CS 4348: Operating Systems
Programming Assignment 2

Instructor: Ravi Prakash

Assigned on: June 29 2022
Due date: July 10, 2022

This is an individual project and sharing of code is strictly prohibited and will be dealt with as per the university's rules governing academic misconduct. You are expected to demonstrate the operation of your project to the instructor or the TA. You are required to write your code in C, C++ or Java. The programs must run on UTD machines (dc01, dc02, ..., dc45).

In this project, you will use socket connections for message passing between processes running on different machines. Additionally, you will use multithreading to send and receive messages in parallel. Please contact the TA for help if you are unfamiliar with these topics.

Assumptions

- There are four processes in the system, numbered from 1 to 4. Each process is run on a separate machine. You cannot log directly into these machines. First, you need to be on the university network. For additional help, please contact the TA.
- There are reliable socket connections (TCP) between each pair of processes.
- Each process is listening for incoming messages from other nodes.
- Each process, upon starting, takes input from the user.

Functionalities

- The inputs from the user are the command being handled at the processes. In this project, the only commands to be implemented are as follows:
  - **Send message to another process:** The following command should send a message to the specified process:

    send receiver_id MESSAGE
    (e.g.: "*send 1 Hello!*" sends *Hello!* to process 1)
  - **Send message to all processes:** The following command should send a message to all the other three processes:

    send 0 MESSAGE
    (e.g.: "*send 0 Hello!*" sends *Hello!* to all processes)
  - **Stop:** sends a *Stop* message to all other processes, and marks own state as *stopped*. When a process has received *Stop* messages from all other processes and its own state is *stopped*, the process can close all its socket connections and exit.

  For the purpose of this project, assume that the value of *receiver_id* is in the range 1 through 4, and a process does not send a message to itself. Also, sending a message to all the other processes is to be implemented at three separate messages, one sent to each recipient, with the same content.

- Each process, upon receiving a message, prints the message on its display.
- Each process has a thread for listening to incoming messages and a separate thread for accepting inputs from the user. Therefore, listening for incoming messages and accepting commands from the user should be done concurrently.

Submission Information

The submission should be through eLearning in the form of an archive consisting of the source code.
Your source code must have the following, otherwise you will lose points:

1. Proper comments indicating what is being done
2. Error checking for all function and system calls