

Status of This Memo:

This memo is provided as-is. The protocol is relatively new and has not yet undergone extensive testing and therefore this memo is open to review, comments and questions. The protocol may change in the near future. Distribution of this memo is unlimited.

Abstract:

This memo defines a simple chat protocol between a server and multiple clients for the Internetworking Protocols class at Portland State University.

1. Introduction

The protocol supports a client/server architecture. All connections are made over TCP/IP. The server listens on TCP port 23456. All communication happens over this channel. The client sends messages to the server and the server responds. All valid message from the client shall follow the message format as described below. The server need not respond to any invalid messages, although the current reference implementation does. All text sent from the client to the server and visa-versa shall be encoded in UTF-8. Server responses are sent without any particular format to be displayed as-is by the client.

2. Concepts

2.1 Rooms

Clients can join and create rooms. Clients who are members of rooms can broadcast messages to everyone in the room by targeting the room with the PRIVMSG command. A room is created when a client joins a non-existent room. A room is deleted when the last remaining client leaves a room.

2.2 Nicks

Clients are identified by their Nick. When a client connects to a server, they are required to send their chosen Nick via the NICK command. If a Nick is already in use, the user client will have to choose a new Nick. The server may choose to disconnect the client, or simply to prompt the client for a new NICK command.

3. Architecture

3.1 Server

The server listens on port 23456 for incoming connections. When a client connects, they must provide a desired name via the NAME command. If a name is not provided or if the name is already in use, the server has the option of either disconnecting the client or waiting until a NAME command is sent.

When the server undergoes a controlled shut down, the following message must be sent to all connected clients:

"Server shutting down."

3.2 Client

The client is responsible for sending well-formatted commands to the server. The client connects to the server on port 23456. After providing the Nick, the client can send whatever commands are needed.

4. Commands

Following is a listing of available commands which can be sent by the client. All messages from the client to the server have the following format, in BNF format:

```
<message>      ::= ':'<command> <whitespace> <params> 'CR' 'LF'
<params>       ::= <letters> <whitespace> [<params>]
<whitespace>  ::= <SPACE> [ <whitespace> ]
<letters>     ::= <character> [ <letters> ]
<character>   ::= <any valid UTF-8 character>
```

Note the command is terminated by a Carriage Return and Linefeed characters.

Some additional syntaxes are as follows:

```
<targets>    = <nickname>|<roomname> [',' <targets>]
<nickname>   = <Any letter that is not '#'><letters>
<roomname>   = <#><letters>
```

4.1 Nick

Command: NICK

Parameter: <nickname>

The NICK command should be the first command sent by the client upon connecting to the server. The user provides a nickname that they would like to be identified as. If the nickname is already present, the server may decide to drop the user or prompt for a new Nick

4.2 Join

Command: JOIN

Parameter: <roomname>

The JOIN command allows a client to join a room. If the room does not exist, the room will be created and the client will be added to the room.

4.3 Part

Command: PART

Parameter: <roomname>

The PART command allows a client to leave a room. If the room becomes empty after the client leaves the room is removed from the server. If other users exist, they shall receive the following message: "***<nickname> has left <roomname>"

4.4 Names

Command: NAMES

Parameter: <roomname>

Upon receiving the NAMES command, the server will send the client a message for each user who is current in the roomname parameter.

4.5 List

Command: LIST

This command has no parameters. Upon receiving this command, the server will send the client a message for each of the rooms that currently exist on the server.

4.6 PrivMsg

Command: PRIVMSG

Parameter: <targets> <letters>

When the server receives this command, it sends a message to all the targets listed in the parameter. If the target is a room, every member will receive a message from the client. If the target is a nick, if a client with that nick exists on the server, they will receive the message from the client. If the nick does not exist, the client will receive a message that the nick does not exist on the server.

5. Error handling

The server must account for unexpected breaks in the TCP connection. If this occurs, the client is removed from the server in the same manner as if he had run the QUIT command.