

CPSC 457 – Fall 2016
Assignment 5
Readers-Writers and POSIX Threads
Due: December 9th @11:59PM
Worth 8% of total mark

Objective: The main objective of this assignment is to work with POSIX threads in order to gain experience with concurrent programs.

Outcomes: In this assignment, you will implement 3 different versions of the reader-writer problem using POSIX threads and study their behaviour.

Background

Read the paper entitled “Process Synchronization with Readers and Writers Re-visited” (available on D2L). You are not required to read the formal verification (proof of correctness) part.

Implementation

Write an implementation for each of the three solutions. You may need the following POSIX threads functions:

```
int pthread_mutex_lock(pthread_mutex_t *mut)
int pthread_mutex_unlock(pthread_mutex_t *mut)
int pthread_mutex_trylock(pthread_mutex_t *mut)
int pthread_mutex_destroy(pthread_mutex_t *mut)
```

For each version, you must provide at least the following functions

- **Reader:** The logic of a reader thread
- **Writer:** The logic of a writer thread
- **Database:** The shared object between readers and writers. It contains all shared information between the various threads
- **Server:** Instantiates and runs the threads; it collects and shows statistics

Questions

For each of the three versions for the readers-writers problem, setup an experiment that allows you to:

1. Plot the average turnaround time of reader threads, for a fixed number of writer threads (100 writers). Vary the number of readers from 5 to 500 in increments of 5. Plot all three versions in one graph.
2. Plot the average turnaround time of writer threads, for a fixed number of reader threads (500 readers). Vary the number of writers from 5 to 100 in increments of 5. Plot all three versions in one graph.
3. Write a short paragraph summarizing your findings.
4. **Bonus:** There can be a behaviour in the third version that limits concurrency; demonstrate it experimentally and describe it in a short paragraph.

Deliverables and Marking Scheme

(Please submit your program to D2L. Submit the following in hard copy in the exact order as mentioned. Marks may be deducted if the ordering is not followed in the submission.)

1. Implementation of version 1 (20 marks)
2. Output for version 1 (5 marks)
3. Implementation of version 2 (20 marks)
4. Output for version 2 (5 marks)
5. Implementation of version 3 (20 marks)
6. Output for version 3 (5 marks)
7. Question 1 (plot) (10 marks)
8. Question 2 (plot) (10 marks)
9. Question 3 (written part) (5 marks)
10. Bonus (Output and written part) (5 marks)

Administrative Information**Teams**

You are advised, but not required, to work in a team of two members. Teams of more than 2 students are not allowed. You and your partner must be in tutorials that are taught by the same TA. One submission per team is required.

Academic misconduct

Any similarities between assignments will be further investigated for academic misconduct. While you are encouraged to discuss the assignment with your colleagues, your final submission must be your own original work. Any re-used code of excess of 10 lines must be cited and have its source acknowledged. Failure to credit the source will also result in a misconduct investigation.

Late submission policy

Late submissions are penalized with 12.5% deduction for each late day or portion of a day. Hence no submission are accepted 8 days after the deadline

D2L Marks

Any marks posted on D2L or made available using any other mean are tentative are subject to change (after posting). They can go UP or DOWN due to necessary corrections.

Happy Coding :)

Administrative Information**Teams**

You are advised, but not required, to work in a team of two members. Teams of more than 2 students are not allowed. You and your partner must be in tutorials that are taught by the same TA. One submission per team is required.

Academic misconduct

Any similarities between assignments will be further investigated for academic misconduct. While you are encouraged to discuss the assignment with your colleagues, your final submission must be your own original work. Any re-used code of excess of 10 lines must be cited and have its source acknowledged. Failure to credit the source will also result in a misconduct investigation.

Late submission policy

Late submissions are penalized with 12.5% deduction for each late day or portion of a day. Hence no submission are accepted 8 days after the deadline

D2L Marks

Any marks posted on D2L or made available using any other mean are tentative are subject to change (after posting). They can go UP or DOWN due to necessary corrections.

Happy Coding :)