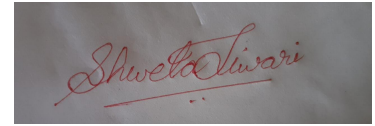


By Shweta Tiwari from IT Department

Covered Topics Under UNIT-5 of "PPS-PROGRAMMING FOR PROBLEM SOLVING (BCS101 / BCS201)"

shwetatiwario8@recabn.ac.in
shwetatiwario8aug@gmail.com



Class Notes

Published Date: November, 2022

PPS: UNIT-5

Pointers

File Handling

FALL SEMESTER, YEAR (I/II sem, 1st yr)

*FALL SESSION (2022-23)
(PPS)*

MS. SHWETA TIWARI

Published Date: November, 2022

shwetatiwario8@recabn.ac.in
shwetatiwario8aug@gmail.com

By SHWETA TIWARI
Under On: UNIT-5

PREPARED FOR
Engineering Students
All Engineering College

PREPARED BY
SHWETA TIWARI

Units Classified into Chapters

PPS: UNIT-5:

Chapter-14 FILE HANDLING IN C

Chapter-15 MACRO

Chapter 14

FILE HANDLING IN C

To understand the concept of File handling, we must understand the following things;

1. Data : Raw information is called data. For example student name, father's name, address, dob, rollno etc. Each individual item is called data.

For example

Name is a data, rollno is a data, dob is a data. In electricity dept person name, address, meter no is a data.

2. Record: Record is a collection of data of same type. It is a detail of data. For example all things in your admission form name, fname, dob, rollno, marks of yours together is known as record.

For example (navjeet, s. karamjit singh, 2-08- 1983, 1. 89). It is a record of a single student.

In one record data should be of same concerned item or thing or person. In one record we cannot mix data of different things.

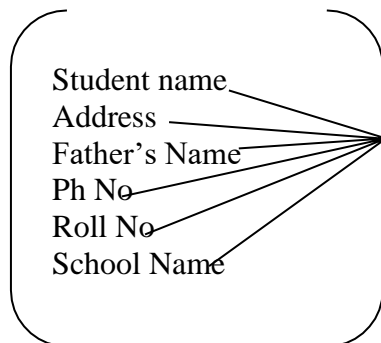
Eg; name of ram

Roll no of sham

This is wrong. Here Roll No and Name must be of same person.

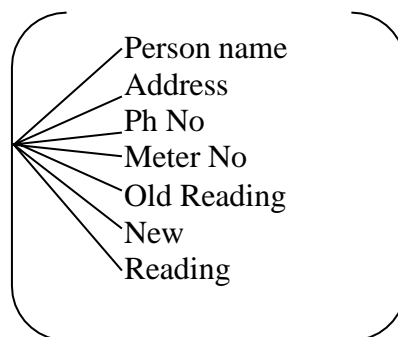
Now you see two different records

Student's Record



D
A
T
A

Record in Electricity dept



Suppose these two different records are of yours. Name, Ph.no, Address is common. But Roll no, meter no is different that's why we create different records.

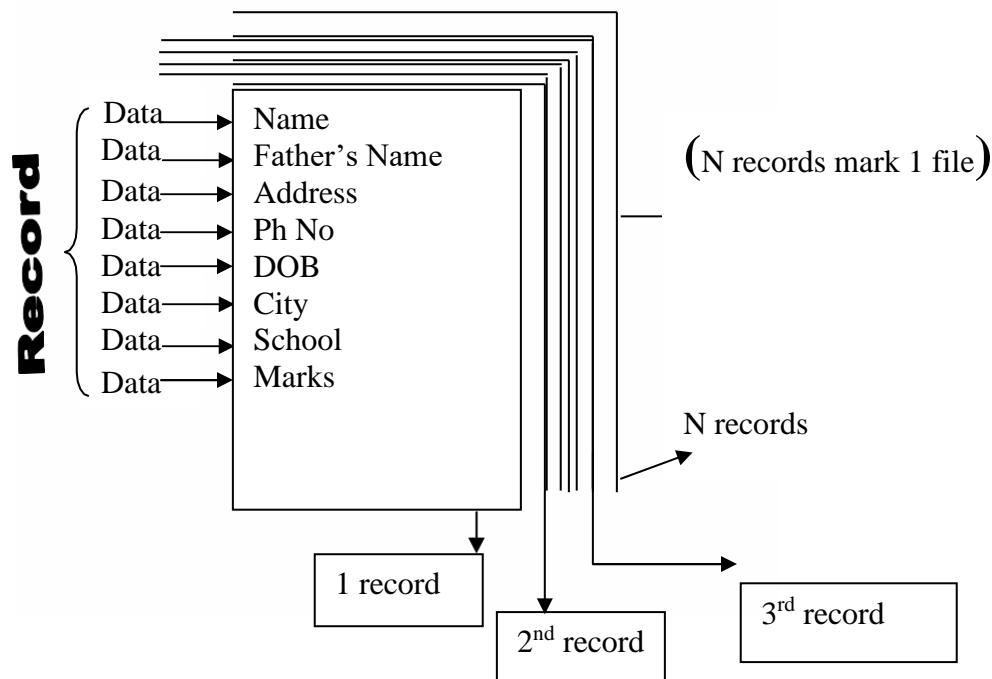
3. File : File is a collection of similar same type of records. For example student's records will be placed in a one file. (Student name, Address, father's name, phno, rollno, school name)

