

## SOS Assignment 1 – Task 1

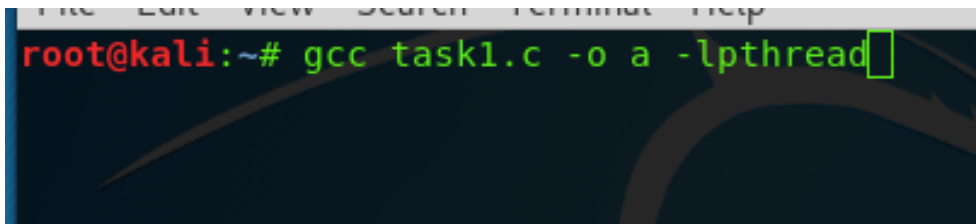
IT19001180

Rodrigo K. A. M.

### File name: task1.c

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

Figure 1



```
root@kali:~# gcc task1.c -o a -lpthread
```

Figure 2

When we use threads in our program, we must include the header file `#include <pthread.h>` on the top of the program [Figure 1]. And, when we compile the program, we must use the parameter `-lpthread` at the end of the line [Figure 2].

```
#include <stdio.h>

void * writeData (void * value) //writing data in a file
{
    FILE * cfPtr;    //creating a file pointer
    cfPtr = fopen ("dataset.txt", "w");    //creating the dataset.txt in write mode

    if (cfPtr == NULL){    //check whether the file can be created or not
        printf ("Cannot create file\n");
    }

    int dSet [10];    //creating an array to store the values in the text file
    int flag = 0;    //flag variable

    for (int r = 0; r < 10; r++){
        printf ("Enter number %d: ", r + 1);
        scanf ("%d", &dSet[r]);    //data writing

        fprintf (cfPtr,"%d\n",dSet[r]);    //printing the values in the dataset.txt

        if (dSet[r] != NULL){    //flag
            flag = 1;
            if (flag == 1){
                printf ("Line %d added\n", r + 1);
            }
        }
    }
}
```

Figure 3

In the first function, writeData dataset.txt file is created and getting user inputs with the help of a loop and stored them in the text file. Also, there is added a flag to let the program know that a certain condition has met. It is usually a Boolean value. But in here I used it as a sentence to easy to recognize [Figure 3].

```
void * calculateData (void * value)
{
    FILE * cfPtr;
    cfPtr = fopen ("dataset.txt", "r"); //opening dataset.txt file in read mode

    FILE * cfPtr1;
    cfPtr1 = fopen ("average.txt", "w"); //creating average.txt file to store the average of the values in dataset.txt

    int dSet [10]; //creating an array to retrieve data from dataset.txt
    int tot = 0; //creating a variable to get the total
    int avg; //creating a variable for the average

    for (int r = 0; r < 10; r++){
        fscanf (cfPtr, "%d", &dSet[r]); //retrieving values from dataset.txt
        tot = tot + dSet[r]; //calculating the total of those values
        printf ("%d\n", dSet[r]); //printing the values on the screen
    }
    avg = tot / 10; //calculating the average

    fprintf (cfPtr1, "%d", avg); //storing the average in average.txt
    printf ("Average is %d and it is written in 'average.txt' file\n", avg); //printing the average on the screen

    fclose (cfPtr); //closing dataset.txt
    fclose (cfPtr1); //closing average.txt
    return NULL;
}
```

