CSE232: Assignment 2

Deadline: October 7, 2022

Note: This is a lengthy assignment. If you do not start by tomorrow you will not be able to finish it on time.

Write a client server socket program in C. Your program should have the following features.

1. **Write a client program:**
    a. **Sequential client:** Each client generates 20 requests; Each request consists of an integer, i.e., i^{th} request is an integer "i" ranging from 1 to 20; display the received response on the terminal; close the connection after 20 requests. **[3]**
    b. **Concurrent client:** For designs 2(b), 2(c), and 2(d), the client program is a multithreaded program that generates concurrent requests. For example, if we want 10 concurrent client programs, the client code should run 10 parallel threads. **[5]**
2. **Server program:** opens a new file when the first client connects; for each client request: compute the factorial of the received number; store the result along with client-id (IP address, port number) in the file and also send the result back to the client; close the file when all the client connections is closed.
    Design the following server programs
    a. Sequential server program **[5]**
    b. Concurrent server program with multiple processes (using fork system call) **[3]**
    c. Concurrent server program with multiple threads (using pthreads) **[3]**
    d. Non-blocking server that can manage total 10 clients
        i. Using select() system call **[5]**
        ii. Using poll() system call **[5]**
        iii. Using epoll API **[5]**
3. For each server design, each server (except for design 2a) concurrently processes 10 client requests. Note that you should be able to run 10 concurrent client programs. Observe the following metrics.
    a. Time required by the server to process all (i.e., 10) concurrent client requests **[2.5]**
    b. Server process's CPU utilization (use "top" or "htop" command) **[2.5]**
    c. Server process's memory utilization (use "top" command) **[2.5]**
4. Justify the results obtained in (3). **[2]**