

**Lab 4 Report**

**Report Subject: OS Experiment - Lab 4**

**Student ID: 2018380039**

**Student Name: Dikshya Kafle**

**Data：2021/11/22**

**Computer Operating System Experiment**

**Laboratory 4**

**Threads**

1. **Objective:**

* Practice working with multi-threaded C programs.
* Practice working with C function pointers.

1. **Equipment:**

* VirtualBox with Ubuntu Linux

1. **Experiments:**

**Experiment 1: data sharing**

Questions:

1. What would be the output from the program at LINE A and LINE B? Please use the data sharing mechanism to explain it.

Answer:

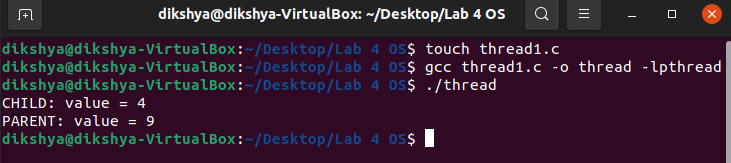
* LINE A: CHILD: value = 4
* LINE B: PARENT: value = 9

When we fork() a new process, the parent and child process are both separate processes and each one of them has its copy of the data. So editing the data in one process does not affect the variables of the other process. However, threads that belong to the same process share the same data. So within the same process, if a thread edits a variables, this change is visible to all the other threads.

In our case, myvalue=10 in the parent process and myvalue=5 in the child process (Two non-shared separate copies of data). Each thread performs value = myvalue -1 (value is also no shared between the 2 processes and each thread only edits the variable belonging to its process). As a result we get the following output:

