

Assignment Number

Problem Statement

Program in C to create a file that consist of some numbers and separate that numbers in two consecutive files such that one contains the even numbers and other containing the odd ones.

Theory

A **file** represents a sequence of bytes on the disk where a group of related data is stored. File is created for permanent storage of data. It is a ready made structure.

In C language, we use a structure **pointer of file type** to declare a file. Using file Handling programs in C we can store the fetched information in a file. Different operations that can be performed on a file are:

1. Creation of a new file (fopen with attributes as "a" or "a+" or "w" or "w++")
2. Opening an existing file (fopen)
3. Reading from file (fscanf or fgetc)
4. Writing to a file (fprintf or fputs)
5. Moving to a specific location in a file (fseek, rewind)
6. Closing a file (fclose)

Algorithm

Input: Some numbers will be given by the user to the program.

Output: Creation of two separate files data_even.txt and data_odd.txt. One containing the even numbers and one containing the odd numbers

Steps:

```
Step 1 : Set f1 = FileOpen("data.txt", Write)
          // FileOpen is a procedure that opens the argument file in the
          // specified mode
Step 2 : Repeat through 2.a to 2.b for i = 0 to n
          // n stores the total count of numbers that is to be stored
          a) Input number
          b) WriteNumber(number, "data.txt")
              // WriteNumber is a procedure that writes the specified number
              // into the argument style
          [ End Of If Structure ]
Step 3 : FileClose(f1)
          // FileClose is a procedure that releases all open handles
          // to the given file
Step 4 : Set f1 = FileOpen("data.txt", Read)
Step 5 : Set f2 = FileOpen("data_even.txt", Write)
Step 6 : Set f3 = FileOpen("data_odd.txt", Write)
Step 7 : Repeat through step 7.a to 7.b While f1 is not exhausted
          a) Set t = ReadNumber(f1)
              // ReadNumber is a procedure that reads one number
              // from the input file
          b) If (t mod 2 = 0 )
              Then
                  1. WriteNumber(t, f2)
          c) Else
                  1. WriteNumber(t, f3)
          [ End Of If Structure ]
Step 8 : FileClose(f1)
Step 9 : FileClose(f2)
Step 10 : FileClose(f3)
```

Source Code

```
#include <stdio.h>
int main(){
    FILE *f1, *f2, *f3;
    int t, i, n;
    printf("\nHow many numbers you want to store : ");
    scanf("%d", &n);
    printf("\nContent of data file : ");
    f1 = fopen("data.txt", "w");
    for(i = 0; i < n; i++){
        printf("\nEnter number : ");
        scanf("%d", &t);
        if(t == -1){
            break;
        }
        else{
            fprintf(f1, "%d ", t);
        }
    }
    fclose(f1);
    f1 = fopen("data.txt", "rb");
    f2 = fopen("data_even.txt", "w");
    f3 = fopen("data_odd.txt", "w");
    while(fscanf(f1, "%d", &t) != EOF){ // read from data file
        if(t % 2 == 0)
            fprintf(f2, "%d ", t); // write to even file
        else
            fprintf(f3, "%d ", t); // write to odd file
    }
    fcloseall();
    f2 = fopen("data_even.txt", "rb");
    f3 = fopen("data_odd.txt", "rb");
    printf("\nContents of odd file : \n");
    while(fscanf(f3, "%d", &t) != EOF)
        printf("%d ", t);
    printf("\nContents of even file : \n");
    while(fscanf(f2, "%d", &t) != EOF)
        printf("%d ", t);
    return 0;
}
```

Input and Output

Set 1:

How many numbers you want to store : 4

Content of data file :

Enter number : 27

Enter number : 12

Enter number : 33

Enter number : 46

Contents of odd file :

27 33

Contents of even file :

12 46

Set 2:

How many numbers you want to store : 4

Content of data file :

Enter number : 372

Enter number : 291

Enter number : 473

Enter number : 861

Contents of odd file :

291 473 861

Contents of even file :

372

Discussion

1. Here one disadvantage is that there is no such checking whether the data file consist of some numbers or not.
2. Even if we do not provide any number, the program creates two seperated files(data_even.txt and data_odd.txt) which is not necessary. If this control would have been done in our program to create no file when the user do not provide any number then this unnecessary creation of data_even.txt and data_odd.txt in disk could be terminated.
3. There might not be enough space in the target disk to write the files. In that case, errornous output will be received.