BLG 312E– Computer Operating Systems
2019–2020, Assignment 2

Introduction

You will code a multi-processing and multi-threading program that finds the prime numbers. The number interval for search (**interval_min** and **interval_max**), the number of processes (**np**) and the number of threads (**nt**) will be given as command line arguments. The number interval will be used as a search space to find prime numbers. You need to divide the number interval among **np** process, and each process should also divide them among **nt** threads. Threads will be responsible for finding the prime numbers in the assigned range. Upon completion of threads, processes should relay the found prime numbers to the main process through a shared memory. Finally, the main process prints the prime numbers that are calculated in ascending order. You will only submit a single source code file through Ninova.

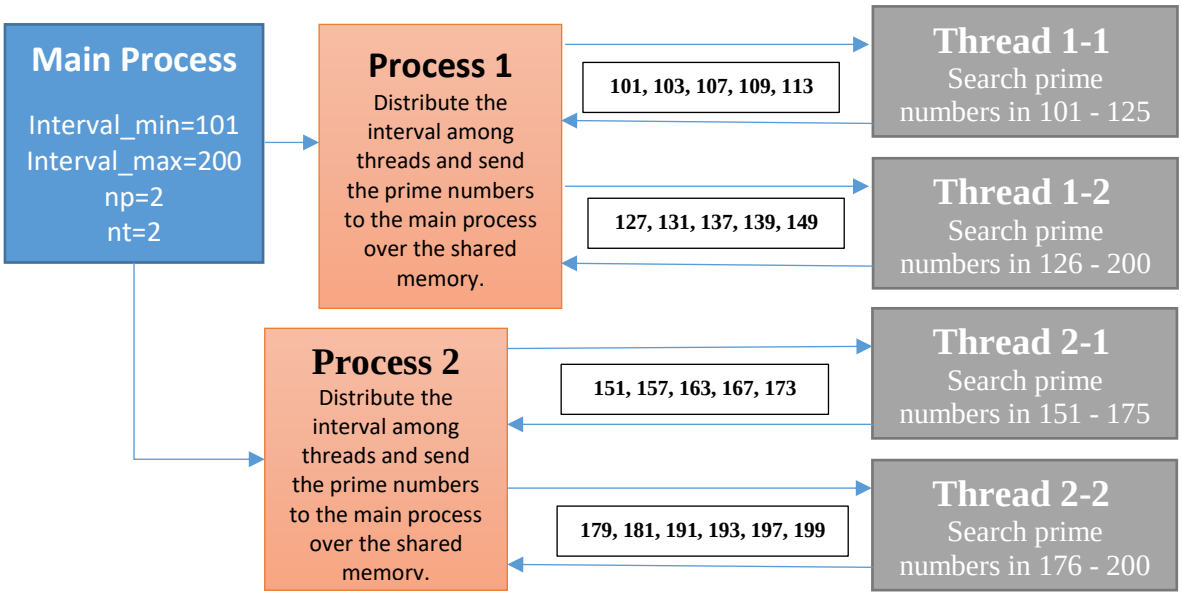
Program Input

Program inputs are the number interval (interval_min, interval_max), the number of processes(**np**) and the number of threads(**nt**). These inputs should be received as a command line argument. To evaluate your programs, multiple options will be used.

