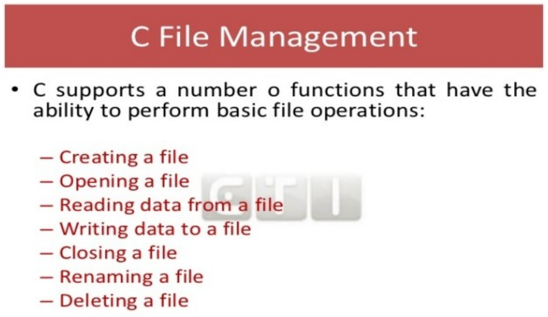
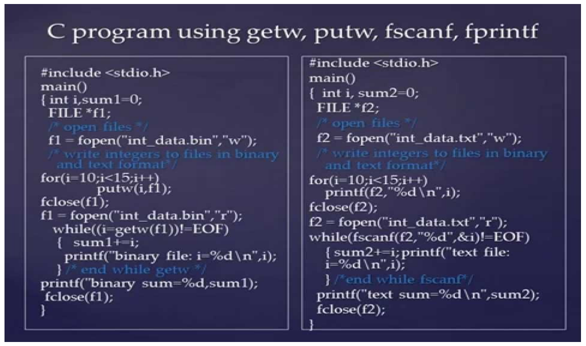
AMIRAJ COLLEGE OF ENGINEERING & TECHNOLOGY

CHAPTER - 10
FILE MANAGEMENT



C File Management

- C supports a number o functions that have the ability to perform basic file operations:
  - Creating a file
  - Opening a file
  - Reading data from a file
  - Writing data to a file
  - Closing a file
  - Renaming a file
  - Deleting a file



C program using getw, putw, fscanf, fprintf

```
#include <stdio.h>
main()
{ int i,sum1=0;
 FILE *f1;
 /* open files */
 f1 = fopen("int_data.bin","w");
 /* write integers to files in binary
   and text format*/
for(i=10;i<15;i++)
        putw(i,f1);
fclose(f1);
f1 = fopen("int_data.bin","r");
  while((i=getw(f1))!=EOF)
   {  sum1+=i;
    printf("binary file: i=%d\n",i);
   } /* end while getw */
printf("binary sum=%d,sum1);
 fclose(f1);
}
```

```
#include <stdio.h>
main()
{ int i, sum2=0;
 FILE *f2;
 /* open files */
 f2 = fopen("int_data.txt","w");
 /* write integers to files in binary
   and text format*/
for(i=10;i<15;i++)
  printf(f2,"%d\n",i);
fclose(f2);
f2 = fopen("int_data.txt","r");
while(fscanf(f2,"%d",&i)!=EOF)
  { sum2+=i;printf("text file:
  i=%d\n",i);
  } /*end while fscanf*/
 printf("text sum=%d\n",sum2);
 fclose(f2);
}
```

Subject : PPS Code : 3110003	**Prepared By:** Asst. Prof. Rupali Patel (CSE Department, ACET)	AMIRAJ COLLEGE OF ENGINEERING & TECHNOLOGY

File Handling in C

The data to be displayed may be very large, and only a limited amount of data can be displayed on the console, and since the memory is volatile, it is impossible to recover the programmatically generated data again and again.

However, if we need to do so, we may store it onto the local file system which is volatile and can be accessed every time.

Here, comes the need of file handling in C.

File Handling in C (cont..)

File handling in C enables us to create, update, read, and delete the files stored on the local file system through our C program.

The following operations can be performed on a file.

- Creation of the new file
- Opening an existing file
- Reading from the file
- Writing to the file
- Deleting the file

Functions for File Handling

No.	Function	Description
1	fopen()	opens new or existing file
2	fprintf()	write data into the file
3	fscanf()	reads data from the file
4	fputc()	writes a character into the file
5	fgetc()	reads a character from file

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

Functions for File Handling (cont..)

6	fclose()	closes the file
7	fseek()	sets the file pointer to given position
8	fputw()	writes an integer to file
9	fgetw()	reads an integer from file
10	ftell()	returns current position
11	rewind()	sets the file pointer to the beginning of the file

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

Open File : FOPEN()

We must open a file before it can be read, write, or update. The fopen() function is used to open a file.

The syntax of the fopen() is given below.

FILE *fopen(**const char** * filename, **const char** * mode);

The fopen() function accepts two parameters:

1. The file name (string). If the file is stored at some specific location, then we must mention the path at which the file is stored. For example, a file name can be like **"c://some_folder/some_file.txt"**.
2. The mode in which the file is to be opened. It is a string.

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

Open File : FOPEN() (cont..)

Mode	Description
r	opens a text file in read mode
w	opens a text file in write mode
a	opens a text file in append mode
r+	opens a text file in read and write mode
w+	opens a text file in read and write mode
a+	opens a text file in read and write mode

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

Open File : FOPEN() (cont..)

Mode	Description
rb	opens a binary file in read mode
wb	opens a binary file in write mode
ab	opens a binary file in append mode
rb+	opens a binary file in read and write mode
wb+	opens a binary file in read and write mode
ab+	opens a binary file in read and write mode

Open File : FOPEN() (cont..)

The fopen function works in the following way.

Firstly, It searches the file to be opened.

Then, it loads the file from the disk and place it into the buffer. The buffer is used to provide efficiency for the read operations.

It sets up a character pointer which points to the first character of the file.

Open File : FOPEN() (cont..)

open a file in read mode.

```
#include<stdio.h>
void main( )
{
FILE *fp ;
char ch ;
fp = fopen("file_handle.c","r")    ;
while ( 1 )
{
        ch = fgetc ( fp ) ;
        if ( ch == EOF )
        break ;
        printf("%c",ch) ;
}
        fclose (fp ) ;
}
```

AMIRAJ COLLEGE OF ENGINEERING & TECHNOLOGY

Closing File : Fclose()

The fclose() function is used to close a file. The file must be closed after performing all the operations on it.

The syntax of fclose() function is given below:

int fclose(**FILE** *fp);

FSEEK() Function

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

It is used to write data into file at desired location.

int fseek(**FILE** *stream, **long int** offset, **int** whence)

There are 3 constants used in the fseek() function for whence: SEEK_SET, SEEK_CUR and SEEK_END.

FSEEK() Function (cont..)

Example :

```
#include <stdio.h>
void main()
{
        FILE *fp;
        fp = fopen("myfile.txt","w+");
        fputs("This is a C Language File", fp);
        fseek( fp, 7, SEEK_SET );
        fputs("sonu jaiswal", fp);
        fclose(fp);
}
```

REWIND() Function

It sets the file pointer to the beginning of the file

Declaration: void rewind(FILE *fp)

Rewind function is used to move file pointer position to the beginning of the file.

In a C program, we use rewind() as below.

```
rewind(fp);
```

REWIND() Function (cont..)

Example:

```
#include<stdio.h>
#include<conio.h>
void main()
{
FILE *fp;
char c;
fp=fopen("file.txt","r");
  while((c=fgetc(fp))!=EOF)
{
printf("%c",c);
}
rewind(fp);
while((c=fgetc(fp))!=EOF)
{
printf("%c",c);
}

fclose(fp);
getch();
}
```

AMIRAJ COLLEGE OF ENGINEERING & TECHNOLOGY

Thank you