

CS 3305A: Operating Systems
Department of Computer Science
Western University
Assignment 3
Fall 2021
Due Date: October 25, 2021

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). The *parent* process will read X and Y from the command line. The *parent* process will create three threads with corresponding thread ID 100, 101, and 102, respectively. The parent process will write X and Y to the shared memory using pipe. The first thread (i.e., thread 100) will read X and Y from the pipe and perform the sum, $S = X + Y$, and then the result S will be written to the pipe. Next, the second thread (i.e., thread 101) will read the S from the pipe and determine whether S is an even or odd number, and then S will be written again to the pipe by the second thread. Finally, the third thread (i.e., thread 102) will read S from the pipe and count the total number of digits in S. The expected output from your program should look like the following (for this example below, X and Y represent 7 and 6, respectively):

1. parent (PID 2209) receives $X = 7$ and $Y = 6$ from the user
2. parent (PID 2209) writes $X = 7$ and $Y = 6$ to the pipe
3. thread (TID 100) reads $X = 7$ and $Y = 6$ from the pipe
4. thread (TID 100) writes $X + Y = 13$ to the pipe
5. thread (TID 101) reads $X + Y = 13$ from the pipe
6. thread (TID 101) identifies $X + Y = 13$ as an odd number
7. thread (TID 101) writes $X + Y = 13$ to the pipe
8. thread (TID 102) reads $X + Y = 13$ from the pipe
9. thread (TID 102) identifies $X + Y = 13$ as a 2 digit number

In the above example, if the sum S is an even number, then the word “odd” in line number 6 above must be replaced with the word “even”. 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. Your implementation must have the following functions:

1. `void *sum(void *thread_id)`: This function is executed by thread 100. This function reads X and Y from the pipe, performs addition i.e., $S = X + Y$, and writes S to the pipe.

2. `void *odd_even(void *thread_id)`: This function is executed by thread 101. This function reads `S` from the pipe and determines if the `S` is an even or odd number.
3. `void *digit_count(void *thread_id)`: This function is executed by thread 102. This function determines the number of digits in `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 & Y* from user: 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 odd/even number: 15
 - f) The third thread reads the result from the pipe: 10
 - g) The third thread identifies the number of digits: 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.

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.