

**Operating Systems**  
**University at Albany**  
**Department of Computer Science**  
**Chongqing University of Posts and Telecommunications**  
**Computer Science, International College**  
**ICSI 412**

**Project-2**

**Assigned: Saturday, May 20<sup>th</sup>, 2023.**

**Due: Saturday, May 27<sup>th</sup> by 11:59 PM.**

**Student Name:**

**PURPOSE**

The goal of this assignment is to gain an understanding of semaphores through the use of *pthread*s by designing and implementing a multi-threaded application. A secondary goal is to understand how semaphores are used in operating systems and applications. You are to write both a producer and a consumer C programs that uses *pthread* to process and synchronize concurrent file activities.

**PROBLEM**

Your solution must include the following:

1. **The Consumer**

Submits processing requests to the producer by supplying a file name, its location and a character. It also outputs the contents of the file provided by the producer to the standard output.

2. **The Producer**

Accepts multiple consumer requests and processes each request by creating the following four threads.

- 1 The **reader** thread will read an input file, one line at a time. It will pass each line of input to the character thread through a queue of messages.
- 2 The **character** thread component will scan the line and replace each blank character by the character supplied by the client. It will then pass the line to the toUpper thread through another queue of messages.
- 3 The **toUpper** thread component will scan the line and convert all lowercase letters to upper case. It will then pass the converted line to the writer thread through another queue of messages.
- 4 The **writer** thread will write the line to an output file.

The producer will also return both the file name as well as its location to the consumer when the end of input file is reached.

**IMPLEMENTATION DETAILS**

1. You must develop a module that implements a queue of character string buffer.

2. This structure will be an array of pointers to strings with integers (pointers) to indicate the head and tail of the list.
3. The maximum size of the buffer array will be 10.
4. Buffers will be created by the reader thread and destroyed by the writer thread.
5. Threads should terminate when end of input file is reached.
6. Your solution **must not** use any libraries for the memory sharing management, except the *pthread* library. This means all data structures included in your solution must have been developed by you.

## WHAT TO SUBMIT

The following are to be submitted to your co-instructors:

Your solution must be placed in a Microsoft compressed (zipped) folder (.zip). Your .zip folder must be named: *412 Project 2- Your Name*.

Your program should be layered, modularized, and well commented. The following is a tentative marking scheme and what is expected to be submitted for this exercise:

1. External Documentation including [5-10 pages]:
  - a. Title page.
  - b. A table of contents.
  - c. [20%] System documentation.
    - i. [8%] A high-level data flow diagram for the complete system.
    - ii. [5%] A list of routines and their brief descriptions.
    - iii. [5%] A list of all semaphores and/or mutexes used and a description of the specific behavior each one is assigned to. This means you are to describe the purpose of each semaphore/mutex included in your solution.
    - iv. [2%] Implementation details.
  - d. [5%] Test documentation.
    - i. [4%] How you tested your program.
      - Data used for all tests done.
      - Testing outputs produced.
    - ii. [1%] List of program bugs and your plans to address them.
  - e. [5%] User documentation.
    - i. [3%] How to run your program.
    - ii. [2%] Describe parameters (if any).
2. Source Code
  - a. [65%] Correctness.
  - b. [5%] Programming style.
    - i. [2%] Layering.
    - ii. [1%] Readability.
    - iii. [1%] Comments.
    - iv. [1%] Efficiency.