**File handling**

**File** is a collection of records related to each other. The file size is limited by the size of memory and storage medium.  
  
**There are two important features of file:**  
  
**1.** File Activity  
**2.** File Volatility  
  
**File activity** specifies percent of actual records which proceed in a single run.  
  
**File volatility** addresses the properties of record changes. It helps to increase the efficiency of disk design than tape.  
  
**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

**1. 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.

**2. 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.

**3. 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.

**File handling operations**

1. **Opening a file-** The fopen() function is used to create a new file or to open an existing file.

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

Here, \*fp is the FILE pointer (FILE \*fp), which will hold the reference to the opened(or created) file. filename is the name of the file to be ope**ned and mode specifies the purpose of opening the file.**

1. **Closing a file -** The fclose() function is used to close an already opened file.

General Syntax - int fclose( FILE \*fp);

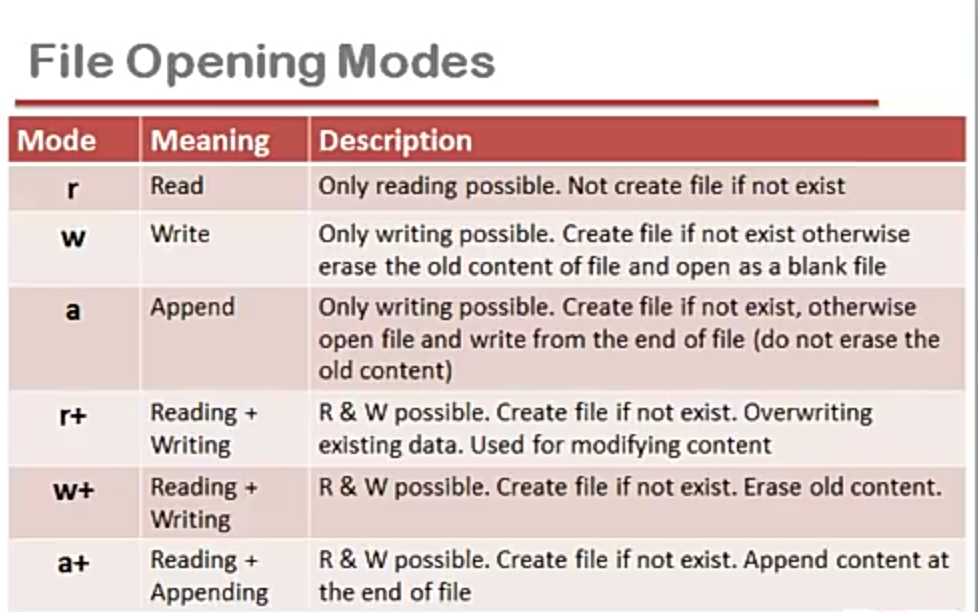
Here fclose() function closes the file and returns zero on success, or EOF if there is an error in closing the file. This EOF is a constant defined in the header file stdio.h.

1. **Writing in a file** – putc(), puts(),fwrite() etc are various function used to write in a file.
2. **Reading a file**- getc(),gets(),fread() etc are various functions used to read a file.

