Wei Fei (2538810)
EECS678 Lab4 Report
Prasanth Vivekanandan
02/25/15

1. Briefly describe the design of the program. How does your program control when the client runs and when the server runs?

In this program, we need to create both of server and client process. First, we need to I create a handshake socket, which can listen to the connection requests from client. We also bind the handshake socket to an address. When we create client socket, the listen socket from the server can accept the connection request from the client.

When we run the program, we need to run server firstly, and then run the client. Because the server can make a handshake socket to listen to communications to the socket address. And the client can connect to the handshake socket in the same address. When they work together, the client can write message and send it to the server and the server can read message from the socket. For example, in "client.c" file, we have the function "write(sockfd, strs[i], BSIZE);" And in "server.c", we have the function "while (read(session_sockfd, buf, BSIZE) > 0)" to read message until reading them all. Also, no matter what from server to client or client to server, either way works in socket.

2. What is the purpose of the handshake socket? Why not have the server create and bind session sockets that clients may connect to directly?

Handshake socket achieve a two direction ways and let datum can be passed away fast between server and client. Server and client can make secure connection through handshake socket and server will send messages to client spontaneously, but does not need multiple requests from client.

We do not have the server create and bind session sockets that clients may connect directly because handshake connection is only for listening. Before send data, the client needs to know whether the server is ready to communicate.

3. For the simple / client server program, we chose to use sockets instead of pipes to send messages between the client and server. Why are sockets preferred over pipes for this program? Give at least two reasons.

First, sockets can send message and communicate in both of two directions, but pipes only can send message in one direction.

Second, sockets can be used for IPC in different host systems, which can let server and client communicate with each other conveniently. However, pipes only can communicate in the same host system, which are between two processes in one host system.