18/20

some exp. were too naive.

CSE 101 ASSIGNMENT

GROUP- 0

Topic: Experiment with FILE open mode and various operations

Submitted to : Dr. Mohammad Saifur Rahman

Submitted by : 1805072
1805084
1805096
1805108
1805120

Date of Submission : 10 September, 2019

If we open or create a file in “w” mode, what happens when we perform different kinds of operations in the program?

Let’s have a look.

fgetc Everything in the file gets removed and the file is ready to be written, so it doesn’t display anything in the output.

CODE:
```
FILE *f1;
if((f1=fopen("myfile.txt","w"))!=NULL){
char ch=fgetc(f1);
printf("%c",ch);}
fclose(f1);
```
INPUT: a

OUTPUT: Remains blank.

fputc Writes character in a file.

CODE:
```
FILE *f1;
if((f1=fopen("myfile.txt","w"))!=NULL){
printf("%c",fputc(f1));}
fclose(f1);
```
INPUT: a

OUTPUT: Writes ‘a’ in myfile.txt.

fgets Everything in the file gets deleted, so it doesn’t display anything in output.

CODE:
```
FILE *f1;
char ch[100];
f1=fopen("myfile.txt","w");
fgets(ch,100,f1);
printf("%s",ch);
fclose(f1);
```

INPUT: HELLO WORLD!

OUTPUT: Remains blank.

fputs Writes a string in the file until it finds a new line or NULL character.

CODE:
```
FILE *f1;
if((f1=fopen("myfile.txt","w"))!=NULL)
fputs(“Hey there!",f1);
fclose(f1);
```

OUTPUT: Writes ‘Hey there!’ in myfile.txt.

fprintf Any number or character or string given as input can be printed in the file.

CODE:
```
FILE *f1;
char a;
if((f1=fopen("myfile.txt","w"))!=NULL)
fprintf(f1,“%c”, a);
```

INPUT: S

OUTPUT: S

fscanf Everything in the file gets deleted and the file is ready to be written, so printing the output shows only garbage value.

CODE:
```
FILE *f1;
char a;
if((f1=fopen("myfile.txt","w"))!=NULL)
fscanf(f1,“%c”, a);
fclose(f1);
```

INPUT: S

OUTPUT: Garbage value.

fseek

Works fine and can point to any position.

CODE:

```
FILE *f1;
if((f1=fopen("myfile.txt","w"))!=NULL)
fseek(f1,10,SEEK_END);
fclose(f1);
```

OUTPUT: Points perfectly.

ftell

Tells the position of the pointer as long as file size is less than the size of a long integer.

CODE:

```
FILE *f1;
char a;
if((f1=fopen("myfile.txt","w")!=NULL)
printf("%ld",ftell(f1));
fclose(f1);
```

OUTPUT: 10

rewind

Brings the position of the pointer to the very beginning of the file.

CODE:

```
FILE *f1;
if((f1=fopen("myfile.txt","w"))!=NULL)
rewind(f1);
fclose(f1);
```

OUTPUT: Works fine.

feof

CODE:

```
FILE *f1;
char a;
if((f1=fopen("myfile.txt","w"))!=NULL){
while(!feof(f1))
printf(1);}
fclose(f1);
```

OUTPUT: The program crash.

fwrite Writes in a binary file.

CODE:
```
FILE *f1;
int a;
if((f1=fopen("myfile.txt","w"))!=NULL){
scanf("%d",&a);}
fwrite(&a,sizeof(a),1,fp);
fclose(fp);
```

INPUT: 1

OUTPUT: 1

fread Reads from a binary file.

CODE:
```
FILE *f1;
int a;
if((f1=fopen("myfile.dat","w"))!=NULL){
while(fread(&a,sizeof(a),1,fp)>0)
printf("%d",a);}
fclose(fp);
```

OUTPUT: Does not read anything.

ferror Checks if any error has occurred.

CODE:

```
FILE *f1;
char *s = "Important information";
if((f1 = fopen("input.txt","w"))!=NULL){
fprintf(f1, "%s\n", s);}
if (ferror(f1)){
printf(“Error");
    clearerr(f1);}
```

OUTPUT: No output is shown due to no error.

fflush Flushes the buffer to load data from the file. If successful, returns 0, else returns EOF.

CODE:

```
char buf[50];
FILE *fp;
fp = fopen("output.txt", “w");
if (fp)
{
fputs("This is a line.", fp);
fflush(buf);
fgets(buf, 10, fp);
puts(buf);
fclose(fp);
}
```

OUTPUT: Clears the file when opened. So does not show any output.

More Observations

- **A file is opened in “w” mode with.bin extension:**

Observation: A character or string using fwrite() and fprintf() functions can be written but an integer with fwrite() and putw() proves otherwise and writes a character according to ASCII value instead.An integer can be written using fprintf().

- **A file is opened in "w" mode with .txt extension.**

Observation: Same result with .bin extension.

CODE:

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
FILE *fp;
if((fp=fopen("test.txt","w"))==NULL){
printf("file not found");
exit(1);}
int  x=97;
putw(x,fp);
fprintf(fp,“ %d",x);
fwrite(&x,sizeof(int),1,fp);
fclose(fp);
}
```

OUTPUT: a 97 a

NOTE: putw() is a funtion which is similar to putc().The only difference is that it deals with integers and after writing a word it gives a tab.

- **No difference found between “w” or “wb” mode.**

CODE:

```
#include<stdio.h>
int main()
{
FILE *fp;
char ch,ara[100];
int ara1[]={1,2,3,4,5};
if((fp=fopen("cse.txt","w"))==NULL){
printf("Can not open file");
exit(1);
}
gets(ara);
fputc('a',fp);
fputs(ara,fp);
fprintf(fp,"%s",ara);
fwrite(ara1,sizeof(int),5,fp);
fclose(fp);
return 0;
}
```

OUTPUT: Same if we open this file in **“wb”** mode.

THANK
YOU!