

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

Protected by PDF Anti-Copy Free

(Upgrade to Pro Version to Remove the Watermark)

Lecture-19 By Vivek Dubey

PDF

COMPUTER SCIENCE AND ENGINEERING

CS403PC

COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

CS403PC

BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY

Scheme of Operating System

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



S. No.	Course Code	Course Title	L	T	P	Credits
3	CS403PC	Operating Systems	3	0	0	3
6	CS406PC	Operating Systems Lab	0	0	3	1.5

UNIT-2:

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



- **Process and CPU Scheduling –**
 - Process concepts and scheduling, Operations on Processes, Cooperating Processes, Threads, and Interposes Communication, Scheduling Criteria, **Scheduling Algorithms**, Multiple-Processor Scheduling.
- **System call interface for process management-**
 - fork, exit, wait, waitpid, exec

Basics of File Handling in C



Different operations that can be performed on a file are:

1. Creation of a new file (**fopen** with attributes as “a” or “r” or “w”)
2. Opening an existing file (**fopen**)
3. Reading from file (**fscanf** or **fgets**)
4. Writing to a file (**fprintf** or **fputs**)
5. Moving to a specific location in a file (**fseek**, **rewind**)
6. Closing a file (**fclose**)

Opening or Creating File



- **"r"** – Searches file. If the file is opened successfully `fopen( )` loads it into memory and sets up a pointer which points to the first character in it. If the file cannot be opened `fopen( )` returns `NULL`.
- **"w"** – Searches file. If the file exists, its contents are overwritten. If the file doesn't exist, a new file is created. Returns `NULL`, if unable to open file.

Program to

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Open a File, Write in it, And Close the File

- Output File Name is FromFile2.txt
- Input Content
 - My name is Vivek Dubey.

Part-1

Open a File

```
# include <stdio.h>  
# include <string.h>
```

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



```
int main( )  
{
```

```
    // Declare the file pointer  
    FILE *filePointer ;
```

```
    // Get the data to be written in file  
    char dataToBeWritten[50]  
        = "My name is Vivek DUBey";
```

```
    // Open the existing file using fopen()  
    // in write mode using "w" attribute  
    filePointer = fopen("FromFile2.c", "w") ;
```

Part-2

Write in File

```
// Check if this filePointer is null
// which maybe if the file does not exist
if ( filePointer == NULL )
{
    printf( "FromFile2.c file failed to open." ) ;
}
else
{
    printf("The file is now opened.\n") ;

    // Write the dataToBeWritten into the file
    if ( strlen ( dataToBeWritten ) > 0 )
    {
        // writing in the file using fputs()
        fputs(dataToBeWritten, filePointer) ;
        fputs("\n", filePointer) ;
    }
}
```

Part-3

Close a File

Protected by PDF Anti-Copy Free

(Upgrade to Pro Version to Remove the Watermark)

```
// Closing the file using fclose()  
fclose(fp);
```

```
printf("Data successfully written in file FromFile.c\n");  
printf("The file is now closed.");
```

```
}  
return 0;
```

```
}
```

Program to

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

Open a File, read from file, And Close the File

- Input File is FromFile1.txt
- Output Content
 - My name is Vivek Dubey.

Part-1

Open a File

```
# include <stdio.h>  
# include <string.h>
```

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



```
int main( )  
{
```

```
// Declare the file pointer  
FILE *filePointer ;
```

```
// Declare the variable for the data to be read from file  
char dataToBeRead[50];
```

```
// Open the existing file FromFile2.txt using fopen()  
// in read mode using "r" attribute  
filePointer = fopen("FromFile2.txt", "r") ;
```

Part-2

Write in File

Lecture-19 By Vivek Dubey

```
// Check if this filePointer is null
// which maybe if the file does not exist
if ( filePointer = NULL )
{
    printf( "FromFile.txt file failed to open." ) ;
}
else
{
    printf("The file is now opened.\n") ;
    // Read the dataToBeRead from the file
    // using fgets() method
    while( fgets ( dataToBeRead, 50, filePointer ) != NULL )
    {

        // Print the dataToBeRead
        printf( "%s" , dataToBeRead ) ;

    }
}
```

Part-3

Close a File

Protected by PDF Anti-Copy Free

(Upgrade to Pro Version to Remove the Watermark)

```
// Closing the file using fclose()  
fclose(file_pointer) ;
```

```
printf("\nData successfully read from file FromFile2.txt\n");  
printf("The file is now closed.") ;
```

```
}
```

```
return 0;
```

```
}
```

Copy Contents from One File to Other File



- Input File
 - File Name is FromFile.txt
 - Content - My name is Vivek Dubey.
- Output File
 - File Name is ToFile.txt
 - Content – My name is Vivek Dubey.

Part-1

```
#include <stdio.h>  
#include <stdlib.h> // For exit()
```

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



```
int main()  
{  
    FILE *fptr1, *fptr2;  
    char filename[100], c;  
  
    // Open one file for reading  
    fptr1 = fopen("FromFile.txt", "r");  
    if (fptr1 == NULL)  
    {  
        printf("Cannot open file %s \n", filename);  
        exit(0);  
    }  
}
```

Part-2

```
// Open another file for writing
fptr2 = fopen("ToFile.txt", "w");
if (fptr2 == NULL)
{
    printf("Cannot open file %s \n", filename); exit(0);
}
// Read contents from file
c = fgetc(fptr1);
while (c != EOF)
{
    fputc(c, fptr2);
    c = fgetc(fptr1);
}
printf("\nContents copied to %s", filename);
fclose(fptr1);
fclose(fptr2);
return 0;
```

```
}
```

feesk Command

- `#include <stdio.h>`
- `void main(){`
- `FILE *fp;`
-
- `fp = fopen("test.txt","w+");`
- `fputs("My name is Vivek Dubey", fp);`
-
- `fseek( fp, 11, SEEK_SET );`
- `fputs("Dubey Vivek", fp);`
- `fclose(fp);`
- `}`

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)



The `fseek()` function is used to set the file pointer to the specified offset

Input

My name is Vivek Dubey

Output

My name is Dubey Vivek

rewind()

Input

My name is Vivek

Output

My name is VivekMy name is Vivek

Lecture-19 By Vivek Dubey

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

```
#include<stdio.h>  
#include<conio.h>  
void main()
```

```
FILE *fp;
```

```
char c;
```

```
fp=fopen("test.txt","r");
```

```
while((c=fgetc(fp))!=EOF){
```

```
printf("%c",c);
```

```
}
```

```
rewind(fp); //moves the file pointer at beginning of the file
```

```
while((c=fgetc(fp))!=EOF){
```

```
printf("%c",c);
```

```
}
```

```
fclose(fp);
```

```
getch();
```

```
}
```

the rewind function **sets the file position to the beginning of the file for the stream pointed to by stream**

ftell()

The ftell() function **returns the current file position of the specified stream.**

Input

My name is Vivek Dubey

Output

Size of File: 22 bytes

Lecture-19 By Vivek Dubey

Protected by PDF Anti-Copy Free
(Upgrade to Pro Version to Remove the Watermark)

```
#include <stdio.h>
#include <conio.h>

void main (){
```

- FILE *fp;
- int length;
-
- fp = fopen("test.txt", "r");
- fseek(fp, 0, SEEK_END);
-
- length = ftell(fp);
-
- fclose(fp);
- printf("Size of file: %d bytes", length);
- getch();
- }