

**Jackson State University**  
**Department of Computer Science**  
**CSC 435 Computer Networks, Spring 2019**  
**Instructor: Dr. Natarajan Meghanathan**

**Lab Project # 3: Concurrent File Server (using Java Threads) and Comparison with an Iterative File Server**

**Due:** March 18, 2019: 4 PM

**Max. Points:** 100

**Project Description:** Develop a concurrent file server that sends the contents of a file, line by line, to the requesting clients. The client program sends the name of the file to be downloaded to the server. Use connection-oriented sockets for implementing the client-server communication. The concurrent file server will be implemented as follows: The concurrent server spawns several threads, one for each client requesting a specific file. The server creates the thread by passing the name of the file as the argument for the thread constructor. From then on, the server thread is responsible for transferring the contents of the requested file, line by line. Let the transfer size be one line per message. After a flush operation that leads to a message (line) transmission, the server thread sleeps for 200 milliseconds before sending the next line.

In addition to the concurrent server, implement an iterative server that processes only one client request at a time (i.e., a client has to wait in a queue until all the clients ahead of it in the queue have been served). The iterative server also sleeps for 200 milliseconds after sending a line.

Test your concurrent file server and iterative file server with three client programs that request files with different number of lines. Create your own text files of 8, 64 and 256 lines. You can just have a word say “OK” per line – copy and paste it in a binary exponential order (1 line, 2 lines, 4 lines, 8 lines and so on). Determine the total time it took for each client to download their file from each of the two servers.

**Testing:** For testing with each of the two servers (concurrent and iterative servers), the three client programs are to be started in the decreasing order of the number of lines in the files requested (i.e., start first the client that requests the file with 256 lines; then start the client that requests the file with 64 lines and then start the client that requests the file with 8 lines).

Run the iterative server first and have three client programs contacting the server in the decreasing order of the number of lines in the requested file. Determine the time taken for each of the three clients with the iterative server.

Repeat the above testing procedure for the concurrent server.

**Where to run the programs:** You could run on your personal computer, using the localhost option.

**What to Submit:**

(1) **A video file** (either one of these formats: .mp4, .wmv, .avi) that is generated by desktop recording your explanation of the working of your concurrent and iterative file servers as well as the client and the logic/approach you took to implement the programs to satisfy the design requirements. You should display the program(s) on the desktop and walkover the different sections of your client and server code as well as explain the execution flow of the programs. You should also record demonstrating the working of your programs: first the iterative server and the three clients, followed by the concurrent server with the three clients.

Note that the contents of the desktop/programs captured through your video should be clearly readable. Submit the video through Google Drive (using your JSU email address) and send the link via email to [natarajan.meghanathan@jsums.edu](mailto:natarajan.meghanathan@jsums.edu)

You could try using one of the **desktop recording software** (or anything of your choice):

CamStudio: <http://sourceforge.net/projects/camstudio/files/legacy/>

Debut: <http://www.nchsoftware.com/capture/index.html>

(2) A **report** featuring the following: Your code for the iterative server, concurrent server and the clients and snapshots of at least three clients contacting the iterative server with the time taken for the file transfer in each case, as well as snapshots of at least three clients contacting the concurrent server with the time taken for the file transfer in each case.

**Submission:**

(1) Video file shared (to: [natarajan.meghanathan@jsums.edu](mailto:natarajan.meghanathan@jsums.edu)) via Google Drive

(2) Hardcopy of the report, submit in class