Electricity dept will hold records of bill in different file(Person Name, Address , Ph No, Meter No, Old Reading, New Reading)

Note

We cannot mix different records in a one file. To maintain different records we need to maintain different records. That's why we need to maintain different files.

Diagram of file



In earlier days, the amount of information is stored in manual form. The major drawbacks of manual system are Required more space to store the data in file Updating of data was very difficult.

Searching is quite difficult.

No security in manual Files Manual file were not reliable means if file was lost all the data would also lost.

Now modern files are made on computers. The data is stored in computerized files.

It has following advantages;

1. Data can be easily stored in compact space updating of data is very easy. Using the pointers concept when we perform any changes in a file, all the files which related with that file are automatically updated.
2. Searching is very easy.
3. The data in modern file are very secure. Only the authorized persons can view data.
4. Files are very reliable because adequate backup checks are there.

The main motive of file handling is to transfer the data from primary memory to secondary memory.

Types of Files

There are three types of files, which are;

1. **Text Files**
2. **Data Files**
3. **Binary Files**

1. **Text Files:** - Text files are those files, which store the information in the text format. It stores the data into character or strings. The extension of text file is .txt.
2. **Data Files:** - Data files are known as database files, which store the data in different formats. The extension of data file is .dat.
3. **Binary Files:** - These files store data in binary form. These files store data in sequence of bytes. It is an Encrypted file so we cannot read it.

FILE Structure

FILE is our inbuilt structure, which is used to create a file. We can create a pointer of the FILE structure. It is known as file declaration. As shown in the example and syntax.

Syntax:-

```
FILE filepointer;
```

Example:-

```
FILE *fp;
```

Note: The type name FILE is always written in capital letters

File handling Functions

1. **fopen() function:-** Before using data file it is necessary to open that file. It can be opened by using fopen() function. The function has two arguments, first is name of the file to be opened , and second is string representing mode in which the file is to be opened. As shown in the syntax and example given below:

Syntax:-

```
fp=fopen("filename","mode");
```

e.g.

```
fp=fopen("CJSOFTECH.txt","w");
```

File name: - It specifies the file name where we want to store the data. Like "CJSOFTECH.txt", here CJSOFTECH is the file name and .txt its extension. It means it is a text file.

Modes: -File can be opened in following modes:

1. **Read:** - In this mode we can read the data from the file. When a file is opened in this mode, it is not possible to write the data into the file. We can use "r" as a parameter in fopen function.
2. **Write:** - If a file in this mode is opened we can write the data into the file but it is not possible to read the data from file. A file already exists is opened in the write mode, the old contents of the file will be deleted and new data can be written in the file. We can use "w" as a parameter in fopen function.

3. Append: - If a file in this mode is opened we can add the data into the file at the end of the file. Parameter used in fopen is "a".

r+ mode:- To open the file for reading and writing the data.

w+ mode:- To open the file for writing. Data can be read after writing.

a+ mode:- To open the file ,write the data at the end of the file. Data can be read after writing.

2. fclose() function:- It is used to close a file. This function contains a single argument, which is the file pointer. As shown in the syntax and example given below:

Syntax: -

fclose(file pointer);

Example:

fclose(fp);

3. fputs() function:- It is used to write a string into the file. This function contains two arguments first is string and second is the file pointer. As shown in the syntax and example

Syntax : -

fputs(string, file pointer);

Example:-

fputs(str, fp);

4. fgets() function:- This function is used to read the data from a file in string format. This function contains three arguments. First is string in which data is stored, second argument is the length of the string and third is the file pointer (fp). As shown in the syntax and example

Syntax: -

fgets(string, length, file pointer);

Example: -

fgets(str,40,fp);

5. getc() Function:- This function is used to read a character from a file. This function contains a single argument of file pointer. As shown in the syntax and example-

Syntax: -

variable=getc(file pointer);

Example:

char a;

a=getc(fp);

Note:

The character read from the file is assigned to a char variable a.

