SINDH MADRESSATUL ISLAM UNIVERISTY, KARACHI

DEPARTMENT OF SOFTWARE ENGINEERING

FALL 2022

CSC103 - PROGRAMMING FUNDAMENTALS

ZUBAIR-UDDIN SHAIKH

SECTION SE1A/SE1B/SE1C/CS1D[e]

LAB MANUAL 11

BASIC FILE I/O IN C

BASIC FILE I/O IN C

1. Filing in C

- C language allows programmers to read, write and manipulate files exclusive of the C program.
- To access a file in C, there are some built-in functions and commands.
- Using those functions, a file can be opened, read, written and can be processed according to the algorithm needs.

2. Opening Files

- You can use the **fopen( )** function to create a new file or to open an existing file.
- This call will initialize an object of the type **FILE**, which contains all the information necessary to control the stream. The prototype of this function call is as follows:

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

- Here, **filename** is a string literal, which you will use to name your file, and access **mode** can have one of the following values:

Mode	***Description***
r	Opens an existing text file for reading purpose.
w	Opens a text file for writing. If it does not exist, then a new file is created. Here your program will start writing content from the beginning of the file.
a	Opens a text file for writing in appending mode. If it does not exist, then a new file is created. Here your program will start appending content in the existing file content.
r+	Opens a text file for both reading and writing.
w+	Opens a text file for 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+	Opens a text file for 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.

Example 01: Checking Opening of a File

```
#include<stdio.h>
int main()
{
      char filename[ 80 ];
    FILE *inputFile;

     printf( "Please enter a file name: " );
     scanf( "%s", filename );

     inputFile = fopen( filename, "r" );

     if( inputFile == NULL ) {
          printf("Error\n");
          return 0;
     }

     printf( "%s succesfuly opened", filename );

     fclose( inputFile );
     inputFile = NULL;  // Safety precaution, to prevent trying to use a closed file.
}
```

Example 02: Reading from a Text File (An Integer)

```
#include<stdio.h>
int main()
{
      char filename[ 80 ];
    FILE *inputFile;
     int data;

     printf( "Please enter a file name: " );
     scanf( "%s", filename );

     inputFile = fopen( filename, "r" );

     if( inputFile == NULL ) {
          printf("Error\n");
          return 0;
     }

     int ret = fscanf( inputFile, "%d", &data );
     if(  ret != 1 ) {
          printf( "Error reading from %s\n", filename );
          return 0;
     }

     printf( "Successfully read in %d from %s\n", data, filename );

     fclose( inputFile );
}
```

Example 03: Writing to a Text File (An Integer)

```
include <stdio.h>
#include <stdlib.h> /* For exit() function */
int main()
{
   int num = 97;
   FILE *fptr;
   fptr = fopen("file.txt","w");
   if(fptr == NULL)
   {
      printf("Error!");
      return 0;
   }
   fprintf(fptr,"%d",num);
   fclose(fptr);
   return 0;
}
```

Example 04: Reading From A Text File (An String Using fscanf())

```
#include <stdio.h>
#include <stdlib.h> /* For exit() function*/
int main()
{
   char c[1000];
   FILE *fptr;
      fptr=fopen("file.txt","r");
   if (fptr==NULL){
       printf("Error! opening file");
       return 0;         /* Program exits if file pointer returns NULL. */
   }
   fscanf(fptr,"%s",c);
   printf("Data from file:\n%s",c);
   fclose(fptr);
   return 0;
}
```

Example 05: Writing To A Text File (An String Using fprintf())

```
#include <stdio.h>
#include <stdlib.h>  /* For exit() function */
int main()
{
   char c[1000];
   FILE *fptr;
   fptr=fopen("file.txt","w");
   if(fptr==NULL){
      printf("Error!");
      return 0;
   }
   printf("Enter a sentence:\n");
   gets(c);
   fprintf(fptr,"%s",c);
   fclose(fptr);
   return 0;
}
```

Example 06: Reading From A Text File (An String Using fgets())

```
#include <stdio.h>
#include <stdlib.h> /* For exit() function*/
int main()
{
   char c[1000];
   FILE *fptr;
      fptr=fopen("file.txt","r");
   if (fptr==NULL){
       printf("Error! opening file");
       exit(1);         /* Program exits if file pointer returns NULL. */
   }
   fgets(c,1000,fptr);
   printf("Data from file:\n%s",c);
   fclose(fptr);
   return 0;
}
```

Example 07: Program To Count Characters In A File

```
#include<stdio.h>
int main()
{
      char filename[ 80 ];
    FILE *inputFile;
      char data;
      int count =0;

      printf( "Please enter a file name: " );
      scanf( "%s", filename );

      inputFile = fopen( filename, "r" );

      if( inputFile == NULL ) {
            printf("Error\n");
            return 0;
      }

      do{
            data = getc(inputFile);
            count++;
      }
      while(data != EOF);

      printf( "%s contains %d characters\n", filename, count );

      fclose( inputFile );
}
```

Lab Task 11:

1. **Write a C program to count the number of vowels in a given input file.**
2. **Write a C program to read a four digit integer from an input file and write the sum of those four digits in another file.**
3. **Write a C program to replace all numbers in a file to $.**

Input: I have 4 tables, 2 of them have 3 legs and 2 of them have 4.

Output: I have $ tables, $ of them have $ legs and $ of them have $.

Submission Instructions:

Due Date: Jan 08, 2023

1. **For C files, name your C files as questionNumber_yourRollNum_yourSection_LTNumber.c (e.g. Q1_BSE-22F-123_SE1A_LT1.c).**
2. **Place all files in a folder and name the folder as yourRollNum_yourSection_LTNumber (e.g. BSE-22F-123_SE1A_LT1).**
3. **Compress the folder by using either Winrar or 7Zip with the same name.**
4. **Go to tiny.cc/pffall2022smiu and in the "Coordination Document Folder" open the "PF-Activity Submission Form".**
5. **Fill out all the details with your correct password and submit the form by the due date.**