CSE Assignment

Group: 7

Topic: w+b mode in file

Group Members' Student ID and Name:

1805067 Muhammad Ehsanul Kader
1805079 Sumaiya Sultana Any
1805091 Ruhul Azgor
1805103 Sohidul Islam
1805115 Azizur Rahman Anik

1. What will happen if we open a binary file in w+b mode and after writing to it, read the file?

Experimental Code:

```
 FILE *fp;
    int i;
    double d[10]= {10.23, 19.87, 1002.23, 12.9, 0.897, 11.45, 75.34, 0.0, 1.01,
875.875};


    if((fp=fopen("myfile", "w+b"))==NULL)
    {
        printf("Cannot open file\n");
        exit(1);
    }
    if(fwrite(d, sizeof(d), 1, fp)!=1)
    {
        printf("Write error\n");
        exit(1);
    }

    for(i=0; i<10; i++)
        d[i]=-1.0;

    if(fread(d, sizeof(d), 1, fp)!=1)
    {
        printf("Read error\n");
        exit(1);
    }

    fclose(fp);

    for(i=0; i<10; i++)
    {
        printf("%.2f\t", d[i]);
    }
```

Expected Output: 10.23 19.87 1002.23 12.90 0.90 11.45 75.34 0.00 1.01 875.88

Actual Output: 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

Comment: We have to use fseek to bring back the pointer at the starting position

or we may close the file and again open it to read, then the saved data occurs perfectly

2. What will happen if we open a binary file in w+b mode and without writing anything, we run read operation?

Experimental Code:

```
 FILE *fp;
    int i;
    double d[10]={10.23, 19.87, 1002.23, 12.9, 0.897, 11.45, 75.34, 0.0, 1.01,
875.875};


    if((fp=fopen("myfile", "w+b"))==NULL)
    {
        printf("Cannot open file\n");
        exit(1);
    }


    if(fread(d, sizeof(d), 1, fp)!=1)
    {
        printf("Read error\n");
        exit(1);
    }

    fclose(fp);

    for(i=0;i<10;i++)
    {
        printf("%.2f\t",d[i]);
    }
```

Output: Read error

3. What will happen if we first write something, then do fseek and again try to write something? What will happen if we try to read that file?

Experimental Code:

```
   FILE *fp;
    int i;
    double d[10]= {10.23, 19.87, 1002.23, 12.9, 0.897, 11.45, 75.34, 0.0, 1.01,
875.875};
    int a[5]= {1,2,3,4,5};


    if((fp=fopen("myfile", "w+b"))==NULL)
    {
        printf("Cannot open file\n");
        exit(1);
    }

    if(fwrite(d, sizeof(d), 1, fp)!=1)
    {
        printf("Write error\n");
        exit(1);
    }

    fseek(fp, sizeof(d), SEEK_SET);

    if(fwrite(a, sizeof(a), 1, fp)!=1)
```

```
    {
        printf("Write error\n");
        exit(1);
    }

    fseek(fp, 0, SEEK_SET);

    for(i=0; i<10; i++)
        d[i]=-1.0;

    for(i=0;i<5;i++)
        a[i]=-1;


    if(fread(d, sizeof(d), 1, fp)!=1)
    {
        printf("Read error\n");
        exit(1);
    }

     if(fread(a, sizeof(a), 1, fp)!=1)
    {
        printf("Read error\n");
        exit(1);
    }

    for(i=0;i<10;i++)
        printf("%.2lf\t",d[i]);

    printf("\n\n");

    for(i=0;i<5;i++)
    {
        printf("%d\t",a[i]);
    }


    fclose(fp);
```

Output: 10.23 19.87 1002.23 12.90 0.90 11.45 75.34 0.00 1.01 875.88

1 2 3 4 5

Comment: That works fine if we seek that number of bytes we have written at first. But if we seek less or more than that:

