

Design Document: Simple Remote Shell

Aaron Nguyen

CruzID: anguy200

1 Goals

The goal of this assignment was to create a simple remote shell using c/c++. This shell was required to be able to connect to the server from any location so on the same network or through the internet. In addition the server was required to be able to have multiple connections and be able to concurrently communicate with each client.

2 Design

To achieve code from Lab 2 was used primarily the TCP socket code. Like creating the socket and filling it with the destination info, and establishing the connection between the source and destination. The client portion of the assignment was simple using functions like readline, send, and recv. All the client needed to be able to do was get inputs from the user, send it to the server, and wait to receive a response from the server. The server was a little bit more complicated. Forking was used to allow for the server to have multiple concurrent connections. Popen was used to execute terminal commands. Error handling is handled using try and catch. If the error is due to a connection issue the server will just exit out of the fork and does not crash. In terms of testing it was primarily done in the UCSC timeshare with one server running and having upwards of 1 to 10 clients connecting the server at the same time spamming the server with random strings and commands.

Protocol and Handshake

The protocol used for communication between client and server was TCP. To indicate the end of the message was the use of “\r\n” being added at the end of each message sent between server and client. Both the client and server utilize a recv while loop so if recv returned a 0 that indicates that the connection between the server and client has been closed. In terms of processing the server is handling a majority of the load due to it having to execute the command in the machines local terminal.