

CSCI 375 Project #2

Due date: July 10, 2018

How to submit? : Submit the hardcopy of your source code and sample result

In this lab you will simulate one of the classical synchronization problems in order to see how the (semi) critical section problem could be implemented using binary and counting semaphores. The lab is due in 2 weeks, 10th July 2018.

5 processes are characterized by 3 readers and 2 writers.

Up to two reader processes can be inside their critical section without any writer process. For writer process to go into its critical section, it should check whether there is any reader or writer process is in the critical section.

Critical section in this problem is reading shared data buffer for reader and updating shared data buffer for writer processes. It is up to you to implement any shared data for readers and writers but you have to specify clearly following things in your sample output.

- When reader or writer enters its critical section, it has to report whether there are any reader(s) or writer(s) other than itself.
- You may print out the data you read or write when you implement real buffer. (Optional)
- You have to print out “Panic Messages” when the rules behind this semi critical section problem are not observed.

In your main program, you run the random number generator function to choose process to execute. The chosen process starts (resumes) execution and after one instruction, it will be returned. (You should force each process run exactly one instruction then returns and waiting for its turn.)

You can implement this using switch statement in C or C++. Do not use any multi-threading nor mutex feature from programming language. Each process is one big switch statement and will be returned after each instruction. You need to keep track of program counter of each process to resume at the right place once it will be chosen to run by keeping global counter variable per process.

Subproject 1: You should implement binary and counting semaphores as studied in the class for this project.

Subproject 2: You should implement swap operation as studied in the class for this project.

You should not copy from others or let other students use your code. Violation to this policy will result in automatic fail. (I might ask you to explain your code as well.)