6. putc() Function:- This function is used to write a character to a file. It has two arguments, first is the character variable and second is the file pointer. As shown in the example:-

Example:char a;

putc(a,fp);

Note:

The value of a char variable 'a' is written in the file.

7. getw () Function:- This function is used to read an integer from a file. This function also contains a single argument, which is the file pointer. As shown in the example: -

```
a=getw(fp);
```

8. putw() Function:- This function is used to write an integer to a file. This function contains two arguments, first is the integer variable and second is the file pointer. As shown in the example: -

```
putw(a,fp);
```

9. feof function :- feof means file end of file. This function reads the file till end. This function returns a negative number when we reach at the end of the file. As shown in the example:-

```
while(feof(fp)!=-1)
```

Write a program to store the data in the file using fputs() function.

```
include<stdio.h>
include<conio.h>
void main()
{
char str[32];
FILE *fp;
clrscr();
printf("enter any string");
gets (str);
fp=fopen("CJSOFTECH.txt","w");
puts(str,fp);
printf("data is written into the file");
fclose(fp);
getch(); }
```

WAP to read the data from a file using fgets() function.

```
include<stdio.h>
include<conio.h>
void main()
{
char str[32],*p;
FILE *fp;
clrscr();
fp=fopen("CJSOFTECH.txt","r");
while((p=fgets(str,40,fp))!='\0')
{
printf("%s",str);
}
fclose(fp);
getch(); }
```

WAP to copy the contents of one file into another file.

```
include<stdio.h>
```

```

include<conio.h>
void main()
{
char str[32],*p;
FILE *fp,*fp1;
clrscr();
fp=fopen("CJSOFTECH.txt","r");
fp1=fopen("CJSOFTECH.txt","w");
while((p=fgets(str,40,fp))!='\0')
{
fputs(str,fp1);
}
fclose(fp);
printf("data is copied");
fclose(fp1);
getch();
}

```

Write a program to write Integer data in a Binary file.

```

void main()
{
FILE *fp;
int i;
clrscr();
fp=fopen("CJSOFTECH.dat","w");
for(i=1;i<=10;i++)
{
putw(i,fp);
}
printf("data is written");
fclose(fp);
getch();
}

```

Write a program to read integer data from a Binary file.

```

#include<stdio.h>
#include<conio.h>
void main()
{
FILE *fp;
int n;
clrscr();
fp=fopen("CJSOFTECH.dat","r");
while((n=getw(fp))!=EOF)
{

```



```
printf("%d",n);  
}  
fclose(fp);  
getch();  
}
```

Random Access Files

Random access file are those files in which we can read and write the data at random location. But in some situation, when we need to access the data from a specific part of a file., then we use `fseek()`, `ftell()`, and `rewind()` functions.

1. `fseek()` Function:- `fseek` function is used to move the file pointer to a desired location within the file. This function contains three arguments . The first argument is the file pointer, second argument tells how many bytes move in the file and third argument tells the current position of the pointer. As shown in the syntax: -

`fseek(file pointer, n, position);`

2. `ftell()` Function:- This function tells us the current position of the pointer. This function contains a single argument of file pointer. As shown in the given syntax:-

`n=ftell(file pointer);`

3. `rewind()` Function:- This function is used to set the pointer or cursor in the beginning of the file. This function contains a single argument of file pointer. As shown in the syntax:

`rewind(fp);`

Error Handling during input/output operations

Error is occurring during input/output operation of the file when an error is occurred in the program the program is terminated. There are so many reasons for an error like file not found, null pointer etc.

Error Handling

Error handling means to solve the problem or to remove the errors with the help of inbuilt function, which are provided us by the programming language.

