Lecture-5

File Handling

What is a FILE

A file is a collection of bytes stored on a secondary storage device, which is generally a disk of some kind. The collection of bytes may be interpreted, for example, as characters, words, lines, paragraphs and pages from a textual document; fields and records belonging to a database; or pixels from a graphical image. There are two kinds of files that programmers deal with

- Text file
- Binary file

What is a FILE

Text Files

A text file can be a stream of characters that a computer can process sequentially. It is not only processed sequentially but only in forward direction. For this reason a text file is usually opened for only one kind of operation (reading, writing, or appending) at any given time.

Binary Files

A binary file is no different to a text file. It is a collection of bytes. In C Programming Language a byte and a character are equivalent. No special processing of the data occurs and each byte of data is transferred to or from the disk unprocessed. C Programming Language places no constructs on the file, and it may be read from, or written to, in any manner chosen by the programmer.

C.

Function Name	**Operation**
fopen()	Creates a new file. Opens an existing file.
fclose	Closes a file which has been opened for use
getc()	Reads a character from a file
putc()	Writes a character to a file
fprintf()	Writes a set of data values to a file
fscanf()	Reads a set of data values from a file
getw()	Reads a integer from a file
putw()	Writes an integer to the file
fseek()	Sets the position to a desired point in the file
ftell()	Gives the current position in the file
rewind()	Sets the position to the beginning of the file

Opening a file:

/* Program to create a file and write some data the file */

```
#include <stdio.h>
#include <stdio.h>
main( )
{
     FILE *fp;
     char stuff[25];
     int index;
     fp = fopen("TENLINES.TXT","w"); /* open for writing */
     strcpy(stuff,"This is an example line.");
     for (index = 1; index <= 10; index++)
          fprintf(fp,"%s Line number %d\n", stuff,
     fclose(fp); /* close the file before ending program */
}
```

/* Program to display the contents of a file on screen */

```
#include <stdio.h>
void main()
{
   FILE *fopen(), *fp;
   int c;
   fp = fopen("prog.c","r");
   c = getc(fp) ;
   while (c!= EOF)
   {
                  putchar(c);
                  c = getc(fp);
   }
   fclose(fp);
}
```

```
/* Program to  create a file and write some data into the file.*/

#include <stdio.h>
int main()
{
 FILE *fp;
 file = fopen("file.txt","w");
 /*Create a file and add text*/
 fprintf(fp,"%s","This is just an example :)"); /*writes data to the file*/
 fclose(fp); /*done!*/
 return 0;
}
```

```
/* Program to add text to a file which already exists and
there is some text in the file.*/

#include <stdio.h>
int main()
{
    FILE *fp
    file = fopen("file.txt","a");
    fprintf(fp,"%s","This is just an example :)"); /*append
some text*/
    fclose(fp);
    return 0;
}
```

A file must be closed as soon as all operations on it have been completed. This would close the file associated with the file pointer. The input output library supports the function to close a file.

Syntax to close file

```
fclose(filepointer);
```

Example

```
#include
void main(void)
{
FILE *myfile;
char c;
myfile = fopen("firstfile.txt", "r");
if (myfile == NULL) printf("File doesn't exist\n");
else {
do {
c = getc(myfile);

putchar(c);

} while (c != EOF);

}
fclose(myfile);

}
```