**Handling File Streams in C**

A file can be treated as external storage. It consists of a sequence of bytes residing on the disk. Groups of related data can be stored in a single file. A program can create, read, and write to a file. Unlike an array, the data in the file is retained even after the program finishes its execution. It's a permanent storage medium.

**Declaring a File Pointer**

In C language, in order to declare a file, we use a file pointer. A file pointer is a pointer variable that specifies the next byte to be read or written to. Every time a file is opened, the file pointer points to the beginning of the file. A file is declared as follows:

*FILE \*fp;*

//fp is the name of the file pointer

**Opening, Creating, Closing**

C language provides a number of functions to perform file handling. fopen() is used to create a new file or open an existing file. The syntax is as follows:

*fp = FILE \*fopen(const char \*filename, const char \*mode);*

fp is the file pointer that holds the reference to the file, the filename is the name of the file to be opened or created, and mode specifies the purpose of opening a file such as for reading or writing. FILE is an object type used for storing information about the file stream.

A file can be opened in different modes. Below are some of the most commonly used modes for opening or creating a file.

r : opens a text file in reading mode.

w : opens or creates a text file in writing mode.

a : opens a text file in append mode.

r+ : opens a text file in both reading and writing mode. The file must exist.

w+ : opens a text file in both reading and writing mode. If the file exists, it's truncated first before overwriting. Any old data will be lost. If the file doesn't exist, a new file will be created.

a+ : opens a text file in both reading and appending mode. New data is appended at the end of the file and does not overwrite the existing content.

rb : opens a binary file in reading mode.

wb : opens or creates a binary file in writing mode.

ab : opens a binary file in append mode.

rb+ : opens a binary file in both reading and writing mode, and the original content is overwritten if the file exists.

wb+: opens a binary file in both reading and writing mode and works similar to the w+ mode for binary files. The file content is deleted first and then new content is added.

ab+: opens a binary file in both reading and appending mode and appends data at the end of the file without overwriting the existing content.

A file needs to be closed after a read or write operation to release the memory allocated by the program. In C, a file is closed using the fclose() function. This returns 0 on success and EOF in the case of a failure. An EOF is defined in the library called stdio.h. EOF is a constant defined as a negative integer value which denotes that the end of the file has reached.

*fclose( FILE \*fp );*

**Writing and Reading**

Perform input and output operations in a file. The fputc() and fputs() functions are used to write characters and strings respectively in a file. Let's see how it's used.

*int fputc( int c, FILE \*fp );*

This function writes a character c into the file with the help of the file pointer fp. It returns EOF in the case of an error.

*int fputs( const char \*s, FILE \*fp );*

This function writes string s to the file with the help of the reference pointer fp.

The fgetc(), fgets(), and fscanf() are functions used in C programming language to read characters or strings from a file. Let's discuss these, one by one.

*int fgetc( FILE \*fp );*

The fgetc() accepts the file pointer as the parameter and reads one character at a time from the file. The return value is the character that's read. In case of any error, it returns EOF.

*char \*fgets( char \*buffer, int n, FILE \*fp );*

***File Organization***

File organization ensures that records are available for processing. It is used to determine an efficient file organization for each base relation.

For example, if we want to retrieve employee records in alphabetical order of name. Sorting the file by employee name is a good file organization. However, if we want to retrieve all employees whose marks are in a certain range, a file is ordered by employee name would not be a good file organization.

**Types of File Organization**

There are three types of organizing the file:

1. Sequential access file organization

2. Direct access file organization

3. Indexed sequential access file organization

***Sequential access file organization***

* Storing and sorting in contiguous block within files on tape or disk is called as sequential access file organization.
* In sequential access file organization, all records are stored in a sequential order. The records are arranged in the ascending or descending order of a key field.
* Sequential file search starts from the beginning of the file and the records can be added at the end of the file.
* In sequential file, it is not possible to add a record in the middle of the file without rewriting the file.

***Advantages of sequential file***

* It is simple to program and easy to design.
* Sequential file is best use if storage space.
* Disadvantages of sequential file
* Sequential file is time consuming process.
* It has high data redundancy.
* Random searching is not possible.

***Direct access file organization***

* Direct access file is also known as random access or relative file organization.
* In direct access file, all records are stored in direct access storage device (DASD), such as hard disk. The records are randomly placed throughout the file.
* The records does not need to be in sequence because they are updated directly and rewritten back in the same location.
* This file organization is useful for immediate access to large amount of information. It is used in accessing large databases.
* It is also called as hashing.

***Advantages of direct access file organization***

* Direct access file helps in online transaction processing system (OLTP) like online railway reservation system.
* In direct access file, sorting of the records are not required.
* It accesses the desired records immediately.
* It updates several files quickly.
* It has better control over record allocation.
* Disadvantages of direct access file organization
* Direct access file does not provide back up facility.
* It is expensive.
* It has less storage space as compared to sequential file.

***Indexed sequential access file organization***

* Indexed sequential access file combines both sequential file and direct access file organization.
* In indexed sequential access file, records are stored randomly on a direct access device such as magnetic disk by a primary key.
* This file have multiple keys. These keys can be alphanumeric in which the records are ordered is called primary key.
* The data can be access either sequentially or randomly using the index. The index is stored in a file and read into memory when the file is opened.

***Advantages of Indexed sequential access file organization***

* In indexed sequential access file, sequential file and random file access is possible.
* It accesses the records very fast if the index table is properly organized.
* The records can be inserted in the middle of the file.
* It provides quick access for sequential and direct processing.
* It reduces the degree of the sequential search.
* Disadvantages of Indexed sequential access file organization
* Indexed sequential access file requires unique keys and periodic reorganization.
* Indexed sequential access file takes longer time to search the index for the data access or retrieval.
* It requires more storage space.
* It is expensive because it requires special software.
* It is less efficient in the use of storage space as compared to other file organizations.