Assignment 3 Testing

Manual:

1) Compilation of C++ Client – Make sure bin folder is empty (rm bin/\*) and run “make”.

If make completes with an executable – 50 points

If make has less then 5 warnings – 25 points

If make has less then 10 warnings – 25 points

2) Compilation of Java Servers – Make sure there is no target directory (~/bin/mvn clean) and run “~/bin/mvn compile”

If make completes without errors – 50 points

If make has less then 5 warnings – 25 points

If make has less then 10 warnings – 25 points

3) protocol.start(...) call must end before any call to protocol.process and must NOT be called by the main thread (Reactor's selector thread or TPC accept loop thread)

This means:

TPC – Must call it from the run function in the ConnectionHandler (or something the run() calls too)

Reactor – Must call it via ActorThreadPool for the connection. Best way is to send it to the pool.submit function in the HandleAccept function (Reactor.java) Can also be done in the process lambda built in the continueRead function (NonBlockingConnectionHandler.java) with an  
if (firstTime) protocol.start(...). This is ugly but OK

TPC correct – 100 points

Reactor correct – 100 points

4) Server Pattern still general. Make sure BaseServer class and Reactor class do not know the actual protocol or message type. It can (must) know the Connection<T> interface implementation but cannot know the BidiMessagingProtocol or MessageEncoderDecoder implementation. It must work with those interfaces and be generalized using <T>. You can just look at the constructors to make sure.

TPC correct – 100 points

Reactor correct – 100 points

~~5) Running client via valgrind –leak-check=yes.~~

~~Run an instance of the server, run an instance of the client. Do “LOGRQ test” after getting ACK 0 do “DISC” (for discconect) after getting ACK 0 the program should terminate (if not then press any key)~~

~~program ending by itself (not ctrl + c) – 25 points~~

~~no errors or warnings from valgrind – 75 points~~

Automatic:

A program called TestSpl171Ass3 is attached. It is responsible for testing a single assignment (after compilation). The program should run **TWICE** per group. In one instance on the original application. In the other take the Client folder from “Solution” and replace with their client and re run the application. You can use a different output csv for each.

It receives the following parameters:

1) Path to student directory – Wherever the Server and Client directories sit. This should be the first thing inside the zip.

2) port – The port for both server and clients.

3) output file - A .csv file where the results will be appended.

4) Group number – just group number... for the students OFC

5) Add header? - if there is a fifth parameter, a header including “group No, Test1\_name, …, TestN\_name” will be appended before the actual scores to the file. It's good to call it for the first group

There is also a Logs folder. The logs for each group No / test / client & server will be saved there. We can later use it to verify problems.