

SURAKU ACADEMY

'C'

File Handling

FILE

A collection of data or information that has a name is called file. Almost all information stored in a computer must be in a file.

Files allow us to store information permanently on to the disk and then access them when needed.

TYPES OF FILE

TEXT FILE:

They consist of sequence of characters. These characters can be interpreted as individual data item.

BINARY FILES:

They organize data into block of contiguous bytes of information. These blocks represent complex data structure such as arrays and structures.

FILE ACCESS TYPES

SEQUENTIAL FILE:

In this type of file, data are kept sequentially. If we want to read the last record of the file it is expected to read all the records before it.

RANDOM ACCESS FILE:

In this type, data can be read and modified randomly.

WORKING WITH FILE IN 'C'

To work with file in **C** language, it provides new data type called **FILE**.

STEPS FOR PROCESSING FILE

DECLARATION OF FILE TYPE POINTER:

C communicates with files using a new data type called a **FILE**.

Syntax:

```
FILE *Variable_Name;
```

Example:

```
FILE *fp;
```


STEPS FOR PROCESSING FILE

OPEN THE FILE:

A file has to be opened before read and write operations. For opening the file we use the predefined function **fopen ()**.

Syntax:

```
fp=fopen("File_Name", "Opening_Mode");
```

Example:

```
FILE *fp;
```

```
fp=fopen("store.txt", "r");
```

If the file does not exist this function returns a **NULL**.

STEPS FOR PROCESSING FILE

OPERATION WITH FILE:

Basically, three types of operation perform with file:

1. Reading from file.
2. Writing to file.
3. Append content to file.

STEPS FOR PROCESSING FILE

CLOSING A FILE:

The file is opened by the **fopen()** should be closed after the work is over.

Syntax:

```
fclose(file_pointer);  
fcloseall();
```

Example:

```
fclose(fp);
```

FILE OPENING MODE

MODE	MEANING	DESCRIPTION
r	Read	Only reading possible. Not create file if not exist.
w	Write	Only writing possible. Create file if not exist otherwise erase the old content of file and open as a blank file.
a	Append	Only writing possible. Create file if not exist; otherwise open file and write from the end of file.
w+	Writing+Reading	Writing and reading possible. Create file if not exist. Erase old content.
r+	Reading+Writing	Reading and writing possible. Do not create file if file not exist. Overwriting existing data.
a+	Appending+Reading	Reading and writing possible. Create file if not exist. Append content at the end of file.

FILE OPENING MODE

BINARY MODES ARE:

1. **wb.**
2. **rb.**
3. **ab.**
4. **r+b.**
5. **w+b.**
6. **a+b.**

FILE ACCESS FUNCTION

TEXT FILE:

- `fprintf();`
- `fscanf();`
- `fgetc();`
- `fputc();`
- `fgets();`
- `fputs();`

BINARY FILE:

- `fprintf();`
- `fscanf();`
- `fread();`
- `fwrite();`

FILE ACCESS FUNCTION

RANDOM FILE ACCESS FUNCTION:

1. **ftell();**
2. **rewind();**
3. **fseek();**

SURAKU ACADEMY

*Thank
you*



SURAKU ACADEMY