Experimental codes:

```
fseek(fp, 56, SEEK_SET);
fseek(fp, sizeof(a), SEEK_SET);
fseek(fp, sizeof(d)+12, SEEK_SET);
```

Corresponding Outputs:

```
Read error
Read error
10.23   19.87   1002.23 12.90   0.90    11.45   75.34   0.00    1.01    875.88

0       0       0       1       2
```

4. What will happen if we use 2 different file pointers to do w+b and rb operation in a same file?

Experimental Code:

```
    FILE *fp, *fp2;
    int i;
    int a[5]= {1,2,3,4,5};


    if((fp=fopen("myfile", "w+b"))==NULL)
    {
        printf("Cannot open file\n");
        exit(1);
    }

    if((fp2=fopen("myfile", "rb"))==NULL)
    {
        printf("Cannot open file\n");
        exit(1);
    }


    if(fwrite(a, sizeof(a), 1, fp)!=1)
    {
        printf("Write error\n");
        exit(1);
    }

    fseek(fp, 0, SEEK_SET);
  //fseek(fp2,0,SEEK_SET);

    for(i=0;i<5;i++)
        a[i]=-1;

    if(fread(a, sizeof(a), 1, fp2)!=1)
    {
        printf("Read error\n");
        exit(1);
    }


    for(i=0;i<5;i++)
    {
        printf("%d\t",a[i]);
    }


    fclose(fp);
    fclose(fp2);
```

Output: 1 2 3 4 5

Comment: It works fine if we do fseek with fp pointer but if we do fseek with fp2 pointer, than we get read error.
The fseek operation doesn't work with fp2 pointer but we can read from file with fp2 pointer in this program.
The reason behind this kind of act is uncleared to us.

5. What will happen if we do w+b operation & rb operation over the same file using different file pointers and try to read the file using both file pointers?

Experimental Code:

```
if(fread(a, sizeof(a), 1, fp2)!=1)
    {
        printf("Read error\n");
        exit(1);
    }

    for(i=0; i<5; i++)
    {
        printf("%d\t",a[i]);
    }

    for(i=0; i<5; i++)
        a[i]=-1;

    if(fread(a, sizeof(a), 1, fp)!=1)
    {
        printf("Read error\n");
        exit(1);
    }

    printf("\n");

    for(i=0; i<5; i++)
    {
        printf("%d\t",a[i]);
    }


    fclose(fp2);
    fclose(fp);
```

Output:

```
1       2       3       4       5
1       2       3       4       5
```

Comment: It works fine

6. What will happen if we write two different data on a same file using w+b operation with different file pointers?

**In case of reopening the file:

Experimental Code:

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

int main()
{
    FILE *fp, *fp2;
    int i;
    int a[5]= {1,2,3,4,5};
    double d[10]= {10.23, 19.87, 1002.23, 12.9, 0.897, 11.45, 75.34, 0.0, 1.01,
875.875};

    if((fp=fopen("myfile", "w+b"))==NULL)
    {
        printf("Cannot open file\n");
```

```
    exit(1);
}

if((fp2=fopen("myfile", "w+b"))==NULL)
{
    printf("Cannot open file\n");
    exit(1);
}

if(fwrite(d, sizeof(d), 1, fp2)!=1)
{
    printf("Write error\n");
    exit(1);
}


if(fwrite(a, sizeof(a), 1, fp)!=1)
{
    printf("Write error\n");
    exit(1);
}

fclose(fp);
fclose(fp2);


if((fp2=fopen("myfile", "rb"))==NULL)
{
    printf("Cannot open file\n");
    exit(1);
}

if((fp=fopen("myfile", "rb"))==NULL)
{
    printf("Cannot open file\n");
    exit(1);
}


for(i=0; i<5; i++)
    a[i]=-1;


if(fread(a, sizeof(a), 1, fp)!=1)
{
    printf("Read error\n");
    exit(1);
}

for(i=0; i<5; i++)
{
    printf("%d\t",a[i]);
}

  for(i=0; i<10; i++)
    d[i]=-1.0;


if(fread(d, sizeof(d), 1, fp2)!=1)
{
    printf("Read error\n");
    exit(1);
}
```

