Operating Systems - Project 3

University at Albany
Department of Computer Science
ICSI 500

Project 3
Assigned: Thursday, November 9th, 2023
Due: Saturday, December 2nd, by 11:59 PM. Submissions with 20% penalty will be Accepted by Monday, December 4th, by 11:59 PM.

Note: Unlimited number of submissions is allowed.

PURPOSE

Develop a practical understanding of task collaboration using socket programming, semaphores, multithreading.

OBJECTIVES

Develop a client/server application using Linux TCP sockets and the C programming language. Your solution will be based on the code you developed for project-2. You are to add to the Chat system you developed for project-2 an additional service provided by the server to all client communication.

PROBLEM

You are to develop a data processing system to process strings of characters. The solution must replace all instances of lowercase vowels with uppercase ones.

You are to modify your Chat application such that the server will collaborate with another node in order to successfully respond to requests from the different clients. The server you developed for project-2 will share, through a socket, the data component of the frames received with this new helper node. Your project-2 server will forward the text information from the frames to the helper node for processing. The helper node will create six threads, process the data received, and use a socket to share the generated results to the server. All communications between all nodes will be encoded according to the format defined in project-2.

THE HELPER NODE

The word provided by this new node is structured as **serverDecoder** and **ServerEncoder.** Such The **ServerDecoder** will obtain the text component of the client message, and will create the following six threads:

1 The **charA** thread will read data provided by the decoder process and will replace all lowercase **a** character with its corresponding uppercase. It will share the data received with the *charE* thread through a queue of messages.
2 The **charE** thread component will scan the data shared and replace all lowercase **e** characters with uppercase E. It will then share the modified data with the *charI* thread through another queue of messages.
3 The **charI** thread component will scan the data shared and replace all lowercase **i** characters with uppercase **I**. It will then share the modified data with the *charO* thread through another queue of messages.
4 The **charO** thread component will scan the data shared and replace all lowercase **o** characters with uppercase **O**. It will then share the modified data with the *charU* thread through another queue of messages.
5 The **charU** thread component will scan the data shared and replace all lowercase **u** characters with uppercase **U**. It will then share the modified data with the *writer* thread through another queue of messages.
6 The **writer** thread will share the data received with the **serverEncoder** component.

The **serverEncoder** will share the modified text with the chat server through a socket connection.

HELPER NODE IMPLEMENTATION DETAILS

1. You must develop a module that implements a queue of character strings.
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 5.
4. Threads should terminate when end of input data is reached.

TESTING

You are to use the same data you have used for testing your project-2.

DOCUMENTATION

Your program should be developed using GNU versions of the C compiler. You are to use POSIX Threads only. This means your solution must use the functions available in the pthread.h header file. It should be layered, modularized, and well commented. The following is a tentative marking scheme and what is expected to be submitted for this assignment:

1. External Documentation (as many pages necessary to fulfill the requirements listed below.) including the following:
    a. Title page
    b. A table of contents

c. [20%] System documentation
    i. A high-level data flow diagram for the system
    ii. A list of routines and their brief descriptions
    iii. Implementation details
d. [5%] Test documentation
    i. How you tested your program
    ii. Testing outputs
e. [5%] User documentation
    i. Where is your source
    ii. How to run your program
    iii. Describe parameter (if any)

2. Source Code
    a. [65%] Correctness
    b. [5%] Programming style
        i. Layering
        ii. Readability
        iii. Comments
        iv. Efficiency

WHAT TO SUBMIT

The following are to be submitted through **Blackboard**:

1. Your documentation for project 3. This document must be typeset and saved in MS Word.
2. Copies of all your source code files as well as their executables.
3. Copies of all data used for the testing of your solution.
4. All input files used and their generated output files.

You are to place all files that are related to your solution in a .zip file. Your .zip file must follow the format: *500 Project3 Your Name*. Marks will be deducted if you do not follow this requirement.