CS6650 Fall 2019

Lab 3 - Playing with Sockets

Aims:

1. Experiment with a multithreaded socket server
2. Understand barrier synchronization witha CyclicBarrier

Run the Socket Server

1. Copy and build the code from week 3
2. Run SocketServer.java and leave it running - it waist forever for requests
3. Run SocketClientSingleThreaded.java. This should send a message to your server. Check the output window to ensure it works
4. Look at the code in these files to make sure you understand what is going on

Complete the Socket Client

1. Open SocketClientMultithreaded.java and read the code
2. The // TO DO comments point you to where you need to insert code to initialize the CyclicBarrier, create the client threads and implement the barrier synchronization
3. You also need to insert the CyclicBarrier handling into SocketClientThread.java
4. Run the code. You should see multiple requests being processed by the server.
5. Run this test a few times. What is the maximum numbers of active servers you see?

Modify the server to use a thread pool

1. The server right now creates a thread per request. What do you think will happen if you create 100's or 1000's of simultaneous clients? Your laptop might not have enough resources to test this, but you should be able to guess :)
2. Create a new server that utilizes a FIXED SIZE thread pool. Make a sensible guess how large this pool should be (e.g. 20?)
3. In the client, tale two timestamps, one before any threads run and one after all threads complete. Print out the wall time (test duration) in milliseconds before exiting
4. Experiment with different nimbers of clients and thread pool sizes. Do you see much variation in

the wall time for a test?
5. Compare the performance of the system with the original server and your modified server? Is there a noticable difference in performance?

If you want to look at solutions, they are here

Back to Course Home Page