

CSE 3033 - OPERATING SYSTEMS

Programming Assignment # 3 **DUE DATE: 04/01/2024 - 23:59**

In this assignment, you will write a program that uses threads and synchronization to find the sum of the square root of numbers in a range given by the user. An example serial code is as follows:

Program Details

- The program will be executed specifying four input integers: a b c d
- a is a **long long int** variable that is the start of the range.
- b is a **long long int** variable that is the end of the range.
- c is a **int** variable that is the thread number.
- d is a **int** variable that indicates one of the three methods that will be used.
- The output of the program will be the sum of the square roots of the numbers in the range a-b. i.e.

```
double global_sqrt_sum = 0;
long long int x;
for (x = a; x <= b; x++){
    global_sum += sqrt(x);
}
```
- The range (b-a) will be divided to threads almost equally.
- In all three methods, the threads will update a variable of type **double** named **global_sqrt_sum**.
- The details of the methods are as the following:
 - Method 1 (d = 1): All threads update **global_sqrt_sum** concurrently without any critical sections. No mutex variables will be used.
 - Method 2 (d = 2): All threads update **global_sqrt_sum**, but this time serially. A single shared mutex variable will be used to protect the **global_sqrt_sum** variable updates.
 - Method 3 (d = 3): Each thread will use a local variable of type **double** called **local_sqrt_sum** and calculate the sum of the square root of numbers to **local_sqrt_sum**. After the local sums are calculated, each thread will add its local sum to **global_sqrt_sum**. A single shared mutex variable will be used to protect the **global_sqrt_sum** variable updates.

- Your program will be executed with Linux **time** command to get the user, system and total running time:

```
time ./project3.out 880130203012 922823372203 1
```

```
Sum: 4.05346e+16
```

```
./ project3.out 40.13s user 0.01s system 98% cpu 40.720 total
```

Note that **user** indicates the time spent in user mode, **system** indicates the time spent in kernel mode, and **total** indicates the total (turnaround) time for the process.

You should execute your program **prepare a table containing**

- The sum value found,
- User time of the process,
- System time of the process,
- Total time of the process;

for each of the following input configurations:

Method 1 Input:

a = 880130203012

b = 922823372203

c = 1, 2, 4, 8

d = 1

Method 2 Input:

a = 880130203012

b = 922823372203

c = 1, 2, 4, 8, 16, 32

d = 2

Method 3 Input:

a = 880130203012

b = 922823372203

c = 1, 2, 4, 8, 16, 32

d = 3

In addition to the three tables you provided, answer the following questions:

1. Which method(s) provide the correct result and why?
2. Among the method(s) providing the correct result, which method is the fastest?
3. Among the method(s) providing the correct result, does increasing the number of threads always result in smaller total time? Discuss this considering the number of CPU cores available in your computer (in Linux, `lscpu` command provides the number of CPU cores available in your computer).
4. Are there any differences in user time/system time ratio of the processes as the number of threads increases? What could be the cause of these differences?

Notes:

- The project will be done in Linux operating system using C programming language.
- You must use PThread library and synchronization appropriately in your code.
- Consider materials and examples covered in the lab sessions.
- Consider all necessary error checking for the programs.
- No late homework will be accepted!
- In case of any form of copying and cheating on solutions, all parties/groups will get ZERO grade. You should submit your own work.
- You must work in groups of three.

What to submit?

A softcopy of your source codes which are **EXTENSIVELY** commented and appropriately structured and a project report (minimum 2-page) that contains detailed information about your implementation and the run results should be emailed to cse333.projects@gmail.com. All the files should be submitted as one zip file. You should use your student numbers as the name of the file: student#1_student#2_student#3_project3.zip