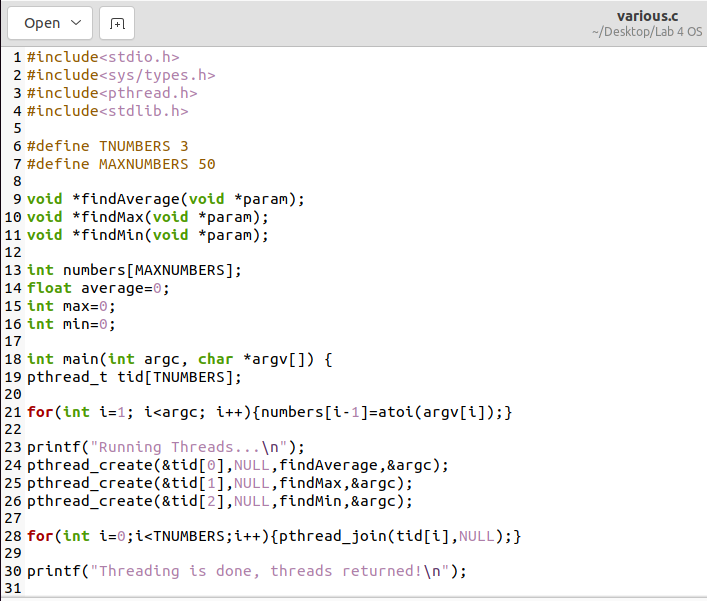
*CHILD: value = 4*

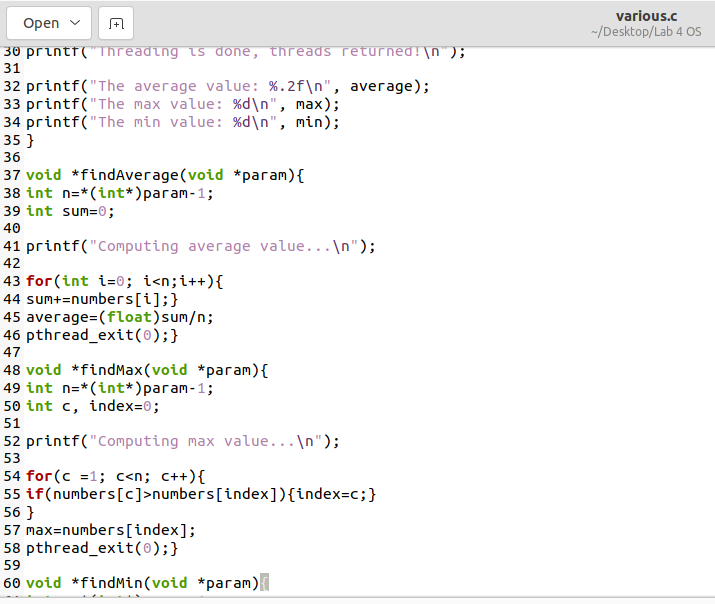
*PARENT: value = 9*

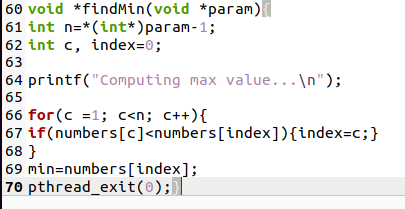


**Experiment 2: calculates various statistical values**

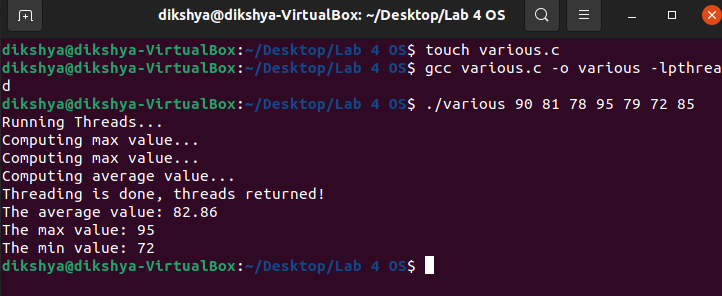
* The code:







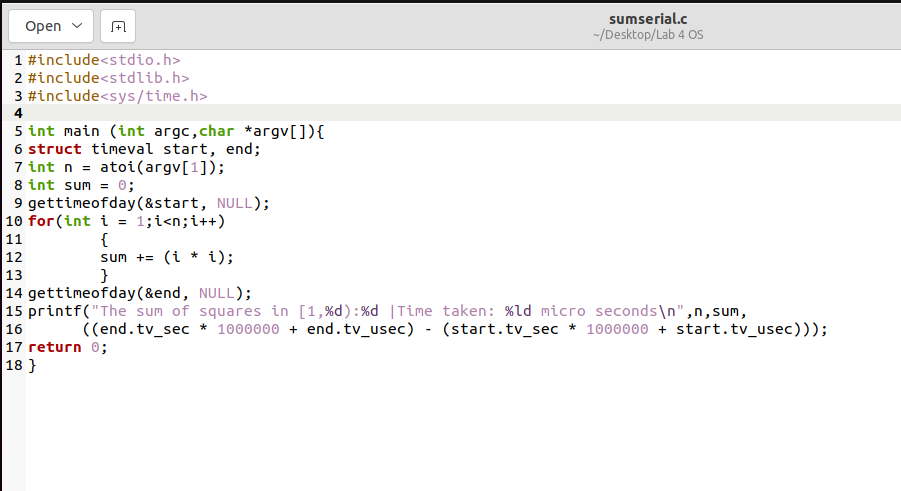
* The result:



The multithreaded program is passed a series of numbers on the command line and will then create three separate worker threads. One thread determines the average of the numbers, the second determines the maximum value, and the third determines the minimum value.

