

Semantics Of File Handling In C Programming

- **What is FILE?**

A FILE represents a sequence of bytes on the disk where a group of related data is stored.

- File resides in the Hard disk and variables, functions and operators perform actions in RAM. So, Buffer State take place in RAM which communicate b/w file and operations.
- What are the operation we perform in file handling?

Various Operations are done like:

- Creating a File
- Opening a File
- Reading a File
- Writing in a File
- Append in a File
- Moving to specific location in a File
- Closing a File.

Various File Handling Function

Function in Use	Description Of Function
fopen()	Used to open an existing file or a new file
fprintf()	Writing data into an available file
fscanf()	Reading the data available in a file
fputc()	Writing any character into the program file
fputs()	Writing any string into the program file
fgetc()	Reading the character from an available file
fclose()	Used to close the program file
fseek()	Used to set the file pointer to the intended file position
fputw()	Writing an integer into an available file
fgetw()	Used to read an integer from the given file
ftell()	Used to get the total size of file after moving the file pointer at the end of the file
rewind()	Sets the file pointer at the beginning of the stream

- **fopen() Function:-**

- **stdio.h** contains a function called **fopen()**, used to open an existing file or create a new file.

- When working with files, we have to declare a pointer an data type FILE.
- This declaration helps us to work with files through C program because it create buffer and allow us to perform operation in file.
- The syntax for opening a file and mode in standard I/O is:

```
FILE *ptr;
ptr= fopen("file1.txt", "mode");
```

Note: FILE is structure or datatype that is defined inside the stdio library. FILE contains metadata of file like status of file, permission of file, pointer location in file etc.

- **fclose() Function:-**

- The C library function fclose(ptr) closes the file. If we don't use fclose() it may lead to loss of data.
- All buffers will be closed after that.

```
fclose(ptr)
```
- This method returns zero if the stream is successfully closed. On failure, EOF is returned.

- **Types of mode:-**

- "r"(read):- Opens the file for reading only. If the file opened successfully fopen() loads it into memory and sets up pointer which points to the character in it. If the file cannot be opened fopen() return NULL.
- "w"(write):- If the file exists already, its contents are overwritten. If the file doesn't exist, a new file is created. It creates a new file for writing only (no reading).
- "a"(append):- If the file is opened successfully fopen() loads it into memory and sets up a pointer that points to the last character in it. If the file doesn't exist, a new file is created. The file is opened only for appending (writing at end of file).
- "r+"(read and write):- Opens the file for both reading and writing. If opened successfully, fopen() loads it into memory and sets up a pointer which points to the first character in it. Returns NULL, if unable to open the file.
- "w+"(read and write):- If the file exists, its contents are overwritten. If the file doesn't exist, a new file is created. The difference between w and w+ is that we can also read the file also in w+. The difference between in r+ and w+ is that in w+ if file doesn't exist it will create the file. It will not through NULL like r+.
- "a+"(reading and appending(writing at the end)):- If the file is opened successfully fopen() loads it into memory and sets up a pointer which points to the last character in it. If the file doesn't exist, a new file is

created. The file is opened for reading and appending (writing at end of file).