



National Textile University

Department of Computer Science

Subject: Operating System

Submitted to: Sir Nasir

Submitted by: Maha

Reg. number: 23-NTU-CS-1170

MIDs Eveluation

Semester:5th

Question:

Write a C program that calculates array sum using multiple threads:

1. Create an array of 20 integers: {1, 2, 3, ..., 20}
2. Create 4 threads, each calculating sum of 5 elements: Thread 1: elements 0-4 (sum of 1 to 5)
Thread 2: elements 5-9 (sum of 6 to 10)
Thread 3: elements 10-14 (sum of 11 to 15)
Thread 4: elements 15-19 (sum of 16 to 20)

Pass the starting index and count to each thread using a structure:

4. Each thread prints its partial sum
5. Main thread collects all return values and calculates total sum

Code:

```
// Write a C program that calculates array sum using multiple threads:
// 1. Create an array of 20 integers: {1, 2, 3, ..., 20}
// 2. Create 4 threads, each calculating sum of 5 elements: Thread 1: elements
0-4 (sum of 1 to 5)
// Thread 2: elements 5-9 (sum of 6 to 10)
// Thread 3: elements 10-14 (sum of 11 to 15)
// Thread 4: elements 15-19 (sum of 16 to 20)
// Pass the starting index and count to each thread using a structure:
// 4. Each thread prints its partial sum
// 5. Main thread collects all return values and calculates total sum

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

//global variables
#define arraysize 20
#define size 20
#define threads 4
#define numthreads 4

int arr[size];

typedef struct{
```

```

    int start;
    int count;
} Sumdata;

void* sumIndexing(void* args){
    Sumdata* data=(Sumdata*)args;
    int start=data->start;
    int count=data->count;

    int* halfsum=malloc(sizeof(int));
    *halfsum=0;

    for(int i=start;i < start + count;i++){
        *halfsum+=arr[i];
    }
    printf("thread %d to %d sum is: %d \n",start,start+count,*halfsum);
    pthread_exit(halfsum);
}

int main(){
    for(int i=0;i<size;i++){
        arr[i]=i+1;
    }
    pthread_t threadArray[threads];
    Sumdata data[threads];

    int part=size/numthreads;

    for(int i=0;i<numthreads;i++){
        data[i].start=i*part;
        data[i].count=part;

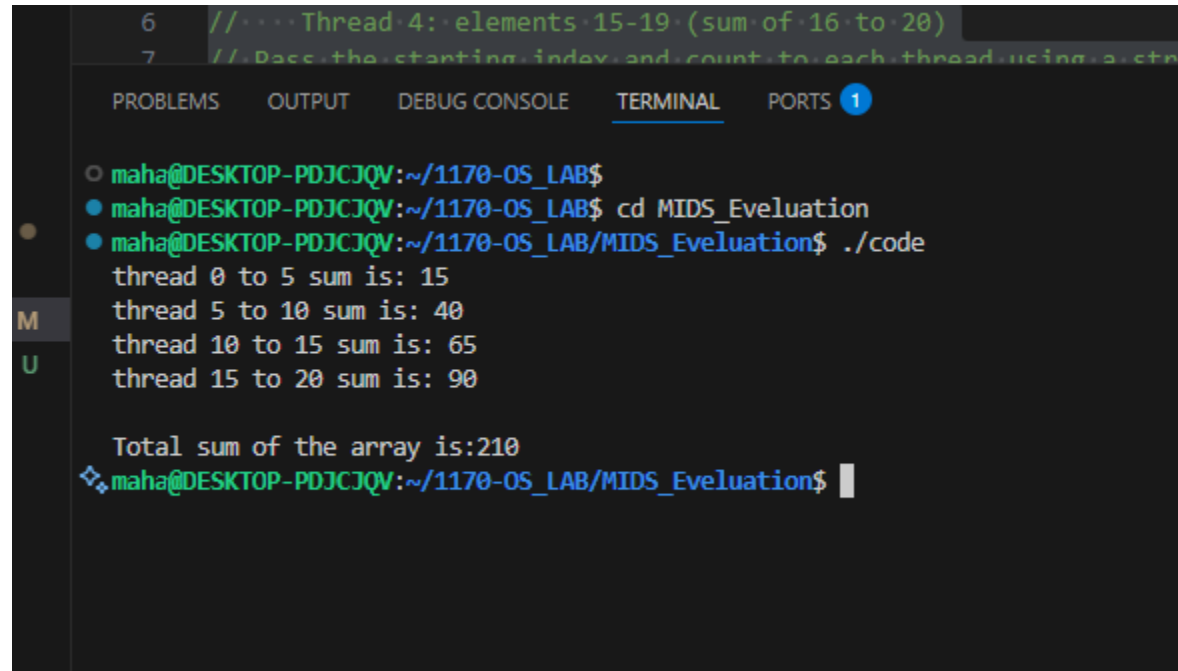
        if(pthread_create(&threadArray[i],NULL,sumIndexing,&data[i])!=0){
            printf("error occurs");
            exit(1);
        }
    }
    int total_sum=0;

    for(int i=0;i<threads;i++){
        int* result;
        pthread_join(threadArray[i],(void**)&result);
        total_sum+=*result;
        free(result);
    }
    printf("\nTotal sum of the array is:%d\n",total_sum);
}

```

```
    return 0;
}
```

Terminal:



```
6 //...Thread 4: elements 15-19 (sum of 16 to 20)
7 // Pass the starting index and count to each thread using a str

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS 1

maha@DESKTOP-PDJCJQV:~/1170-OS_LAB$
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB$ cd MIDS_Eveluation
maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/MIDS_Eveluation$ ./code
thread 0 to 5 sum is: 15
thread 5 to 10 sum is: 40
thread 10 to 15 sum is: 65
thread 15 to 20 sum is: 90

Total sum of the array is:210
❖maha@DESKTOP-PDJCJQV:~/1170-OS_LAB/MIDS_Eveluation$
```

Remarks:

- create four threads to calculate the sum of different parts of the array.
- crete data structure to pass arguments to threads.(start,count)
- create a function to allocate different size of array
- allocate different size of array to each thread
- use malloc() to allocate memory space for each thread's sum value
- each thread create a partial sum
- use pthread_join() to collect all the return values from each thread