EEF110E – INTRODUCTION TO PROGRAMMING LANGUAGE (C)

HOMEWORK – II

Zeynep Sena GÜLLER

040210221

Q1: Search for all functions used for file operations in C, explain how they work and write sample codes for each of them.

*To access the files, first a pointer such as "FILE *fptr;" can be defined to the FILE.*

1- *fopen():* The file must be opened with this function before it can be processed. Use of this function is “fptr = fopen(fileName, mode);”. "fileName" is a string of the name of the file to open and "mode" is another string that specifies the opening mode of the file, which can be considered what you want to do to the file. Once the file is opened, it can be accessed with "fptr". If fopen() fails for any reason, for example the file does not exist, it returns “NULL”. The “mode” parameter is elsed by some kind of combination of several specified characters, such as r, w, b, a and +. (r=read an existing text file, w=write to a new text file, a=append to an existing text file, r+=create a new text file for reading or writing, w+=create a new text file for reading and writing, a+=open or create a text file for appending, rb=read an existing binary file, wb=write to a new binary file, ab=append to an existing binary file, r+b=open an existing binary file for reading or writing, w+b=create a new binary file for reading or writing, a+b=open or create a binary file for appending)

2- *fclose():* Even to use another file function, that file must be closed first. In fact, the purpose of this function is to close the file. Its syntax is “fclose(fptr);”.

Below is an example written by using these functions:

```
#include <stdio.h>

int main()

{

FILE *fptr = fopen("demo1.txt", "w");       // Only write mode is on.

if (fptr == NULL)

{

printf("Problem about file opening!");

return -1;

}

fprintf(fptr, "This text was written for homework 2!");      // A string is written to the file.
fclose(fptr);

return 0;

}
```

3- *fgetc():* It can be considered as reading operator (by using characters) for text files. Use of this function is “fgetc(fPtr)”.

Below is an example written by using this function:

```
#include <stdio.h>

int main ()

{

    FILE *fptr = fopen("demo1.txt","r");

     if (fptr == NULL)

      return 0;


    do

    {

        char c = fgetc(fptr);    // Single character input is taken.

        if (feof(fptr))    // It is checked whether there is end of file or not.

            break ;

       printf("%c", c);

    }

   while(1);

   fclose(fptr);

   return(0);

}
```

4- *fputc():* This function may be thought as writing operator. It is is used to write a single character at a time to a given file. This way, the characters are compile one by one. Its syntax is “fputc(ch, fPtr)”. Here, “ch” is the character you want to write to the file.

Below is an example written by using this function:

```
#include <stdio.h>
int main()
{
  FILE *fptr;
  fptr = fopen("demo1.txt", "w");
  fputc('a',fptr);    //Single character (a) is written into “demo1”.
  fclose(fptr);
}
```

5- *fgets():* It can be considered as reading operator (by using strings) for text files. Use of this function is “fgets(str,n,fptr)”. Here, “str” pointer to strings where they read is copied. Also, “n” is maximum number of characters to be read from file.

Below is an example written by using this function:

```
#include <stdio.h>

int main()

{

char arr[20];

FILE *fptr = fopen("demo1.txt", "r");     // Only reading mode is on.

if (fptr == NULL)

{

printf("Problem about file opening!");

return -1;

}

fgets(arr, 15, fptr);    // Most 15 characters are read from the file.

printf("%s\n", arr);

fclose(fptr);

return 0;

}
```

6- *fputs():* This function is used as writing operator like fputc(). But difference is that what is written is a string. Its syntax is fputs(str,fptr).

Below is an example written by using this function:

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

void writeToFile(char str[])
{
FILE *fptr;
fptr = fopen("demo2.txt", "w");

fputs(str, fptr);     // String is written to file.
fclose(fptr);
}

int main()
{
char str[20];
strcpy(str, "This is  C programming language!");
writeToFile(str);
return 0;
}
```

7- *fprintf():* This function is very similar to printf() function, except that fprintf() needs a FILE pointer.
8- *fscanf():* This function is also very similar to scanf() function, except same argument as above.
9- *feof():* It is used to if the end of the file was reached.

Its sample has been done before.

10- *fgetpos():* It is used to get the initial position of the file.
11- *rewind():* It sets the initial position to beginning file position.

Below is an example written by using these functions:

```
#include <stdio.h>

#include <stdlib.h>

int main(void)

{

   FILE *fptr;

   fpos_t pos;

   int id;


   if ((fptr = fopen ("demo3.txt", "w+")) == NULL)

  {

      printf("Problem about file opening!");

      return 1;

   }

   fgetpos(fptr, &pos);    // Initial position of this file is gotten.
   printf("Initial file position: %ld\n", pos);
   fputs("Hello from ITU!", fptr);
   fgetpos(fptr, &pos);
   printf("Final file position: %ld\n", pos);
   rewind(fptr);     // Final position of file is set the the beginning.
   while ((id=getc(fptr))!= EOF)
  {
     printf("%c", id);
   }
   fclose(fptr);
   return 0;
}
```

12- *fsetpos():* It sets the initial position to desired file position.

Below is an example written by using this function:

```
#include <stdio.h>
int main ()
{
  FILE *fptr;
  fpos_t pos;

  fptr= fopen("demo4.txt","w+");
  fgetpos(fptr, &pos);
  fputs("HelloWorld!", fptr);

  fsetpos(fptr, &pos);     // First part ("Hello") is override.
  fputs("Hey! ", fptr);
  fclose(fptr);
  return(0);
}
```

Q2: Research and explain the following in detail;

- What is dll?
- What is dll used for?
- How is dll created and used in Dev C?

A DLL is a library that contains code and data that can be used by more than one program at the same time. Actually, the task of DLL files is to carry out the operations that the running programs have done jointly in a single file. In addition, there are many reasons why DLL is used. For instance, the use of DLL promotes that the operating system and the programs load and run faster because of efficient memory usage and reducing disk space. Furthermore, in dev c++, what needs to be done to create a DLL file can be summarized as file > new > project > DLL > C project.

A simple example is given below:

```
#include<stdio.h>

#include "hellodll.h"     // Include dll header file



void helloDLL()      // Define your functions in here .c file

{

   printf ("Hello DLL.\n");         //This will be printed when user call this function

}                                   // from his C or C++ applications.
```