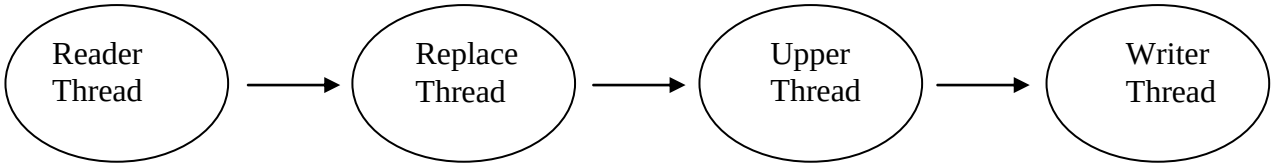
CSE 333 - OPERATING SYSTEMS
Programming Assignment # 3 DUE DATE: 27/12/2016 - 23:00

In this assignment, you will write a program that uses threads and synchronization. You will implement a program with four threads, structured like:



- The Reader Thread will read an input file, one line at a time. It will take the each line of the input and pass it to Replace Thread.
- The Replace Thread will scan the line and replace each blank character with a dash ("-") character. It will then pass the line to Upper Thread.
- The Upper Thread will scan the line and convert all lower case letters to upper case (e.g., convert "a" to "A"). It will then pass the line to Writer Thread.
- The Writer Thread will write the line to `stdout`.

Program Details

- The Reader Thread will read in each input line. If the line is longer than 63 characters, it will truncate it to 63 characters (plus the null byte at the end). Just throw away any extra characters.
- The Writer Thread will count the number of lines and print this number to `stdout`.
- Threads should terminate when there is no more input (end of file).
- There should be a separete queue in the middle of each consecutive threads.
    - A queue between the Reader and the Replace threads.
    - A queue between the Replace and the Upper threads.
    - A queue between the Upper and the Writer threads.
- Each queue needs its own synchronization, which can be realized by semaphores.
- Do not use global variables to signal completion of the input for the various threads.

- The main thread is responsible for creating threads and waiting for the completion of them.
- Preserving consistency and preventing deadlocks are major issues to be considered.
- Multiple simultaneous operations on different lines should be allowed in your synchronization solution.
- Your program should take an input file for the Reader Thread.
    - Example:

      ./Project3.out input.txt
- Your program should produce a log file.
    - Example:

<Thread-type>	<The line before update>	<The line after update (if possible)>
Reader-Thread	"This is a line."	-
Reader-Thread	"This is another line."	-
Replace-Thread	"This is a line."	"This-is-a-line."
Replace-Thread	"This is another line."	"This-is-another-line."
Upper-Thread	"This-is-a-line."	"THIS-IS-A-LINE."
Writer-Thread	"THIS-IS-A-LINE."	-
Upper-Thread	"This-is-another-line."	"THIS-IS-ANOTHER-LINE."
Writer-Thread	"THIS-IS-ANOTHER-LINE."	-
.....	... .	

Notes:

- The project will be done in Linux operating system using C programming language.
- You must use PThread library and synchronization appropriately in your code.
- Take into account materials and examples covered in the lab sessions.
- Consider all necessary error checking for the programs.
- No late homework will be accepted!
- In case of any form of copying and cheating on solutions, all parties/groups will get ZERO grade. You should submit your own work.
- You have to work in groups of two.

What to submit?

A softcopy of your source codes which are extensively commented and appropriately structured and a project report (minimum 2-page) that contains the detailed information about your implementation should be emailed to cse333.projects@gmail.com. All the files should be submitted as one zip file. You should use your surnames as the name of the file: surname1_surname2_project3.zip