**Socket Chat**

**REQUIREMENTS**

**CLIENT:**

Connect to server

Send Message

Recieve Messages

Interactive Terminal UI

**SERVER**:

Manage Connections

Keep a log of recent messages

**DETAILS**

**CLIENT**

**Connect To Server:**

Connect by IP in command line.

“client localhost:8096 username”

**User Interface:**

Interface is separate from actual program.

Curses UI with a text box for chat and an input for sending messages.

Windows are managed by a separate thread.

Send box and receive box are separate windows.

**Send Message:**

Completely disconnected from receiving chat.

Successfully sent messages will be sent back by server.

Messages prefixed with / will be processed before sending, allowing to control the program by command.

Messages will have a max length of 1000 characters. Strings are null terminated.

**Recieve Message:**

Messages are listened for by polling.

Recieved messages are printed out in order of when they were received.

A typical received message looks like this:

`[1] username: A test message`

Where the [] dictates the user number, username being the name given to the server on connection, and the message body always appearing afterwards.

**SERVER:**

**Manage Connections:**

The server will correlate connections with usernames, connection number, and the connection socket itself.

Messages sent to the server will be relayed to every connection including the client which sent the message.

The received message will be formatted by the server before relaying.

**Message Log:**

Simple log which keeps the last arbitrary number of messages sent.

Log will be sent to any new clients connecting to the server.