```
    printf("\n");

    for(i=0; i<10; i++)
    {
        printf("%.3lf\t",d[i]);
    }


    fclose(fp2);
    fclose(fp);


    return 0;

}
```

Output:

```
-1889785610     1076131266      1374389535      1077141176      171798692
10.230  19.870  1002.230        12.900  0.897   11.450  75.340  0.000   1.010
875.875
```

Comment: Though I wrote the floating numbers first & then the integer numbers , in output I had the floating numbers and the integer numbers were lost!

**If I only read the floating point numbers:

Experimental Code:

```
   for(i=0; i<10; i++)
        d[i]=-1.0;

    if(fread(d, sizeof(d), 1, fp2)!=1)
    {
        printf("Read error\n");
        exit(1);
    }

    for(i=0; i<10; i++)
    {
        printf("%.3lf\t",d[i]);
    }


    fclose(fp2);
```

Output: 10.230 19.870 1002.230   12.900 0.897   11.450 75.340 0.000 1.010   875.875

**If I only read the integer numbers:

Experimental code:

```
    if((fp=fopen("myfile", "rb"))==NULL)
    {
        printf("Cannot open file\n");
        exit(1);
    }
```

```
    for(i=0; i<5; i++)
        a[i]=-1;

    if(fread(a, sizeof(a), 1, fp)!=1)
    {
        printf("Read error\n");
        exit(1);
    }

    for(i=0; i<5; i++)
    {
        printf("%d\t",a[i]);
    }
```

Output: -1889785610 1076131266 1374389535 1077141176 171798692

**In case of using fseek after writing:

Experimental Code:

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

int main()
{
    FILE *fp, *fp2;
    int i;
    int a[5]= {1,2,3,4,5};
    double d[10]= {10.23, 19.87, 1002.23, 12.9, 0.897, 11.45, 75.34, 0.0, 1.01, 875.875};

    if((fp=fopen("myfile", "w+b"))==NULL)
    {
        printf("Cannot open file\n");
        exit(1);
    }

    if((fp2=fopen("myfile", "w+b"))==NULL)
    {
        printf("Cannot open file\n");
        exit(1);
    }

    if(fwrite(d, sizeof(d), 1, fp2)!=1)
    {
        printf("Write error\n");
        exit(1);
    }


    if(fwrite(a, sizeof(a), 1, fp)!=1)
    {
        printf("Write error\n");
        exit(1);
    }


    fseek(fp, 0, SEEK_SET);

    for(i=0; i<5; i++)
        a[i]=-1;

    if(fread(a, sizeof(a), 1, fp)!=1)
```

```
    {
        printf("Read error\n");
        exit(1);
    }

    for(i=0; i<5; i++)
    {
        printf("%d\t",a[i]);
    }

    for(i=0; i<10; i++)
        d[i]=-1.0;

    //fseek(fp, 0, SEEK_SET);

    if(fread(d, sizeof(d), 1, fp2)!=1)
    {
        printf("Read error\n");
        exit(1);
    }

    printf("\n");

    for(i=0; i<10; i++)
    {
        printf("%.3lf\t",d[i]);
    }


    fclose(fp2);
    fclose(fp);


    return 0;

}
```

Output:

```
1       2       3       4       5
0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
```

In case of using String:

