File Operation and I/O Functions

There are different operations that can be carried out on a file. These are:

- Creation of a new file
- Opening an existing file
- Closing a file
- Reading from a file
- Writing to a file
- Moving to a specific location in a file (seeking)

Operations can be achieved either be through:

- low-level I/O and uses OS level system calls

or

- function in C's standard I/O library.

Library function on file i/o

Functions	Operations
fopen( )	Creates a new file or open an existing file for use
fclose( )	Closes the file which has been opened for use
fgetc( )	Reads a character from a file
fputc( )	Writes a character to a file
fprintf( )	Writes a set of values to a file
fscanf( )	Reads a set of data values from a file
fseek( )	Sets the position to a desired point in the file

Defining and opening a file

FILE is a defined data type. Data structure of a file is defined as **FILE** in the library of standard I/O function definitions. Therefore, all files should be declared as type **FILE** before they are used.

When we open a file, we must specify the operation(s) we want to do with the file.

```
FILE *fp;
fp = fopen("filename", "mode");
```

File Mode and Description

Mode	Description: Open the file for
r	reading purpose
f	writing. If it does not exist, then a new file is created.
a	writing in appending mode. If it does not exist, then a new file is created.
r+	both reading and writing
w+	both reading and writing. It first truncates the file to zero length if it exists, otherwise creates a file if it does not exist.
a+	both reading and writing. It creates the file if it does not exist. The reading will start from the beginning but writing can only be appended.

- FILE *fp1 = fopen("in.c","r");
- FILE *fp2 = fopen("data.c","a");
- FILE *fp3 = fopen("out.c","w");
- FILE *fp4 = fopen("d1", "r+");
- FILE *fp5 = fopen("d2.dat," "a+");
- FILE *fp6 = fopen("data.c","f");

Read contents of a file, store it onto an array and find sum of the elements

Name of file data.in

10 20 30 40 50 60 70 80 90 11 22 33 44 55

66 77 88 99 12 23 34

Read contents of a file, store it onto an array and find sum of the elements


```
int main( )
{
FILE *fp ;
int a[10], i, sum=0;
fp = fopen ( "data.in", "r" ) ;
for(i=0; i<10; i++)
{
fscanf(fp, “%d”, &a[i]);
sum += a[i];
}
```

Read contents of a file, store it onto an array and find sum of the elements


```
int main( )
{
FILE *fp ;
int a[10], i, sum=0;
fp = fopen ( "data.in", "r" ) ;
for(i=0; i<10; i++)
{
fscanf(fp, “%d”, &a[i]);
sum += a[i];
}
printf(”SUM= %d “,sum);
for(i=9; i>=0; i--)
printf(“\n %d “, a[i]);
fclose ( fp ) ;
return( 0);
}
```

Write contents of an array to a file


```
int main( )
{
FILE *fp ;
int i, a[10] = { 2, 4, 6, 8, 1, 3, 5, 7, 9, 0};
fp = fopen ( "data.out", "w" ) ;
for(i=0; i<10; i++)
      fprintf(fp, “%d”, a[i]);

fclose ( fp ) ;
return( 0);
}
```

Display contents of a file on screen


```
int main( )
{
FILE *fp ;
char ch,;
fp = fopen ( "PR1.C", "r" ) ;

while ( (ch = fgetc ( fp ) )  != EOF )
    printf ( "%c", ch ) ;

fclose ( fp ) ;
return( 0);
}
```

Count chars, blank spaces, tabs and newlines in a file


```
FILE *fp ;
char ch ;
int noc = 0, nob = 0, not = 0, non = 0;
fp = fopen ( "PR1.C", "r" ) ;
while ( (ch = fgetc ( fp ) )  != EOF )
{
 noc++ ;
 if ( ch == ' ' )
  nob++ ;
 if ( ch == '\t' )
   not++ ;
 if ( ch == '\n' )
  non++ ;
 }
fclose ( fp ) ;
```

Writing to a file


```
FILE *fp ;
char s[80] = {“KIIT Deemed to be  University”);
fp = fopen ( "POEM.TXT", "w" ) ;

if ( fp == NULL )
{
  puts ( "Cannot open file" ) ;
  exit( ) ;
}
```

A file-copy Program


```
FILE *fs, *ft ;
char ch ;
fs = fopen ( "pr1.c", "r" ) ;
if ( fs == NULL )
{
 puts ( "Cannot open source file" ) ;
 exit( ) ;
}
ft = fopen ( "pr2.c", "w" ) ;
if ( ft == NULL )
{
  puts ( "Cannot open target file" ) ;
  fclose ( fs ) ;
  exit( ) ;
}
```

```
while ((ch = fgetc(fs)) != EOF )
    fputc ( ch, ft ) ;
fclose ( fs ) ;
fclose ( ft ) ;
```