➤ **NULL:** - This help us to check if the file is exist or not. If file does not exist then it equal to null. As shown in the example:-

```
FILE *fp;  
Fp=fopen("CJSOFTECH.txt","r");  
If(fp==NULL)  
{  
printf("File doesn't exist");  
exit(0);  
}  
else  
{  
Read the file contents" }
```

- **ferror():**- This is known as file error function. This function is used when we access the file, if there is any problem in accessing the file then this function returns a positive or negative number. But when there is no problem in accessing the file then it returns zero. As shown in the example: -

```
FILE *fp;
fp=fopen("CJSOFTECH.txt","w");
if (ferror(fp)==0)
{
printf("No problem in file accessing");
}
else
{
printf("There is problem in file accessing");
}
```

- **EOF():**- This is known as End of file. This function is used to check the file end. In place of EOF we can use -1. It is same as end of file; when our cursor reach as the end of the file then it returns a negative number -1 or EOF. As shown in the example:-

```
FILE *fp;
int a;
fp=fopen("CJSOFTECH.txt","r");
While(a=getw(fp)!=EOF)
{
printf("\n %d",a);
}
```

Important Short Answer Type Questions

1. List down two file opening modes in C?
2. What are random access files?
3. What is a FILE?
4. What are the advantages of data file over other data types?
5. Difference between fseek() and ftell() function.

Important Long Answer Type Questions

1. Explain various File Handling Functions available in C.
2. Write a C program to read a text from the file and display it on the screen.
3. Write a C program to copy the contents of a file into another file and also update that file.
4. Discuss in brief getw(), putw() functions.
5. Distinguish between append mode and write mode.
6. State the difference between Sequential Files and Random Access Files.
7. Explain character input/output handling functions.
8. Write a C program to create a file containing integers and display them.
9. What is the advantage of using a data file?

Chapter 15

MACRO

Macro is also called a substitution string. A macro always start with # define. After the # define, we can give the name to the macro and after that we can give replacement text/value. Wherever the preprocessor encounters that name, it will replace it with the replacement text. Macro is always declared in capital letters, which differentiate the macro from variable. If we want to change the value of the macro variable we can do it at only at place where it is defined. Macro never ends with semicolon.

The value of the macro body specified by a #define directive can be any character string or number. For example, the following definition is NAME with the string "CJSOFTECH".

SYNTAX:

#define <identifier><value>

Example:-

```
#define A 50
```

WAP to use macro in the program.

```
#include<stdio.h>
#include<conio.h>
#define a 10
void main()
{
int b;
clrscr();
b=a*a;
printf("square=%d",b);
getch();
}
```

In macro we can also pass arguments inside the macro. It works as function, which are defined before using it.

Write a program to swap two numbers without using third variable and by using macro.

```
#include<stdio.h>
#include<conio.h>
#define swap(x,y)x=x+y;y=x-y;x=x-y
void main()
{
int a,b;
clrscr();
printf("enter two no");
scanf("%d%d", &a,&b);
swap(a,b);
printf("\n now value of a=%d",a);
printf("\n now value of b=%d",b);
}
```

```
getch();  
}
```

Typedef

Definition: - Typedef means to rename the existing data type. With the help of 'typedef' we can give the new name to the inbuilt data type /structure typedef declaration does not declare any new data type which does not exist in the C language. Typedef means to create new definition name of an existing data type.

SYNTAX: -

typedef<data type><new data type>;

Example: -

typedef int cj ;

Example:-

```
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
typedef int cj;  
clrscr();  
cj a,b,c;  
printf("enter two numbers")  
scanf("%d%d",&a,&b);  
c=a+b;  
printf("sum=%d",c);  
getch();  
}
```

Command Line Arguments

Command line arguments is the another way to input the values into our program using console instead of using scanf() function. In C ,mostly we use scanf() to input the values into the program. Command line arguments are the values passed into the main function from command line (Dos prompt). We can pass two arguments to the main() function:

- 1. argc** : integer type argument
Indicates the number of parameters passed.
- 2. argv** : array of strings
Each string in this array will represent a parameter that is passed to main().

As shown in the program below:

```
#include<stdio.h>
#include<conio.h>
void main(int argc,char *argv[])
{
int i;
clrscr();
for(i=1;i<argc;i++)
{
printf("%s",argv[i]);
}
getch();
}
```

Steps to execute the program

1. After typing the program save the file with .c extension.
2. Now compile the program using ALT+F9.
3. Now click on file menu and select "**Dos shell**" option.
4. Now first type file name and then type number of arguments.

Example: -

C:\tc\bin> filename one two three

Note: -Here CJSOFTECH is the file name and one ,two three are arguments

5. Now press enter key.

To go back to the C program type "exit" and press enter key

Imp Short Answer Type Questions

1. Define Macro.
2. Write a note on typedef.
3. What are command line arguments?
4. Why do we use #define?
5. WAP to create a macro to find the square of a number.