## UNIT V FILE PROCESSING 9

Files – Types of file processing: Sequential access, Random access – Sequential access file - Random access file - Command line arguments.

---

### Input/ Output operation on File

In the above table we have discussed various file I/O functions to perform reading and writing on file. getc() and putc() are the simplest functions which can be used to read and write individual characters to a file.

**Example:**

```
#include<stdio.h>
int main()
{
        FILE *fp;
        char ch;
        fp = fopen("one.txt", "w");
        printf("Enter data...");
        while( (ch = getchar()) != EOF)
        {
                putc(ch, fp);
        }
        fclose(fp);
        fp = fopen("one.txt", "r");
        while( (ch = getc(fp)! = EOF)
        printf("%c",ch);
        // closing the file pointer
        fclose(fp);
        return 0;
}
```

### Reading and Writing to File using fprintf() and fscanf()

```
#include<stdio.h>
struct emp
{
        char name[10];
        int age;
};
void main()
```

```
{
        struct emp e;
        FILE *p,*q;
        p = fopen("one.txt", "a");
        q = fopen("one.txt", "r");
        printf("Enter Name and Age:");
        scanf("%s %d", e.name, &e.age);
        fprintf(p,"%s %d", e.name, e.age);
        fclose(p);
        Do
        {
                fscanf(q,"%s %d", e.name, e.age);
                printf("%s %d", e.name, e.age);
        }
        while(!feof(q));
}
```

In this program, we have created two FILE pointers and both are refering to the same file but in different modes.

fprintf() function directly writes into the file, while fscanf() reads from the file, which can then be printed on the console using standard printf() function.

### Difference between Append and Write Mode

Write (w) mode and Append (a) mode, while opening a file are almost the same. Both are used to write in a file. In both the modes, new file is created if it doesn't exists already. The only difference they have is, when you open a file in the write mode, the file is reset, resulting in deletion of any data already present in the file. While in append mode this will not happen.

Append mode is used to append or add data to the existing data of file(if any). Hence, when you open a file in Append(a) mode, the cursor is positioned at the end of the present data in the file.

### Reading and Writing in a Binary File

A Binary file is similar to a text file, but it contains only large numerical data. The Opening modes are mentioned in the table for opening modes above.

fread() and fwrite() functions are used to read and write is a binary file.

### Syntax for writing a binary file:

**fwrite(data-element-to-be-written, size_of_elements, number_of_elements, pointer-to-file);**

fread() is also used in the same way, with the same arguments like fwrite() function. Below mentioned is a simple example of writing into a binary file

```
const char *mytext = "The quick brown fox jumps over the lazy dog";
FILE *bfp= fopen("test.txt", "wb");
if (bfp)
{
        fwrite(mytext, sizeof(char), strlen(mytext), bfp);
        fclose(bfp);
}
```