./yourprogram interval_min interval_max np nt

Example command: **./yourprogram 101 200 2 2**

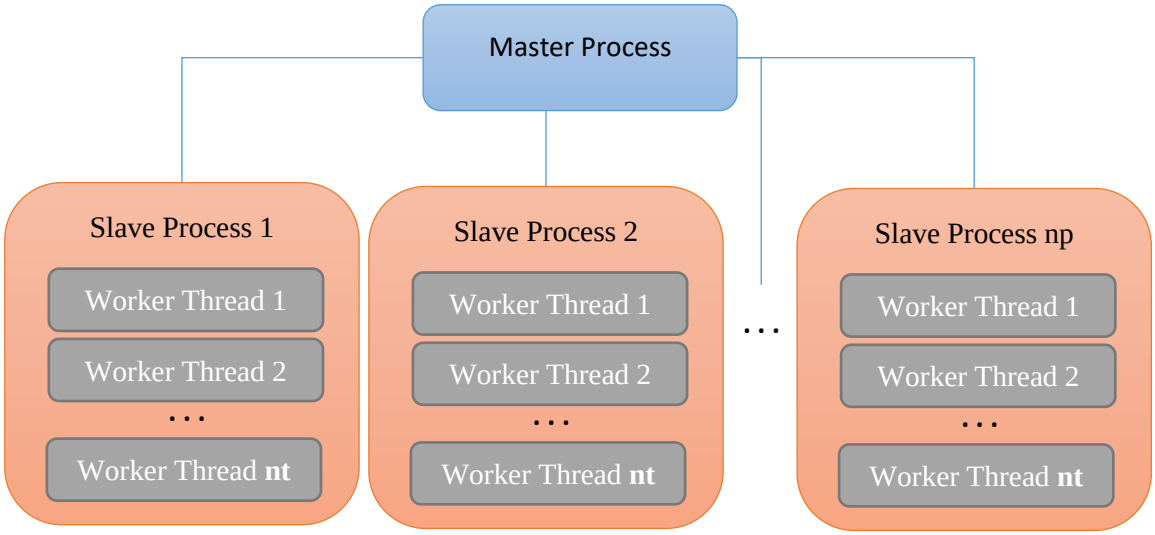
It will create **2** processes and each process will create **2** threads.
Process **1** will be responsible for the numbers between **101-150**
  Thread 1-**1** will check for primality in the range of **101-125**
  Thread 1-**2** will check for primality in the range of **125-150**
Process **2** will be responsible for the numbers between **151-200**
  Thread 2-**1** will check for primality in the range of **151-175**
  Thread 2-**2** will check for primality in the range of **176-200**



Processes and Threads

Your program should have the following functionality for the processes and threads.

- **Master Process:** Master process should receive arguments as a command line argument. Then, it should create **np** slave processes. It will also divide the number interval into the number of processes, and distribute them among the slave processes. (E.g. If the initial range is 1-10 and the number of slave processes is 2, the processes will be assigned to the ranges of 1-5 and 6-10) Upon completion of slave processes, the main process must print the prime numbers **that are sent over the shared memory by the processes** in ascending order.
- **Slave(Child) Processes:** Each slave process should create **nt** worker threads and assign a number range to each thread. Upon completion of its threads, the slave process must send the prime numbers **that are calculated by the worker threads** to the main process over a shared memory.
- **Worker Threads:** Each worker thread is responsible for checking primality of numbers in the defined range. You can use a simple divisibility test for the checking primality.
- **Each processes and threads should print a starting and an ending message as seen in the sample output.**
- **The slave processes and the worker threads must be able to work in parallel.**



Include development environment information, compilation and running commands as a comment in your code.

EXAMPLE:

```
// Development environment: Lubuntu 16.04 or ITU SSH servers
// To compile: g++ -c Homework1.cpp –pthread
// To run: ./a.out interval_min interval_max np nt
// Example: ./a.out 101 200 2 2
```

Sample Program Output

A sample program output is given below. Your program must print similar information to the screen. Please note that, the order of lines may be different at each run.

```
./yourprogram 101 200 2 2

Master: Started.
Slave 2: Started. Interval 151-200
Thread 2.2: searching in 176-200
Slave 1: Started. Interval 101-150
Thread 1.2: searching in 126-150
Thread 1.1: searching in 101-125
Thread 2.1: searching in 151-175
Slave 2: Done.
Slave 1: Done.
Master: Done. Prime numbers are: 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199,
```

Important notes:

- You are expected to work individually on this homework. All forms of collaboration are discouraged and will be treated as plagiarism. This includes actions such as, but not limited to, submitting the work of others as one's own (even if in part and even with modifications) and copy/pasting from other resources (including Internet resources) even with proper reference. Such offenses are reported to the administration for disciplinary measures. All parties involved in the act will be treated equally.
- You should submit your homework (source code) through Ninova system. Late submissions are not accepted.
- Your code must be written in C, and should be compiled and run on ITU's Linux Server (you can access it through SSH) using gcc. Your code must compile without any errors; otherwise, you may get a grade of zero on the assignment.
- You should **include necessary instructions for compiling and running** your program as a comment.
- If you have further questions about the assignment, you may contact the course assistant Çağatay KOÇ(kocca@itu.edu.tr)

Appendix

You may make use of the following code examples for reading command-line arguments and converting strings to numbers in C and C++.

- https://rosettacode.org/wiki/Command-line_arguments
- https://www.geeksforgeeks.org/converting-strings-numbers-cc/