Experimental Code:

```
FILE *fp,*fp2;
    fp=fopen("my file.txt","w+b");
    fp2=fopen("my file.txt","w+b");
    int i;
    char p='P',q='Q',s;
    if(fwrite(&p,sizeof p,1,fp)!=1) printf("error1\n");
    if(fwrite(&q,sizeof q,1,fp2)!=1) printf("error1\n");
    fseek(fp,0,SEEK_SET);
    i=ftell(fp);
    fread(&s,sizeof s,1,fp);
    printf("%c \n",s);
    fseek(fp2,0,SEEK_SET);
    fread(&s,sizeof s,1,fp2);
    printf("%c \n",s);
    fclose(fp);
    fclose(fp2);
```

```
    fp=fopen("my file.txt","rb");
    fp2=fopen("my file.txt","rb");
    //fseek(fp,0,SEEK_SET);
    i=ftell(fp);
    fread(&s,sizeof s,1,fp);
    printf("%c \n",s);
    //fseek(fp2,0,SEEK_SET);
    fread(&s,sizeof s,1,fp2);
    printf("%c \n",s);
```

Output:
P
Q
Q
Q

Comment: If we read with both the file pointers in w+b mode, we get both P and Q. But if we close the file and open it in rb mode, then we get only Q. Can't figure out the reasons in this experiment number 6.

7. What will happen if we open a binary file using 2 different pointers for w+b and rb operation and try to write and read exchanging the pointers?

Experimental Code:

```
    FILE *fp, *fp2;
    int i;
    int a[5]= {1,2,3,4,5};

    if((fp=fopen("myfile", "w+b"))==NULL)
    {
        printf("Cannot open file\n");
        exit(1);
    }

    if((fp2=fopen("myfile", "rb"))==NULL)
    {
        printf("Cannot open file\n");
        exit(1);
    }

    if(fwrite(a, sizeof(a), 1, fp2)!=1)
    {
        printf("Write error\n");
        exit(1);
    }


    for(i=0; i<5; i++)
        a[i]=-1;

    if(fread(a, sizeof(a), 1, fp)!=1)
    {
        printf("Read error\n");
        exit(1);
    }

    for(i=0; i<5; i++)
    {
        printf("%d\t",a[i]);
```

```
    }

    fclose(fp);
    fclose(fp2);
```

Output: Write error

Comment: But if we write in the right way and then read with the other pointer, it works fine.

Experimental Code:

```
 if(fwrite(a, sizeof(a), 1, fp)!=1)
    {
        printf("Write error\n");
        exit(1);
    }

    fseek(fp, 0, SEEK_SET);


    for(i=0; i<5; i++)
        a[i]=-1;

    if(fread(a, sizeof(a), 1, fp2)!=1)
    {
        printf("Read error\n");
        exit(1);
    }

    for(i=0; i<5; i++)
    {
        printf("%d\t",a[i]);
    }

    fclose(fp);
    fclose(fp2);
```

Output: 1 2 3 4 5

8. Can we write and read using fprintf and fscanf in a binary file?

Experimental Code:

```
    FILE *fp, *fp2;
    int i;
    int a[5]= {1,2,3,4,5};

    if((fp=fopen("myfile", "w+b"))==NULL)
    {
        printf("Cannot open file\n");
        exit(1);
    }

    for(i=0;i<5;i++)
    {
        fprintf(fp,"%d ",a[i]);
    }

    for(i=0; i<5; i++)
        a[i]=-1;
```

```
    fseek(fp, 0, SEEK_SET);

    for(i=0;i<5;i++)
    {
        fscanf(fp, "%d",&a[i]);
    }

    for(i=0; i<5; i++)
    {
        printf("%d\t",a[i]);
    }
```

Output: 1 2 3 4 5

Comment: We can do binary file operations using fscanf and fprintf. We can also use fputs and fgets in binary file

Experimental Code:

```
    FILE *fp;
    char a[4]= "abcd";

    if((fp=fopen("myfile", "w+b"))==NULL)
    {
        printf("Cannot open file\n");
        exit(1);
    }

    fputs(a,fp);
    fseek(fp,0,SEEK_SET);

    fgets(a,5,fp);
    printf("%s\n",a);

    fclose(fp);
```

Output: abcd

#Experiment 9:Is it possible to use a single file pointer both for reading and writing a file at a time?(Also using the reading and writing syntax of text mode fprintf and fscanf)

Expeimental code:

```
#include<stdio.h>
int main()
{
    char ch[50];
    FILE* fp;

    fp=fopen("paragraph.txt","wb+");
    fp=fopen("paragraph.txt","rb+");


    fprintf(fp,"hello I am a student");

    fgets(ch,49,fp);
    printf("%s",ch);

    fclose(fp);

    return 0;
```

```
}
```

output:nothing is printed in the console..But in the text file "hello I am a student" is printed...

decision:in this case,writing is possible..But reading from file is not possible using same pointer for reading and writing of a same file..It is really interesting that
fprintf and fscanf can also be used in binary mode...

