

TCP Programming Project

*Assigned: November 20**Due: December 4, midnight*

Overview

For this assignment, you will write a client and a server that use TCP stream sockets to communication. The server provides a random number generation service. The server waits for TCP connections on a port that you choose. After a client connects to that port, the following interaction should occur:

1. The client sends the server a string identifying itself.
2. The server sends the client a confirmation string with the client's identification information and a random number.
3. The client closes the connection

Resources

The course web page has pointers to several sockets tutorials and a FAQ (frequently asked questions) for UNIX sockets. Feel free to also post questions to the class newsgroup. Obviously you should first check to see if your question has already been asked and answered on the newsgroup.

Details

The client and server communicate with each other by exchanging **lines of ASCII characters** using the reliable byte-stream service provided by TCP. Thus, the socket type is `SOCK_STREAM` and the family is `AF_INET`. The details of the protocol are as follows:

1. The server listens for connections on a port of your choosing.
2. The client opens a connection to the server's socket.
3. The server accepts the connection and waits to receive a request string from the client.
4. After connecting, the client sends a request string of the format:

`<WS> <connection-specifier> <WS> <username> <newline>`

where:

- The `<connection-specifier>` is of the form:
`<server endpoint specifier> <WS> <client endpoint specifier>`
An endpoint specifier is of the form `<dotted quad>-<port number>`.
- The `<username>` is the student's name, in the form of first initial, middle initial, last name, all one word with no whitespace (e.g., "G.P.Burdell").
- `<WS>` is "whitespace", one or more blank or tab characters.
- `<newline>` is the end-of-line marker, represented in the C language by the single character `'\n'`.

Thus an example of a client request is:

130.207.8.20-14892 130.207.114.53-2092 G.P.Burdell\n

5. After receiving and parsing the request string, the server responds by sending one or more lines terminated by '\n'. The first line always contains an identification ("CS 6250 Server") and the date and time. If the request was properly formatted, and the connection-specifier refers to the current connection, the second line will contain the string "OK", followed by the identification information from the client request, followed by a random integer. For example:

```
CS 6250 Server Tue Nov 18 14:27:00 EST 2003\nOK G.P.Burdell 1742923819\n
```

If the request is not properly formatted, or does not refer to the present connection, the second line will contain an error message.

6. After sending the second line, the server waits for the client to close the connection. Upon receiving the second '\n' character, the client closes the connection. When the server sees that the client has closed the connection, it closes also, and waits to service the next client request.