**File handling functions**

|  |  |
| --- | --- |
| **File handling functions** | **Description** |
| [fopen ()](http://fresh2refresh.com/c-programming/c-file-handling/fopen-fclose-gets-fputs-functions-c/) | fopen () function creates a new file or opens an existing file. |
| [fclose ()](http://fresh2refresh.com/c-programming/c-file-handling/fopen-fclose-gets-fputs-functions-c/) | fclose () function closes an opened file. |
| [**getw ()**](http://fresh2refresh.com/c-programming/c-file-handling/getw-putw-functions-c/) | getw () function reads an integer from file. |
| [**putw ()**](http://fresh2refresh.com/c-programming/c-file-handling/getw-putw-functions-c/) | putw () functions writes an integer to file. |
| [fgetc ()](http://fresh2refresh.com/c-programming/c-file-handling/fgetc-function-c/) | fgetc () function reads a character from file. |
| [fputc ()](http://fresh2refresh.com/c-programming/c-file-handling/fputc-function-c/) | fputc () functions write a character to file. |
| [gets ()](http://fresh2refresh.com/c-programming/c-file-handling/fopen-fclose-gets-fputs-functions-c/) | gets () function reads line from keyboard. |
| [puts ()](http://fresh2refresh.com/c-programming/c-file-handling/puts-function-c/) | puts () function writes line to o/p screen. |
| [fgets ()](http://fresh2refresh.com/c-programming/c-file-handling/fgets-function-c/) | fgets () function reads string from a file, one line at a time. |
| [fputs ()](http://fresh2refresh.com/c-programming/c-file-handling/fopen-fclose-gets-fputs-functions-c/) | fputs () function writes string to a file. |
| [feof ()](http://fresh2refresh.com/c-programming/c-file-handling/feof-function-c/) | feof () function finds end of file. |
| [fgetchar ()](http://fresh2refresh.com/c-programming/c-file-handling/fgetchar-function-c/) | fgetchar () function reads a character from keyboard. |
| [fprintf ()](http://fresh2refresh.com/c-programming/c-file-handling/fscanf-fprintf-ftell-rewind-functions-c/) | fprintf () function writes formatted data to a file. |
| [fscanf ()](http://fresh2refresh.com/c-programming/c-file-handling/fscanf-fprintf-ftell-rewind-functions-c/) | fscanf () function reads formatted data from a file. |
| [fputchar ()](http://fresh2refresh.com/c-programming/c-file-handling/fputchar-function-c/) | fputchar () function writes a character onto the output screen from keyboard input. |
| [fseek ()](http://fresh2refresh.com/c-programming/c-file-handling/fseek-seek_set-seek_cur-seek_end-functions-c/) | fseek () function moves file pointer position to given location. |
| [SEEK\_SET](http://fresh2refresh.com/c-programming/c-file-handling/fseek-seek_set-seek_cur-seek_end-functions-c/) | SEEK\_SET moves file pointer position to the beginning of the file. |
| [SEEK\_CUR](http://fresh2refresh.com/c-programming/c-file-handling/fseek-seek_set-seek_cur-seek_end-functions-c/) | SEEK\_CUR moves file pointer position to given location. |
| [SEEK\_END](http://fresh2refresh.com/c-programming/c-file-handling/fseek-seek_set-seek_cur-seek_end-functions-c/) | SEEK\_END moves file pointer position to the end of file. |
| [ftell ()](http://fresh2refresh.com/c-programming/c-file-handling/fscanf-fprintf-ftell-rewind-functions-c/) | ftell () function gives current position of file pointer. |
| [rewind ()](http://fresh2refresh.com/c-programming/c-file-handling/fscanf-fprintf-ftell-rewind-functions-c/) | rewind () function moves file pointer position to the beginning of the file. |
| [getc ()](http://fresh2refresh.com/c-programming/c-file-handling/getc-putc-functions-c/) | getc () function reads character from file. |
| [getch ()](http://fresh2refresh.com/c-programming/c-file-handling/getch-function-c/) | getch () function reads character from keyboard. |
| [getche ()](http://fresh2refresh.com/c-programming/c-file-handling/getche-function-c/) | getche () function reads character from keyboard and echoes to o/p screen. |
| [getchar ()](http://fresh2refresh.com/c-programming/c-file-handling/putchar-getchar-function-c/) | getchar () function reads character from keyboard. |
| [putc ()](http://fresh2refresh.com/c-programming/c-file-handling/getc-putc-functions-c/) | putc () function writes a character to file. |
| [putchar ()](http://fresh2refresh.com/c-programming/c-file-handling/putchar-getchar-function-c/) | putchar () function writes a character to screen. |
| [printf ()](http://fresh2refresh.com/c-programming/c-file-handling/printf-scanf-functions-c/) | printf () function writes formatted data to screen. |
| [sprinf ()](http://fresh2refresh.com/c-programming/c-file-handling/sprintf-function-c/) | sprinf () function writes formatted output to string. |
| [scanf ()](http://fresh2refresh.com/c-programming/c-file-handling/printf-scanf-functions-c/) | scanf () function reads formatted data from keyboard. |
| [sscanf ()](http://fresh2refresh.com/c-programming/c-file-handling/sscanf-function-c/) | sscanf () function Reads formatted input from a string. |
| [remove ()](http://fresh2refresh.com/c-programming/c-file-handling/remove-function-c/) | remove () function deletes a file. |
| [fflush ()](http://fresh2refresh.com/c-programming/c-file-handling/fflush-function-c/) | fflush () function flushes a file. |

**File opening modes**

****

**Pattern Matching in c**

In C Programming,**Pattern matching** is the way of checking a series of pattern or a sequence of digits or string with some other pattern and find out if it matches or not, in pattern recognition, the match usually has to be exact.

A pattern can be a series of digits, a string, different types of colours arranged in order. The order is really important in case of pattern matching.

**String Matching**

Pattern P[m] = P[0,1……m-1]

String T[n]= T[0,1,…..n-1]

P occurs with shift S in T if 0<=S<=n-m and T[S+0,……S+m-1] =P[0,1……m-1]

If P occurs with shift S in T then we call S is a valid shift otherwise S is an invalid shift

char T[] =”NISHAMALHOTRA”;

char P[]=”HA”;

**Naïve Pattern matching algorithm(T,P)**

* 1. n = T length
  2. m = P length
  3. for S = 0 to n-m
  4. if p[0,…..m-1] = T[S+0,……S+m-1]
  5. Printf “Pattern occur with shift S”;