**Experiment 3: calculate the sum of number of squares**

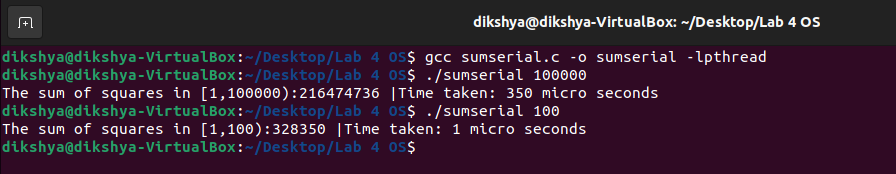
* The code:



* The result:

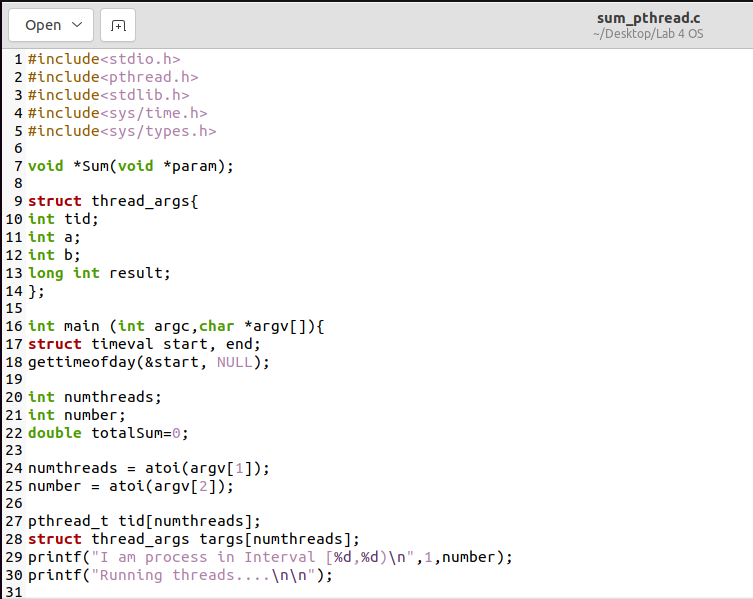
I use the range [1,10000) which sums the squares in integers from 1 to 9999. I take time stamp before and after the look execution. The results are as follow:

