Justin Ard
Matt Johnson
Andrew Montgomery
cs 455
Project Proposal

For our group project we have selected to implement an Internet Relay Chat (IRC) program. Our implementation will include both the client and server applications, as well as the core features of the IRC messaging and command protocol.

Background

IRC is an open protocol created by Jarkko Oikarinen in 1988 as a new paradigm for implementing real-time Internet text communication both between individual clients, and amongst groups of users. In addition to offering one-to-one and conferencing chat features, IRC has been widely expanded in modern applications to include file transfer and various other services. However, our intention is to simply execute the messaging functionality in our implementation.

RFC 1459

In 1993, the original protocol for IRC was published in Request For Comment (RFC) 1459. This document stipulates the features, as well as the types and format of communications that must be implemented in IRC servers and clients. The components of our program, and their design will be taken primarily from this document.

Features

Core IRC features that we will implement include:

- A client program for the end user to send and receive messages.
- Multiple 'channels' to host groups of users in chat room environments.
- One or more networked servers to host and manage chat channels.
- One-to-many communication in which clients can simultaneously chat with groups of users.
- One-to-one communication in which users can privately chat with each other.

Possible extended features may include:

- Direct Client Communication (DCC) protocol for establishing direct, non-relayed communications between individual users.
- Client-To-Client Protocol (CTCP) which includes features for:
  - sending/receiving structured data (graphics, voice, fonts, etc)
  - querying individual user clients
- Persistent username and channel registration.
- Encryption using the Transport Layer Security (TLS) protocol

Technologies

Our server and client applications will be written in the Java programming language which will allow them to be ported to various platforms. We will be implementing our own IRC program from scratch using communications over TCP Sockets, however we may also use the Java libraries IRClib (http://sourceforge.net/projects/moepii/), Realy-IRC (http://relayirc.sourceforge.net/) and PircBot (http://www.jibble.org/pircbot.php) as references. Per the IRC protocol, messages will be composed of 8-bit ASCII text characters, and transferred using the Transmission Control Protocol (TCP). Technologies to be used for hosting server applications is still to be determined.

Responsibilities

Each group member will contribute to the design and implementation of the overall program, however individual responsibilities for specific components are assigned as follows:

- One member will be responsible for implementing and documenting the client: Andrew Montgomery
- One member will be responsible for implementing and documenting the server: Matt Johnson
- One member will be responsible for creating, documenting and contributing to the implementation of the application protocol at both ends: Justin Ard

Milestones

In order to maintain sufficient progress throughout the semester, we plan to complete the program incrementally following the tentative schedule below.

Milestone	Estimated Date of Completion
Design specifications with full description of commands and functions to be implemented	Friday 09/30/11
Working client program capable of sending and receiving messages using TCP sockets	Wednesday 10/05/11
Working server program capable of sending and receiving messages using TCP sockets	Wednesday 10/12/11
Project progress report	Wednesday 10/12/11
Client-Server communication over TCP sockets	Thursday 10/20/11
One-to-one communication between two clients through the IRC server	Monday 11/14/11
One-to-many communication through a single channel capable of group chat	Monday 11/14/11
One-to-many communication through multiple channels created on command	Monday 11/14/11
Project progress report	Monday 11/14/11
Full channel ops management functionality for customizing individual channels	Monday 12/05/11
Project presentation	Monday 12/5/11
Project final report completed	Monday 12/12