#Experiment 10:Is it possible to use interchanged file pointers for reading and writing of a same file??(Here also we want to verify if fscanf and fprintf can be used for
binary mode)
Experimental code:

```
#include<stdio.h>
int main()
{
    char ch[50];
    FILE* fp,*fp1,*fp2;
    fp=fopen("paragraph.txt","rb+");
    fp1=fopen("paragraph.txt","wb+");
    fp2=fp1;
    fp1=fp;
    fp=fp2;
    fprintf(fp,"hello I am a programmer");
    fgets(ch,49,fp1);
    printf("%s",ch);
    fclose(fp);
    fclose(fp1);
    return 0;

}
```

output:garbage value is printed on the console.But in the text file "hello I am a programmer" is printed..
decision:in this case,interchanging pointers writing in the file is possible..but reading from this file causes problems...It is really interesting that
fprintf and fscanf can also be used in binary mode...

Experiment 11:
code:

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    char s1[]="bangladesh university of engineering and technology";

    FILE *fp;
    if((fp=fopen("a.txt","w+b"))==NULL){
        printf("cannot open file");
        exit (0);
    }
    int n=100;
    double d=134.88888;
```

```
    fprintf(fp,"%d%s%lf",n,s1,d);

    fclose(fp);

    if((fp=fopen("a.txt","r+b"))==NULL){    //if I write r+ or r instead of r+b
the result doesnt change
        printf("cannot open file");
        exit (0);
    }

    fscanf(fp,"%d %[^\n]s%lf",&n,s1,&d);

    printf("%d\n%s\n%lf",n,s1,d);
    fclose(fp);
    return 0;
}
```

expected output:

```
100
bangladesh university of engineering and technology
134.888880
```

output:

```
100
bangladesh university of engineering and technology134.888880
134.888880
```

comments:

*****the programme doesn"t return 0

*****after initializing the string like this:

```
      char *s1="bangladesh university of engineering and technology";
```

the programme compiled but it didn't print any output on console and returned some garbage value

*****if the string has only one word & if %s is used instead of %[^\n]s it doesnt change the result:

```
      100
      bangladesh134.888880
      134.888880
```

Experiment 12:

code:

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    char s1[1000]={"dhaka is a city"},s2[1000];

    FILE *fp;
    if((fp=fopen("a.txt","w+b"))==NULL){
        printf("cannot open file");
        exit (0);
    }

    int n=100;
```

```
    double d=134.88888;

    fwrite(&n,sizeof(int),1,fp);
    fwrite(s1,sizeof(s1),1,fp);
    fwrite(&d,sizeof(d),1,fp);

    fclose(fp);

    if((fp=fopen("a.txt","r+b"))==NULL){  //if I write r+,r instead of r+b it
works fine
        printf("cannot open file");
        exit (0);
    }

    fread(&n,sizeof(int),1,fp);
    fread(s1,sizeof(s1),1,fp);
    fread(&d,sizeof(d),1,fp);

    printf("%d %s %lf",n,s1,d);
    fclose(fp);
    return 0;
}

expected output:

100
bangladesh university of engineering and technology
134.888880

output:
100
bangladesh university of engineering and technology134.888880
134.888880


comments:
*****the programme returns 0

***** after initializing the string like this:

      char *s1="bangladesh university of engineering and technology";

      the programme compiled but it didn't print any output on console and
returned some garbage value
```