



CS10003: **Programming & Data Structures**

Dept. of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Autumn 2020



File handling

```

#include<stdio.h>
#include<stdlib.h>
int main()
{
    FILE *f;
    int a=10;
    char *s="Welcome", t[100];
    char c;
    f=fopen("newfile.txt", "a");
    fputs("\nAnother line\n", f);
    fputs("Yet another line\n", f);
    fclose(f);
    f=fopen("newfile.txt", "r");
    fgets(t, 3, f);
    printf(" *** %s ****", t);
    fgets(t, 100, f);
    printf(" *** %s ****", t);
    fgets(t, 100, f);
    printf(" *** %s\n ****", t);
    fgets(t, 100, f);
    printf(" *** %s ****", t);
    return 0;
}

```

The file:

```

Hello World! 10 Welcome
Another line
Yet another line
:

```

```

swagato@swagato-Vostro-3478: ~/Desktop/test
swagato@swagato-Vostro-3478:~/Desktop/test$ gcc test2.c
swagato@swagato-Vostro-3478:~/Desktop/test$ ./a.out
swagato@swagato-Vostro-3478:~/Desktop/test$ gcc test1.c
swagato@swagato-Vostro-3478:~/Desktop/test$ gcc test1.c
swagato@swagato-Vostro-3478:~/Desktop/test$ ./a.out

Hello

Wswagato@swagato-Vostro-3478:~/Desktop/test$ gcc test2.c
swagato@swagato-Vostro-3478:~/Desktop/test$ ./a.out
swagato@swagato-Vostro-3478:~/Desktop/test$ gcc test1.c
swagato@swagato-Vostro-3478:~/Desktop/test$ ./a.out

Hello

Wswagato@swagato-Vostro-3478:~/Desktop/test$ gcc test2.c
swagato@swagato-Vostro-3478:~/Desktop/test$ ./a.out
*** He *** ** llo World! 10 Welcome
*** ** Another line

*** ** Yet another line |
***swagato@swagato-Vostro-3478:~/Desktop/test$

```

/* Count chars, spaces, tabs and newlines in a file */

```
int main( )
{
    FILE *fp ;
    char ch ;
    int nol = 0, not = 0, nob = 0, noc = 0 ;
    fp = fopen ( "myfile.txt", "r" ) ;
    while ( 1 )
    {
        ch = fgetc ( fp ) ;
        if ( ch == EOF )
            break ;
        noc++ ;
        if ( ch == ' ' )
            nob++ ;
        if ( ch == '\n' )
            nol++ ;
        if ( ch == '\t' )
            not++ ;
    }
    fclose ( fp ) ;
    printf ( "\nNumber of characters = %d", noc ) ;
    printf ( "\nNumber of blanks = %d", nob ) ;
    printf ( "\nNumber of tabs = %d", not ) ;
    printf ( "\nNumber of lines = %d", nol ) ;
    return 0;
}
```

Sample run

```
Number of characters = 125
Number of blanks = 25
Number of tabs = 13
Number of lines = 22
```

A File-copy Program

```
#include "stdio.h"
int main( )
{
    FILE *fs, *ft ;
    char ch ;
    fs = fopen ( "pr1.txt", "r" ) ;
    if ( fs == NULL )
    {
        puts ( "Cannot open source file" ) ;
        exit(-1) ;
    }
    ft = fopen ( "pr2.txt", "w" ) ;
```

```
    if ( ft == NULL )
    {
        puts ( "Cannot open target file" ) ;
        fclose ( fs ) ;
        exit( ) ;
    }
    while ( 1 )
    {
        ch = fgetc ( fs ) ;
        if ( ch == EOF )
            break ;
        else
            fputc ( ch, ft ) ;
    }
    fclose ( fs ) ;
    fclose ( ft ) ;
    return 0;
}
```

/* Receives strings from keyboard and writes them to file */

```
int main( )
{
    FILE *fp ;
    int l;
    char s[80] ;
    fp = fopen ( "POEM.TXT", "w" );
    if ( fp == NULL )
    {
        puts ( "Cannot open file" );
        exit(-1);
    }
    printf ( "\nEnter a few lines of text:\n" );
    fgets(s, 100, stdin);
    l=strlen(s);
    while (l > 1 )
    {
        fputs ( s, fp );
        fgets(s, 100, stdin);
        l=strlen(s);
    }
    fclose( fp );
    return 0;
}
```

Sample run

Enter a few lines of text:
Shining and bright, they are forever,
so true about diamonds,
more so of memories,
especially yours !

Writing records to a file using structure

```
#include "stdio.h"
int main( )
{
    FILE *fp ;
    char another = 'Y' ;
    struct emp
    {
        char name[40] ;
        int age ;
        float bs ;
    };
    struct emp e ;
    fp = fopen ( "EMPLOYEE.DAT", "w" ) ;
    if ( fp == NULL )
    {
        puts ( "Cannot open file" ) ;
        exit(-1) ;
    }
    while ( another == 'Y' )
    {
        printf ( "\nEnter name, age and basic salary: " ) ;
        scanf ( "%s %d %f", e.name, &e.age, &e.bs ) ;
        fprintf ( fp, "%s %d %f\n", e.name, e.age, e.bs ) ;
        printf ( "Add another record (Y/N) " ) ;
        scanf(“ %c”,&another);
    }
    fclose ( fp ) ;
    return 0;
}
```

Sample run:

```
Enter name, age and basic salary: Sunil 34 1250.50
Add another record (Y/N) Y
Enter name, age and basic salary: Sameer 21 1300.50
Add another record (Y/N) Y
Enter name, age and basic salary: Rahul 34 1400.55
Add another record (Y/N) N
```

Program that reads the employee records created by the previous program.

```
int main( )
{
    FILE *fp ;
    struct emp
    {
        char name[40] ;
        int age ;
        float bs ;
    };
    struct emp e ;
    fp = fopen ( "EMPLOYEE.DAT", "r" ) ;
    if ( fp == NULL )
    {
        puts ( "Cannot open file" ) ;
        exit(-1) ;
    }
    while ( fscanf ( fp, "%s %d %f", e.name, &e.age, &e.bs ) != EOF )
        printf ( "\n%s %d %f", e.name, e.age, e.bs ) ;
    fclose ( fp ) ;
    return 0;
}
```

Output:

```
Sunil 34 1250.500000
Sameer 21 1300.500000
Rahul 34 1400.500000
```


Random access in files

int fseek(FILE *pointer, long int offset, int position)

- Moves the current position to a location specified by the variables position and offset.
- Position can take one of the following three values:
 - SEEK_END : It denotes end of the file.
 - SEEK_SET : It denotes starting of the file.
 - SEEK_CUR : It denotes file pointer's current position.
- offset is specified in bytes.

long ftell(FILE *pointer)

- Returns the current position.

An example

```
char c;  
FILE *fp;  
fp=fopen("newfile.txt", "r+");  
printf("\n%d", ftell(fp));  
c=fgetc(fp);  
c=fgetc(fp);  
fseek(fp, 6, SEEK_CUR);  
printf("%\nld", ftell(fp));  
fputs("new stuff",fp);  
fclose(fp);  
return 0;
```

Output

0
8

Newfile.txt

Initial: Hello world!

Final: Hello wonew stuff



Thank You!