File

Structured Programming Language (CSE-1271)

Course Instructor : Mohammed Mamun Hossain
Assistant Professor, Dept. of CSE, BAUST

Outline

1. Introduction
2. Why?
3. Operation
4. Open & Close
5. Mode
6. Access Data (Read & Write)

File

- A file represents a sequence of bytes on the disk where a group of related data is stored. File is created for permanent storage of data. It is a ready made structure.

- In C language, we use a structure pointer of file type to declare a file.

```
FILE *fp;
```

File | Why?

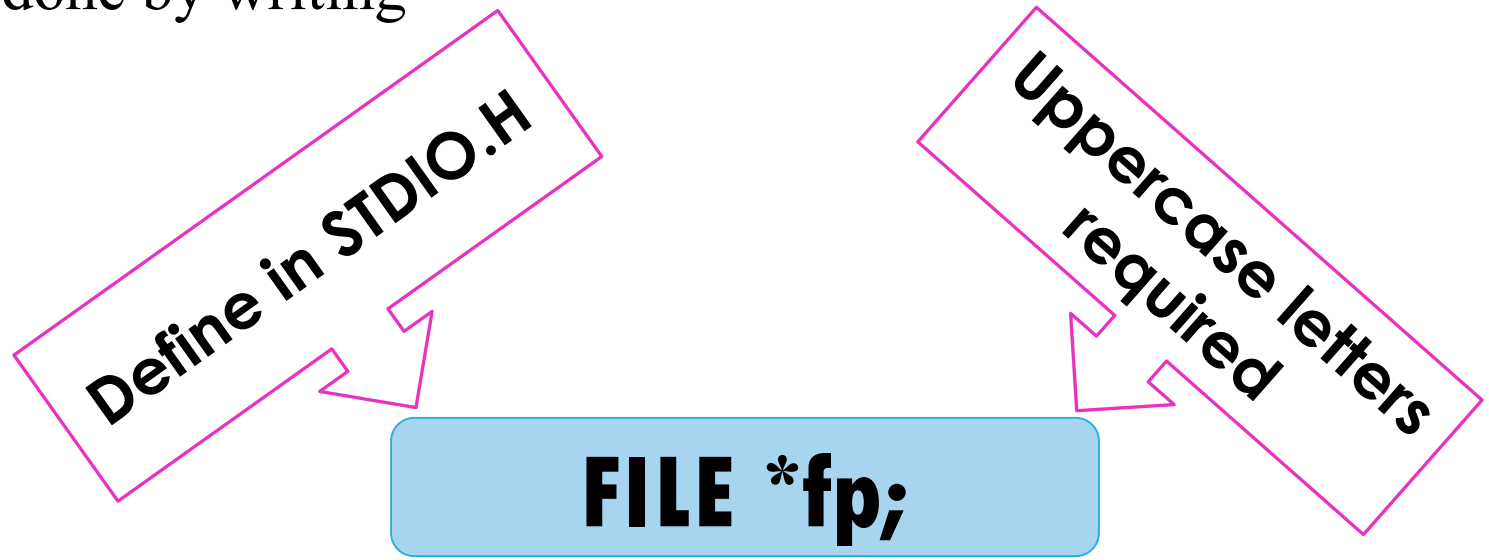
- When the program is terminated, the entire data is lost in C programming.
- If you want to keep large volume of data, it is time consuming to enter the entire data.
- But, if file is created, these information can be accessed using few commands.

File | Operation

- There are large numbers of functions to handle file I/O in C language.
- High level file I/O functions can be categorized as:
  - ✓ Text file
  - ✓ Binary file
- File Operations:
  - Creating a new file
  - Opening an existing file
  - Reading from and writing information to a file
  - Closing a file

File | Open & Close

- To work with stream oriented data file first we need to establish a buffer area.
- Where information is temporarily stored while being transferred between memory and the data file.
- It's done by writing-



File | Open & Close

- Opening a file is performed using library function fopen(). The syntax for opening a file in standard I/O is:

```
fopen (file_name, mode);
```

- If the open operation is successful, fopen() returns a valid file pointer.
- The buffer area hold this.

File | Open & Close

- The FILE is defined in STDIO.H
- It is a structure that holds various kinds of information about the file.
  - ✓ its size
  - ✓ the current location of file
  - ✓ its associated modes
- Closing a File:
  - The file should be closed after reading/writing of a file. Closing a file is performed using library function fclose().

```
fclose(ptr);
```

  - ptr is the file pointer associated with file to be closed.

