

# COP 4600-001: Operating Systems (Fall 2018)

## Group Programming Project

Due by: 11:59 PM, Wednesday, November 14, 2018

You may work in groups of two, three, or four. A group should submit one program and one final report, presentation and all members of the group will get the same grade for the report and different grades for presentation (each member of group must present some part of project).

**Sockets:** The requirement is to develop a writer and a server program that communicates with the writers (identical) by making use of sockets. Three identical writer programs (except for the names, writer 1, writer 2, and writer 3, respectively) will run on one or more machines and the server on another machine.

### Assignment Specifications

- Each writer program connects and sends a message (i.e. message\writer1" will be sent to the server from writer 1).
- The server should then accept the socket connection from a writer, and create a thread (child) after the socket is accepted.
- Once the thread is created, the server is also responsible for the communication (Step 4) with the writer, and then the parent process (in the server) should loop back to listen for other incoming connections - this makes the parent ready for other connections.
- A message from the writer should be stored by the appropriate thread processes in shared memory. This memory is shared by the server and all its children, and the child threads (in the server) can store only one message. The message length will be no more than 15 characters (14 chars plus 1 char string terminator in C), such as \writer1", \writer2" and \writer3".
- A writer process can be invoked (after compiling using the make file provided) with the writer name. For instance, suppose the writer program name is \writerp", the writer should be started by typing \writerp" that will make the writer read the message from the file \input file.txt" (the input file should be present in the same directory as the writer program).
- The shared memory must be protected (critical section) so that the stored message will never get corrupted during the concurrent connections from multiple writers.
- Once a message from a writer is successfully stored in the shared memory by one of the server's (child) threads, after a 2 second sleep, the actual contents of the shared memory should be sent back to the original writer by the same server (child) thread.
- Please set the server up to run on any C4 Lab PC. The server should loop for a total of n connection accepts; where n will be three in our trials.

The writer prints a message just before sending the message to the server. The server should then print the contents of the shared memory (after reading the incoming message into shared memory) and send back the contents of the shared memory to the writer (acknowledgement). As soon as the writer receives the message

## **Format to write your report**

Your report should have the following sections:

- Title: The title should be descriptive and fit in one line across the page.
- Authors: This should be right under the title, says who you are.
- Abstract: This is the paper in brief and should state the basic contents and conclusions of the paper. It should be complete enough to understand what will be covered in the project.
- Intro: This is a short overview of what you did, and what you learnt. This should contain more motivation than the abstract.
- Methodology: Description of algorithms you used and a description of the platform you used to the level of detail such that someone else could reproduce the same experiments elsewhere.
- Results: Include your program, input data, outputs of your program
- Conclusions: Summarize the conclusions here, and discuss things you have learnt during this experimentation.

## **Grading Criteria:**

Total possible points: 100

Code: 65

Report: 25

Presentation: 10

## **Note:**

- This is subjective, but it would not be fair to grade a group that solves a simple problem the same that takes on a challenging one. The amount of challenge depends on the need for advanced structures learned on this course (multi-thread, synchronization) and the amount of logic needed to solve the problem.
- Some points will be given to a correct solution. On top of that, the quality of the solution will be assessed as well. Again, it would not be fair to grade the same a solution that is better than another, given both are correct. Quality aspects include code indentation, code well commented, a factored design, good variable naming, etc.

## **Submission**

- Submission via Canvas Assignment. (Zip file including report, code, slides, input data, and read me file)
- The name of the file should be the name of the students in this format - LastName FirstName (names of all students in the group).
- Presentation will be on Tuesday and Thursday, November 14, 16, 28, and 30.