CS 3305A: Operating Systems
Department of Computer Science
Western University
Assignment 3
Fall 2022
Due Date: October 25, 2022

Purpose

The goals of this assignment are the following:

- Get experience with the *pipe and pthread* system functions.
- Learn how to create multiple threads for different tasks.
- Learn how to share data between threads using the pipe.
- Gain more experience with the C programming language from an OS perspective.

Inter-Thread Communications (100 points)

Write a C program that will accept two integers from the user as **command-line arguments** (for example, X and Y where X ,Y are positive integers & X>Y). The *parent* process will read X and Y from the command line. The *parent* process will create three threads (i.e., pthread_t 1, pthread_t 2, and pthread_t 3). The parent process will write X and Y to the shared memory using pipe. The first thread (i.e., 1) will read X and Y from the pipe and perform the subtract, S = X-Y, and then the result S will be written to the pipe. Next, the second thread (i.e., 2) will read S from the pipe and determine whether S is a prime number, **and then S will be written again to the pipe by the second thread.** Finally, the third thread (i.e., 3) will read S from the pipe and reverse the number S. The expected output from your program should look like the following (for this example below, X and Y represent 33 and 4, respectively):

1. parent (PID 1729) receives X = 33 and Y = 4 from the user
2. parent(PID 1729) writes X = 33 and Y = 4 to the pipe
3. thread(TID 1) reads X = 33 and Y = 4 from the pipe
4. thread(TID 1) writes X-Y = 29 to the pipe
5. thread(TID 2) reads X-Y = 29 from the pipe
6. thread(TID 2) identified that 29 is a prime number
7. thread(TID 2) writes 29 to the pipe
8. thread(TID 3) reads X-Y = 29 from the pipe
9. thread(TID 3) reversed number is 92

In the above example, if S is NOT a prime number, then the phrase “identified that xx is a prime number” in line **number 6 above** must be replaced with the phrase “identified that xx is NOT a prime number”. You must control the execution of the threads to follow the sequence according to the expected output above. You must not use more than one pipe for this assignment. In case of passing multiple parameters through a single pipe, concatenate the parameters using any delimiter so that you can parse it later accordingly. This assignment will be tested given only positive integers where X > Y. Your implementation must have the following functions:

1. void *subtract(void *thread_id): This function is executed by thread 1. This function reads X and Y from the pipe, performs subtraction i.e., S = X-Y, and writes S to the pipe.

2. void *prime_check(void *thread_id): This function is executed by thread 2. This function reads S from the pipe and determines if S is a prime number or not.
3. void *reverse_num(void *thread_id): This function is executed by thread 3. This function reverses the number S.

Mark Distribution

This section describes a tentative allocation of marks assigned for the desired features.

- **Inter-Thread Communications (100 points)**
    a) Parent reads X and Y from user: 10 points
    b) The first thread reads X and Y from pipe: 15 points
    c) The first thread writes results to the pipe: 10 points
    d) The second thread reads the result from the pipe: 10
    e) The second thread identifies if it is a prime number: 15
    f) The third thread reads the result from the pipe: 10
    g) The third thread reverses the number: 15
    h) Control the thread execution flow: 15 points

You must pass the input to the program using the command line argument. The hardcoded input will not be accepted and considered for deducting marks accordingly. Also, marks will be deducted if error handling is not implemented in your code.

Computing Platform for Assignments

You are responsible for ensuring that your program compiles and runs without error on the computing platform mentioned below. **Marks will be deducted** if your program fails to compile or runs into errors on the specified computing platform (see below).

- Students have virtual access to the MC 244 lab, which contains 30 Fedora 28 systems. Linux machines available to you are **linux01.gaul.csd.uwo.ca** through **linux30.gaul.csd.uwo.ca**.
- It is your responsibility to ensure that your code compiles and runs on the above systems. You can SSH into MC 244 machines (please see the Assignment 1 file transfer tutorial).
- If you are off-campus, you have to SSH to **compute.gaul.csd.uwo.ca** first (this server is also known as sylvia.gaul.csd.uwo.ca, in honor of Dr. Sylvia Osborn), and then to one of the MC 244 systems **(linux01.gaul.csd.uwo.ca** through **linux30.gaul.csd.uwo.ca**) [please see the Assignment 1 file transfer tutorial].
- https://wiki.sci.uwo.ca/sts/computer-science/gaul

Assignment Submission

You need to submit only one C file. The name of your submitted C file must be "assignment3.c". Marks will be deducted if your submitted C file name is different. You must submit your assignment through OWL. Be sure to test your code on one of MC 244 systems (see "Computing Platform for Assignments" section above). **Marks will be deducted** if your program fails to compile or runs into errors on the computing platform mentioned above.

Assignment 3 FAQ will be made available on OWL as needed. Also, consult TAs and the Instructor for any questions you may have regarding this assignment.