



Computer Programming & Problem Solving

CS100

Mrs Sanga G. Chaki

**Department of Computer Science and Engineering
National Institute of Technology, Goa**

May, 2023



File

1. What is a file?

- a) A named collection of data, stored in memory**

2. How is a file stored?

- a) Stored as sequence of bytes, logically contiguous (may not be physically contiguous on disk).**

3. Types of files:

- a) Text :: contains ASCII codes only**
- b) Binary :: can contain non-ASCII characters**
 - i. Image, audio, video, executable, etc.**

File



- 1. The last byte of a file contains the end-of-file character (EOF).**
- 2. While reading a text file, the EOF character can be checked to know the end.**
- 3. Typical operations on files:**
 - a) Open – fopen()**
 - b) Read – fgetc(), fscanf()**
 - c) Write – fputc(), fprintf()**
 - d) Close – fclose()**



File handling in C – File Open

1. In C we use FILE * to represent a pointer to a file.
2. fopen() is used to open a file.
3. It returns NULL to indicate that it is unable to open the file.

```
FILE *fptr;  
char filename[ ]= "file2.dat";  
  
fptr = fopen (filename,"w");  
  
if (fptr == NULL) {  
    printf ("ERROR IN FILE CREATION");  
    /* DO SOMETHING */  
}
```



Modes for opening files

1. The second argument of `fopen` is the mode in which we open the file.
2. There are three modes.
 - a) `"r"` opens a file for reading.
 - b) `"w"` creates a file for writing, and writes over all previous contents.
 - c) `"a"` opens a file for appending –writing on the end of the file.



File handling in C – File Read

- 1. Once the file has been opened for reading using `fopen()`**
- 2. The file's contents are brought into an associated buffer in memory.**
- 3. A pointer is set up that points to the first character in the buffer.**
- 4. To read the file's contents from memory, we use `fgetc()`.**
- 5. `fgetc()` reads the character from the current pointer position,**
- 6. Advances the pointer position to the next character, and**
- 7. Returns the character that is read**



File handling in C – File write/close

- 1. File write: fputc()**
- 2. A buffer is again associated.**
- 3. After writing, when we close the file, the following happens:**
 - a) The characters in the buffer are written to the file on the disk.**
 - b) At the end of file a character with ASCII value 26 (EOF) gets written.**
 - c) The buffer would be eliminated from memory.**
- 4. File Close: fclose()**
- 5. On closing the file, the buffer associated with the file is removed from memory.**

File handling in C – File Example



```
/* Display contents of a file on screen. */
#include "stdio.h"
main( )
{
    FILE *fp ;
    char ch ;

    fp = fopen ( "PR1.C", "r" ) ;

    while ( 1 )
    {
        ch = fgetc ( fp ) ;

        if ( ch == EOF )
            break ;

        printf ( "%c", ch ) ;
    }

    fclose ( fp ) ;
}
```


File handling in C – File Example



```
#include<stdio.h>
int main(){
    FILE *fptr, *fptr_1;
    char c;
    fptr = fopen("text1.txt","r");
    fptr_1 = fopen("text2.txt","w");
    if(fptr == NULL)
    {
        printf("Error");
        return 0;
    }
    if(fptr_1 == NULL)
    {
        printf("Error");
        return 0;
    }

    while(1){
        c = fgetc(fptr);
        if(c==EOF)
            break;
        printf("%c ",c);
        fputc(c, fptr_1);
    }
    fclose(fptr);
    fclose(fptr_1);
    return 0;
}
```

1. Read from one file and

2. Write it in another file

File handling in C – File Example



```
#include<stdio.h>
int main()
{
    int i, n=2;
    char str[50];
    FILE *fptr = fopen("text3.txt", "w");
    if (fptr == NULL)
    {
        printf("Could not open file");
        return 0;
    }

    for (i = 0; i < n; i++)
    {
        printf("Enter a name");
        scanf("%s", str);
        fprintf(fptr,"%s\n", str);
    }
    fclose(fptr);
    return 0;
}
```

1. Write in a file using **fprintf()**.
2. You can use this to store the output of any program in a separate file.