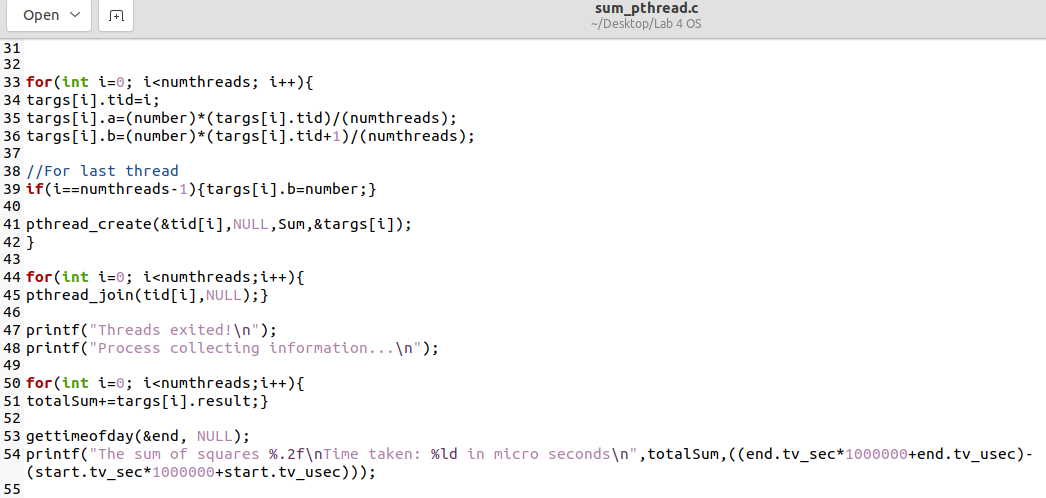


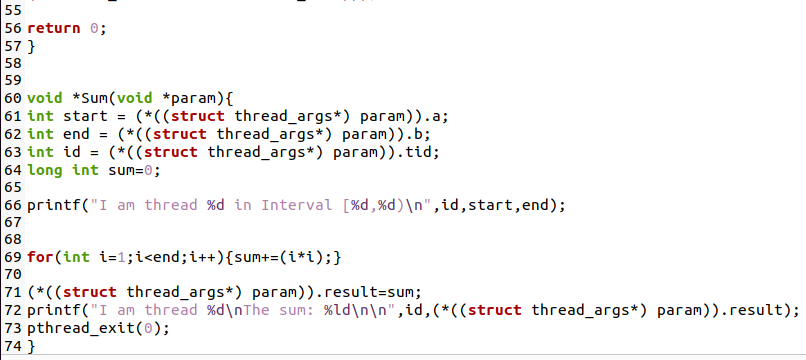


**Experiment 4: calculate the sum of number of squares using Pthread**

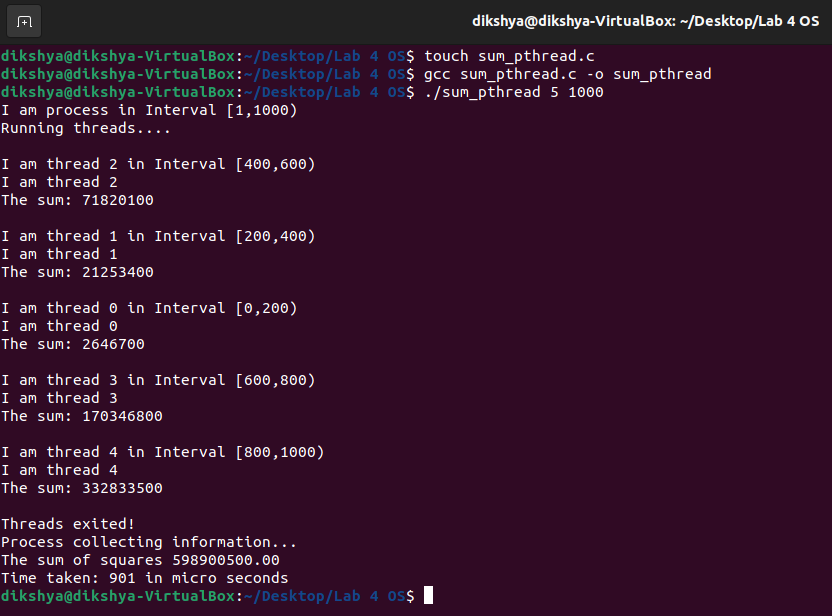
* The code:



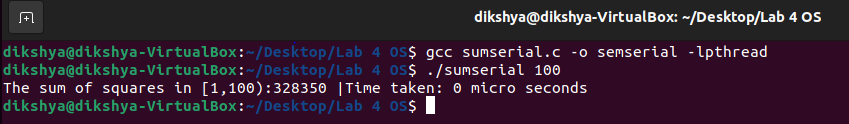




* The Result:



Comparing with the serial implementation:



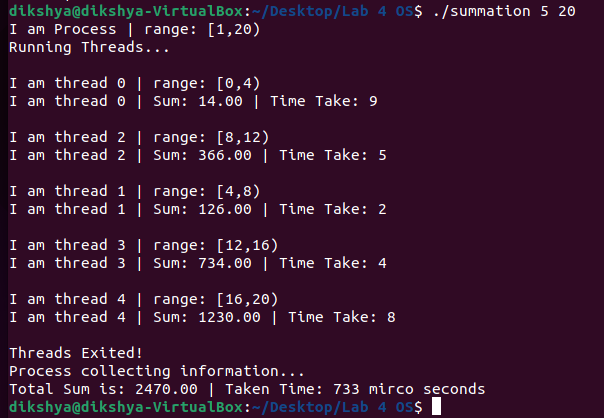
**Experiment 5: analysis the performance**

* The code:





* The results:



* Questions:
* When you increase the number of threads to get the summation, whether the running time is shorter or not? Why?

Answer:

The running time is longer whenever the number of threads increases. Since the division of the superset into subsets take a long time, hence I believe that multithreading is not a good option for the sum of squares, multithreading could be more efficient matrix multiplication and such.

* If there is any problem, explain what causes it and how you can avoid it. Also, remember to modify your source code and submit a version that sure prints out the termination time for each thread.

Answer: Ok