Figure 4

In the second function, calculateData dataset.txt file is opened, and a new file called average.txt is created. In here the values stored in dataset.txt are read and from that values calculate the total and get the average of that and stored the average in the average.txt file [Figure 4].

```
int main (void)
{
    pthread_t thread1;
    pthread_create(&thread1, NULL, writeData, NULL);
    pthread_join(thread1, NULL);

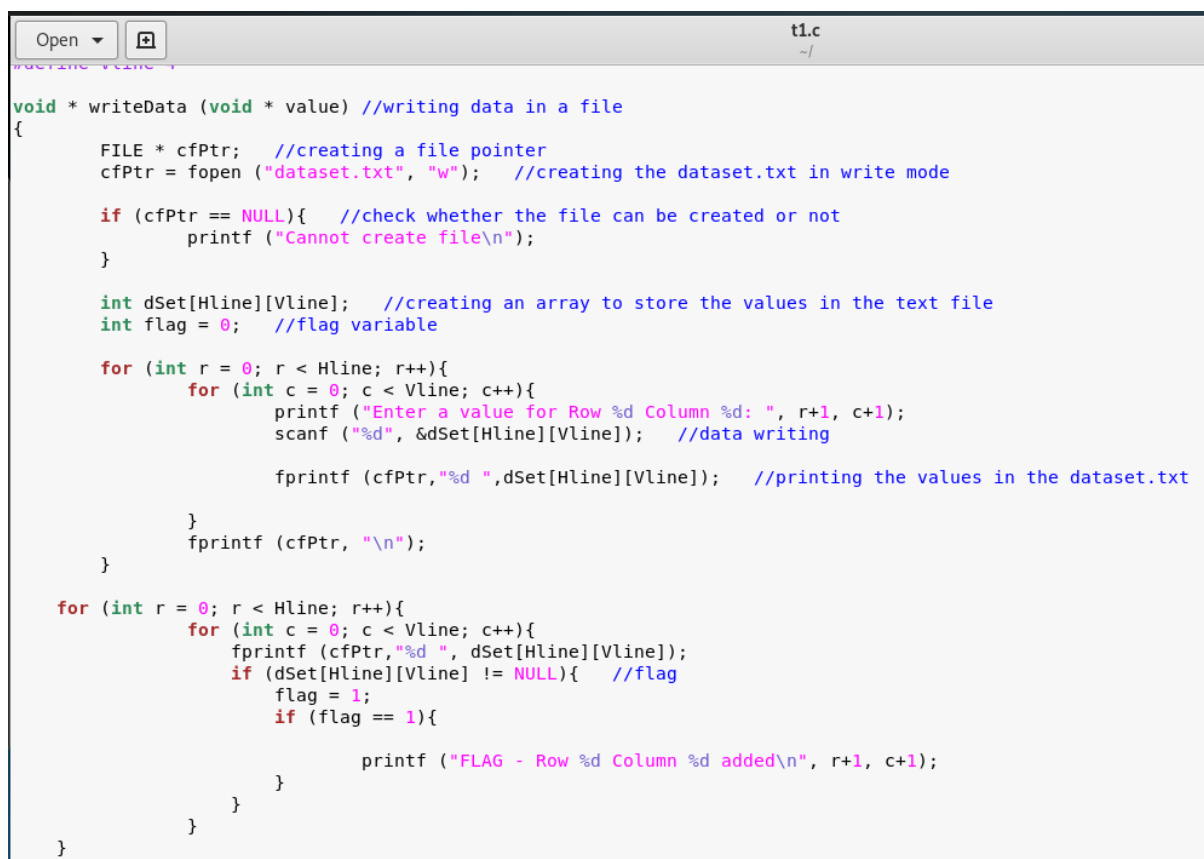
    pthread_t thread2;
    pthread_create(&thread2, NULL, calculateData, NULL);
    pthread_join(thread2, NULL);
}
```

Figure 5

Finally, in the main function, two thread are created, and the two functions above mentioned are joined [Figure 5].

### **Another program (t1.c)**

I have tried another way to implement this program. In here I used a 2D array to store data [Figure 6, 7].

The image shows a screenshot of a code editor window titled 't1.c'. The editor contains C code for a function 'writeData' and a 'main' function. The 'writeData' function takes a 'value' pointer and writes data to a file named 'dataset.txt'. It uses a 2D array 'dSet' to store data. The 'main' function calls 'writeData' and prints a message when the data is added. The code is as follows:

```
#define Hline 10
#define Vline 10

void * writeData (void * value) //writing data in a file
{
    FILE * cfPtr; //creating a file pointer
    cfPtr = fopen ("dataset.txt", "w"); //creating the dataset.txt in write mode

    if (cfPtr == NULL){ //check whether the file can be created or not
        printf ("Cannot create file\n");
    }

    int dSet[Hline][Vline]; //creating an array to store the values in the text file
    int flag = 0; //flag variable

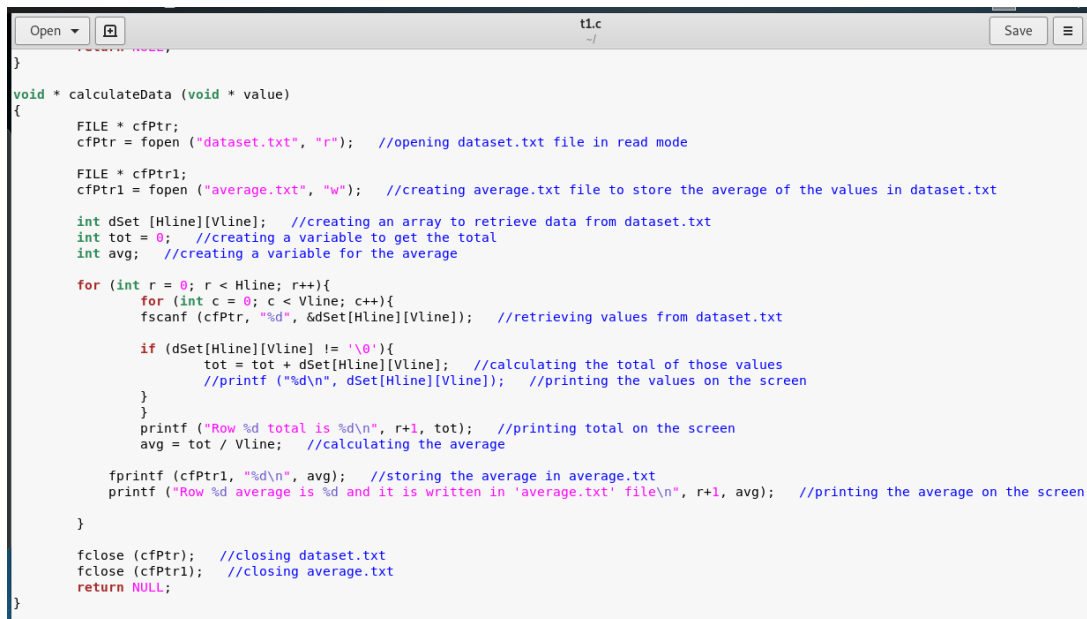
    for (int r = 0; r < Hline; r++){
        for (int c = 0; c < Vline; c++){
            printf ("Enter a value for Row %d Column %d: ", r+1, c+1);
            scanf ("%d", &dSet[Hline][Vline]); //data writing

            fprintf (cfPtr, "%d ", dSet[Hline][Vline]); //printing the values in the dataset.txt

        }
        fprintf (cfPtr, "\n");
    }

    for (int r = 0; r < Hline; r++){
        for (int c = 0; c < Vline; c++){
            fprintf (cfPtr, "%d ", dSet[Hline][Vline]);
            if (dSet[Hline][Vline] != NULL){ //flag
                flag = 1;
                if (flag == 1){
                    printf ("FLAG - Row %d Column %d added\n", r+1, c+1);
                }
            }
        }
    }
}
```

