

CS 3305A: Operating Systems
Department of Computer Science
Western University
Assignment 5
Fall 2021
Due Date: Nov 29, 2021

Purpose

The goals of this assignment are the following:

- Get hands-on experience in developing mutual exclusion / semaphore / critical section techniques/algorithms.
- Gain more experience with the C programming language from an OS's mutual exclusion / semaphore / critical section perspective.

Assignment Description

Using C programming language, you will be developing a mutual exclusion algorithm for a process synchronization problem. You need to make sure that your mutual exclusion algorithm ensures that only one process can access the critical section portion of your code at a given point in time. You are allowed to use any mutual exclusion / semaphore related C function calls.

a) Description of the problem is given below:

Assume that there are a set of n bank accounts ($n \geq 1$) shared by a set of x clients ($x \geq 1$). Clients can perform two different types of transactions with each bank account: deposit and withdraw funds. If a particular transaction results in a negative account balance, that transaction must not take place (i.e., an account balance should never be less than 0).

b) Structure of the input file:

In the following example, there are two bank accounts (a1 and a2) shared by a total of ten clients (c1 to c10). The clients are allowed to deposit money into both accounts and withdraw money from both accounts. The initial balances of the accounts are specified in the input file. An input file is provided below for illustrative purposes.

```
a1 b 7000
a2 b 4500
c1 d a1 100 w a2 500
c2 w a1 2500
...
...
c9 w a1 1000 w a2 500
c10 d a1 50 d a2 200
```

Illustration:

(i) a1 b 7000

The above line specifies the initial balance of account #1 as \$7000

(ii) c1 d a1 100 w a2 500

The above line specifies the operations performed by client #1. client #1 deposits \$100 into Account #1, then withdraws \$500 from Account #2.

A sample input file “assignment_6_input.txt” is provided to you. The file name should be hard-coded in your program, and the input file name must be “**assignment_6_input.txt**”. A different data set will be used to evaluate your program for marking purposes where the structure and the input file name will remain the same, and only the data will be different.

You must output the balances of each bank account after all the transactions have been performed. For each bank account, **your output should display the account followed by the account balance**. Please use the following format for displaying the output on screen:

a1 b 500
a2 b 300

Your C program should output results to the screen. For the given input file assignment_6_input.txt, your output should be:

a1 b 600
a2 b 1700
a3 b 3050
a4 b 3800

Sample Input File and Output Format

You must not use any other format, such as user input from the terminal, command-line argument, etc., for providing input to the program. Also, you must not modify the format of the input file. Any changes to the input file or input-taking procedure will not be considered a successful submission for this assignment. The output of your program must follow the format of the sample output given above.

For testing purposes, we have designed the sample input in a way where the final account balances will always remain the same after completing the transactions. However, when testing your program with another input file, note that due to the non-deterministic nature of threads, the final account balances may vary when running your program multiple times.

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 “**assignment5.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 5 FAQ will be made available on OWL as needed. Also, consult TAs and the Instructor for any questions you may have regarding this assignment.