UNIT-6 (FILE HANDLING)
(RAM GOPAL GUPTA- http://ramgopalgupta.com/)

PART-2

File Handling:

As I had discussed in previous file (Part-1), the operations of which are performed in a file are:

- Creation of a file
- Opening a file
- Reading a file
- Writing to a file
- Closing a file

In this part-2, I am discussing three operations: creating/ opening a file and closing a file.

File handling operations:

1- How to Create/ Open File: -

- The first step is to create/ open a file for writing or reading purpose
  Syntax which is used:

```
FILE *fp;
fp = fopen ("file_name", "mode");
```

Explanation:

- In the above syntax, the **FILE** is a data structure which is defined in the standard library.
- **fopen(…)** is a standard function which is used to open a file and returns a file pointer.
- If the **file_name** is not present on the system, then it is created and then opened.
- If a **file_name** is already present on the system, then it is directly opened using this function.
- **fp** is a file pointer which points to the type **FILE**.
- Whenever you open or create a file, you have to specify what you are going to do with the file. A file in 'C' programming can be created or opened for reading/writing purposes.
- A **mode** is used to specify whether you want to open a file for any of the below-given purposes. Following are the different types of modes in 'C' programming which can be used while working with a file.

Mode	Description
r	Opens a text file in read mode. If a file is in reading mode, then no data is deleted if a file is already present on a system.
w	Opens a text file in write mode. If a file is in writing mode, then a new file is created if a file doesn't exist at all. If a file is already present on a system, then all the data inside the file is truncated, and it is opened for writing purposes.

a	Opens a text file in append mode. If a file is in append mode, then the file is opened. The content within the file doesn't change.
r+	opens a text file in read and write mode
w+	opens a text file in read and write mode
a+	opens a text file in read and write mode

Code Example1:

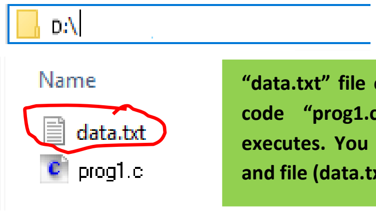
```
// prog1.c
    Line 1-     #include <stdio.h>
    Line 2-     int main() {
    Line 3-     FILE *fp;
    Line 4-     fp  = fopen ("data.txt", "w");
    Line 5-     }
```

In the given syntax, the filename (“**data.txt”**) and the mode (“**w”**) are specified as strings hence they must always be enclosed within double quotes.

Explanation:

Line4: File is created in the same folder where you have saved your code.

Suppose you have saved the above code in a file “**prog1.c**” in **“D” drive**, when you will run this program complier will create a file **“data.txt”** in same location where your program is saved i.e. **“D” drive**



“data.txt” file created in same location, where your code “prog1.c” is saved, when your program executes. You can see here your program (prog1.c) and file (data.txt) are in same location “D:\”

2- How to Close File: -

One should always close a file whenever the operations on file are over. It means the contents and links to the file are terminated. This prevents accidental damage to the file.

'C' provides the fclose() function to perform file closing operation. The syntax of fclose(….) is as follows,

```
fclose (file_pointer);
```

Code Example2:

```
// prog2.c
    Line 1-     #include <stdio.h>
    Line 2-     int main() {
    Line 3-     FILE *fp;
    Line 4-     fp  = fopen ("data.txt", "r");
    Line 5-     fclose(fp);
    Line 6-     }
```

Explanation:

Line5: The fclose() function takes a file pointer as an argument. The file associated with the file pointer is then closed with the help of fclose() function.
It returns 0 if close was successful and EOF (end of file) if there is an error has occurred while file closing.
After closing the file, the same file pointer can also be used with other files.
In 'C' programming, files are automatically close when the program is terminated. Closing a file manually by writing fclose() function is a good programming practice.

In part-3, we will discuss other operations →