Figure 6



```
void * calculateData (void * value)
{
    FILE * cfPtr;
    cfPtr = fopen ("dataset.txt", "r");    //opening dataset.txt file in read mode

    FILE * cfPtr1;
    cfPtr1 = fopen ("average.txt", "w");    //creating average.txt file to store the average of the values in dataset.txt

    int dSet [Hline][Vline];    //creating an array to retrieve data from dataset.txt
    int tot = 0;    //creating a variable to get the total
    int avg;    //creating a variable for the average

    for (int r = 0; r < Hline; r++){
        for (int c = 0; c < Vline; c++){
            fscanf (cfPtr, "%d", &dSet[Hline][Vline]);    //retrieving values from dataset.txt

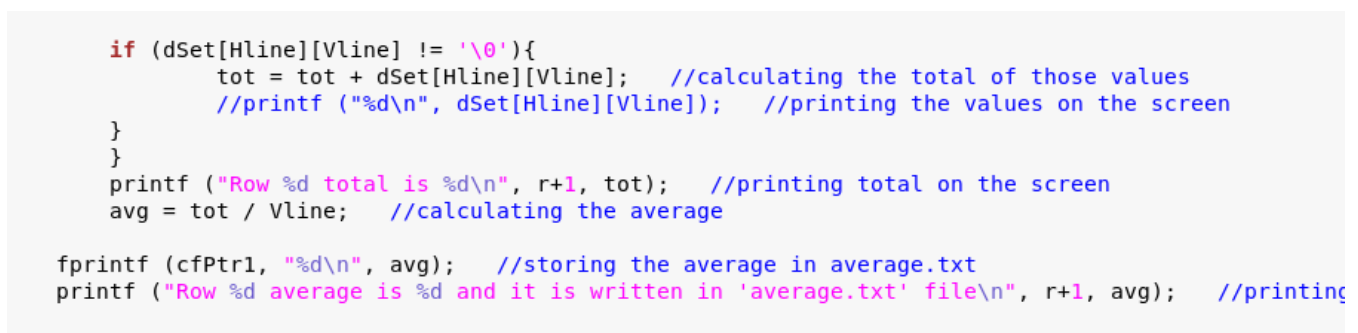
            if (dSet[Hline][Vline] != '\0'){
                tot = tot + dSet[Hline][Vline];    //calculating the total of those values
                printf ("%d\n", dSet[Hline][Vline]);    //printing the values on the screen
            }
        }
        printf ("Row %d total is %d\n", r+1, tot);    //printing total on the screen
        avg = tot / Vline;    //calculating the average

        fprintf (cfPtr1, "%d\n", avg);    //storing the average in average.txt
        printf ("Row %d average is %d and it is written in 'average.txt' file\n", r+1, avg);    //printing the average on the screen
    }

    fclose (cfPtr);    //closing dataset.txt
    fclose (cfPtr1);    //closing average.txt
    return NULL;
}
```

Figure 7

To calculate the total of a line I used '\0'. It is used to indicate the end of the line [Figure 8].



```
if (dSet[Hline][Vline] != '\0'){
    tot = tot + dSet[Hline][Vline];    //calculating the total of those values
    printf ("%d\n", dSet[Hline][Vline]);    //printing the values on the screen
}
}
printf ("Row %d total is %d\n", r+1, tot);    //printing total on the screen
avg = tot / Vline;    //calculating the average

fprintf (cfPtr1, "%d\n", avg);    //storing the average in average.txt
printf ("Row %d average is %d and it is written in 'average.txt' file\n", r+1, avg);    //printing
```

Figure 8

## REFERENCES

<https://www.daniweb.com/programming/software-development/threads/201233/use-of-flag-in-c>

<https://www.geeksforgeeks.org/use-of-flag-in-programming/>

<https://www.youtube.com/watch?v=nVESQQg-Oiw>

<https://www.youtube.com/watch?v=uA8X5zNOGw8>