File | Open Mode

Mode	Meaning
"r"	Open an existing file for reading only
"w"	Open a new file for writing only.
"a"	Open an existing file for appending (for adding new information at the end)
"r+"	Open a text file for both reading and writing
"w+"	Create a text file for both reading and writing
"a+"	Appending read/write

- If you are going to handle binary files, then you will use following access modes instead of the above mentioned ones −

"rb","wb","ab","rb+","r+b","wb+","w+b","ab+", "a+b"

File | Access Data

- Writing to a file:

FileWritee.c

```c
#include <stdio.h>
int main()
{
    int n;
    FILE *ptr;
    //ptr=fopen("D:\\mydata.txt","w");
    ptr=fopen("mydata.txt","w");
    if(ptr==NULL)
    {
        printf("Error!");
        exit(1);
    }
    printf("Enter n: ");
    scanf("%d",&n);
    fprintf(ptr,"%d",n);
    fclose(ptr);
    return 0;
}
```

This program takes the number from user and stores in file.

After run, a text file mydata.txt created in current directory of your computer.

When you open that file, you can see the integer you entered.

File | Access Data

- Reading from file:

leReadd.c

```c
#include <stdio.h>
int main()
{
    int a;
    FILE *ptr;
    //ptr=fopen("D:\\mydata.txt","r");
    ptr=fopen("mydata.txt","r");
    if (ptr==NULL)
    {
        printf("Error! opening file");
        exit(1);
        /* Program exits if file pointer returns NULL. */
    }
    fscanf(ptr,"%d",&a);
    printf("Value of a=%d\n",a);
    fclose(ptr);
    return 0;
}
```

This program read the number from a file mydata.txt in current directory.

After run, a number is printed on screen.

File | Functions

- 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
ftell()	gives current position in the file
rewind()	set the position to the beginning point

File | Example

```c
#include<stdio.h>

int main()
{
    FILE *fp;
    char a[20], c;
    int noL=0,noT=0,noB=0,noC=0;
    printf("Enter the name of File: ");
    gets(a);
    if((fp=fopen(a,"r"))==NULL)
    {
        printf("File dosen't exist.");
    }
    else
    {
        while(1)
        {
            c=fgetc(fp);
            if(c==EOF)
                break;
            noC++;
            if(c==' ')
                noB++;
            if(c=='\n')
                noL++;
            if(c=='\t')
                noT++;
            printf("%c",c);
        }
    }
    fclose(fp);
    printf("\nNumber of characters = %d",noC);
    printf("\nNumber of blanks = %d",noB);
    printf("\nNumber of tabs = %d",noT);
    printf("\nNumber of lines = %d\n",noL);

    return 0;
}
```

- The name of the file is input by the user.
- Remember that you have to write the full name of the file including extension.
- Make sure that the file is present with some text in the current directory of your program.

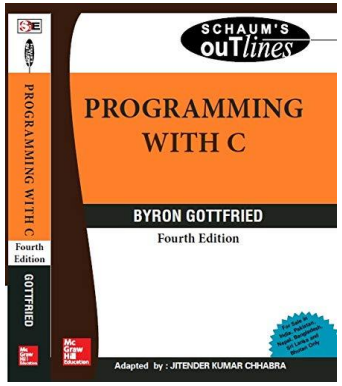
Thank You.

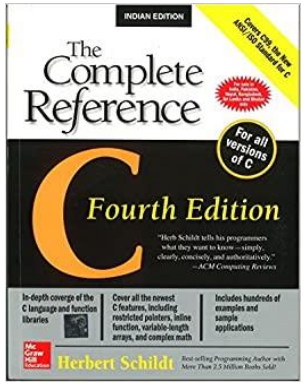
Questions and Answer

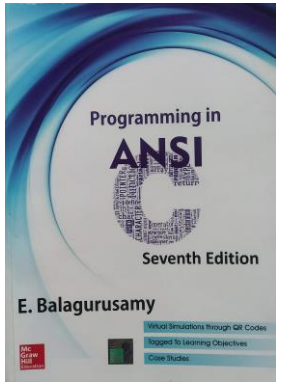
References

Books:

1. Programming With C. *By Byron Gottfried*
2. The Complete Reference C. *By Herbert Shield*
3. Programming in ANSI C *By E. Balagurusamy*
4. Teach yourself C. *By Herbert Shield*







Web:

1. www.wikbooks.org

and other slide, books and web search.