

Server-client system Lab

Simon Rogers

18th Feb 2015

Introduction and aims

The University feels that its staff are not suitably motivated. So, they've decided to build a computer system that will randomly send motivational quotes to staff members every 5 seconds. Your task is to build a Server and Client in Java to test this out.

Tasks

Note that we might spend two weeks on this - there is more than an hours work here.

- Start by building a system that just allows a client to connect to a server and, once the connection has happened, both the client and server close (see e.g. `SimpleServer.java` and `SimpleClient.java`).
- Now modify this so that the server sends a randomly chosen motivational quote to the client every 5 seconds (e.g. `DateServer` and `DateClient`). The `QuoteLoader` class and `quotes.txt` will give you an array list of twenty motivational masterpieces. `LoadExample.java` shows you how to use them.
- The next step is to allow multiple connections from different clients (`DateServerX`)
- After using this system for a while, the University realises that it's a bit annoying and want to try a new system where a quote is only sent to the client when the client sends the string "MOTIVATEME" to the server.
- The system is so popular that staff members are trying to suggest their own quotes. Modify your system (and protocol) to allow clients to add quotes if they send "ADDQUOTE:this is my quote".
- University management don't need motivation, but would like to know how many users are currently connected, how many quotes have been requested and how many quotes there are currently in the system. Create a client for them. The server should only respond when they send particular strings relating to those three statistics - you can decide which strings to use.
- Your system should be able to cope with clients entering and *leaving*.