 18- scanf("%s", name);
Line 19- fprintf(fptr, "Name= %s\n", name);
Line 20- printf("Enter the salary\n");
Line 21- scanf("%f", &salary);
Line 22- fprintf(fptr, "Salary= %.2f\n", salary);
Line 23- fclose(fptr);
Line 24- }

```

Output:

```

Enter the id
1
Enter the name
RGG
Enter the salary
200

```

Now open the file “emp.txt” from the same location where you have saved “example4.c” file
emp.txt

```

id = 1
Name = RGG
Salary= 200

```

Important

In the file part-1 of unit-6; I have mention two functions for writing the data into the file fputc(..) & fprintf(.....) here I am using one more function for writing the data into the file and the name of the function is "fwrite(...)"

: Method 3:

Using 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:

```
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.

Example5:

//example5.c

```
Line 1- #include<stdio.h>
Line 2- struct Student
Line 3- {
Line 4- int roll;
Line 5- char name[25];
Line 6- float marks;
Line 7- };
Line 8- void main()
Line 9- {
Line 10- FILE *fp;
Line 11- char ch;
Line 12- struct Student Stu;
Line 13- fp = fopen("Student.dat","w");
Line 14- do
Line 15- {
Line 16- printf("\nEnter Roll : ");
Line 17- scanf("%d",&Stu.roll);
Line 18- printf("Enter Name : ");
Line 19- scanf("%s",Stu.name);
Line 20- printf("Enter Marks : ");
Line 21- scanf("%f",&Stu.marks);
Line 22- fwrite(&Stu,sizeof(Stu),1,fp);
Line 23- printf("\nDo you want to add another data (y/n) : ");
Line 24- ch = getche();
Line 25- }while(ch=='y' || ch=='Y');
Line 26- printf("\nData written successfully...");
Line 27- fclose(fp);
Line 28- }
```

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...
```

Explanation:

- From line 2 to 7: a structure “Student” has been defined.
- At line 12 structure variable “stu” has been declared.
- At line 13 a “Student.dat” file has been opened in write mode and referenced has been moved to file pointer “fp”.
- From line 14 to 25: do.. while(..) loop is executed to get the record of the student till the user want to add data.
- At line 22 fwrite(...) function is used to write the record into file “Student.dat”.

So in this file we had discussed only about how to write content into the file using different functions.

I would like to mention here that if you write data using fputc(..) and fprintf(..), it will work as character stream. It means when you will open the “txt” file directly from your system, you can see exact data into the file as you have written through your C program.

But in case of fwrite(..) function data is written in byte stream format therefore the file in which data is written through this C program, if open directly from the system the written data will not be in readable format. To read the content from such files you have write another C program.

IN NEXT FILE (PART-4) WILL DISCUSS; HOW TO READ CONTENT FROM THE FILE THROUGH C PROGRAM →