

LAB EXERCISE 11

Threading Applications

Submission Date: 23-05-2022

Name: Jayannthan P T

Dept: CSE 'A'

Roll No.: 205001049

1. Write a multithreaded program that calculates various statistical values for a list of numbers. This program will be passed a series of numbers on the command line and will then create three separate worker threads. One thread will determine the average of the numbers, the second will determine the maximum value, and the third will determine the minimum value. For example, suppose your program is passed the integers

90 81 78 95 79 72 85

The program will report

The average value is 82

The minimum value is 72

The maximum value is 95

Algorithm:

1. Initialize global variables sum, min, max, avg
2. Create functions to calculate avg, min, max
3. Inside the main function, create thread identifiers and set of thread attributes
4. Read the argument from the user as command line arguments and have it in the array
5. Get default attributes and create thread for each function
6. Wait for thread to exit before each creation
7. Using join function, close the existing executed thread to compile and link it
8. Print the data obtained from the thread

Code:

```
#include <stdio.h>
#include <pthread.h>
int arr[50], n, i;

void *calc_avg()
{
```

```

float sum = 0;
float average;
/*printf("enter size:");
scanf("%d", &n);
printf("\nenter array elements:\n");
for (i = 0; i < n; i++)
{
    scanf("%d", &arr[i]);
}*/
for (i = 0; i < n; i++)
{
    sum = sum + arr[i];
}
average = sum / n;
printf("The average value is:%.2f", average);
}

void *calc_min()
{
    int temp = arr[0];
    for (int i = 1; i < n; i++)
    {
        if (temp > arr[i])
        {
            temp = arr[i];
        }
    }
    printf("\nThe Minimum value is:=%d", temp);
}

void *calc_max()
{
    int temp = arr[0];
    for (int i = 1; i < n; i++)
    {
        if (temp < arr[i])
        {
            temp = arr[i];
        }
    }
    printf("\nThe Maximum value is:=%d\n", temp);
}

int main(int argc, char* argv[])
{
    int m;
    n=argc-1;
    for (i = 0; i < n; i++)
    {
        arr[i]=atoi(argv[i+1]);
    }
    pthread_t t;
    m = pthread_create(&t, NULL, &calc_avg, NULL);
    pthread_join(t, NULL);
    m = pthread_create(&t, NULL, &calc_min, NULL);

```

```
pthread_join(t, NULL);  
m = pthread_create(&t, NULL, &calc_max, NULL);  
pthread_join(t, NULL);  
}
```

Output:

```
root@spl25:~/Jayannthan/Threads# ./threads 1 2 3 4 5  
The average value is:3.00  
The Minimum value is:=1  
The Maximum value is:=5
```

Learning Outcome:

- Implemented threading application
- Learnt the importance of threading