CS 494P/594P
Request For Comments: log25
Category: WiP

Logan Peticolas
Winter 2022

Internet Relay Chat Program

Status of This Memo

This memo defines an the IRC (Internet Relay Chat) program to be developed for the final project of CS 494p/594p. Discussion and suggestions for improvement are requested. Please refer to the submitted project proposal for the proposed specifics regarding the proposed grading criteria.
Distribution of this memo is unlimited.

Abstract

The application to be developed is the final project for CS494/594. It is a limited IRC program using the client/server structure using a TCP connection.

This document defines both th client and server protocol. The reader is not assumed to be familiar with the IRC architecture, only the fundamental structure of instant messag communication.

Table of Contents

1. Introduction

For the final project of the CS 494p/594p course, an IRC program will be submitted. This document describes the specifications in detail of this program. Features to be covered include the functionality of the user/client model, the role of said clients and their functionality, as well as that of the server. The goal goal of this program is to increase understanding of networking protocols as well as improve familiarity with socket programming.

2. Labels

This section describes the primary components of this software

2.1. Server

This is a single server implementation, with the location of the server local to the first client's machine. The server has no identifying name. The server has a capacity of 32 clients and 16 sub channels.

2.2. Clients

Clients connected to the server using a TCP connection. Clients are logged by their nickname, local username, and machine hostname. If any of these fields are flawed or duplicate to an alreadyexisting client, the server will terminate the connection with that client. A nickname is a clients display name and must be less than 9 characters long. A client may change their nickname at anytime so long as it does not conflict with an existing user.

2.3 Channel

A channel (also known as room) is a subgroup of the server specific to a chosen topic or theme. Clients may create, join, leave, and communicate specifically with channels. A max of 8 rooms may be instantiated at one time.

3. Messages

Messages sent between client and server have a strict packet format functionally identical to that of the IRC protocol (RFC 2812).

3.1 Content

A message contains at the minimum is a non empty string. Optionally this string may beijing with a command and follow with parameters to that command and/or a context message. Different commands have different requirements. If data required for a command is missing or incorrect the entire string message will be treated as plain text. If an error is encountered when the server processes a message from a client, an error response will be generated and sent to only the original sender. The server does not provide affirmative responses. Available commands (3.3). Message format (3.4)

3.3 Command Format

To send a message a client will provide a string to the server in the form:
“/”[COMMAND]{SPACE}[PARAMETER]{SPACE}[CONTEXT]
[COMMAND] = Command (3.4)
[PARAMETER] = Parameter to complete command. Actual requirement dependent on specific command
[CONTEXT] = Context to be displayed to clients relevant to command used

3.4 Message Packet Format

A client must formally send this string via TCP connection in a packet specified in the following format:

```
    message    =  [ ":" prefix SPACE ] command [ params ] [context] crlf
    prefix     =  servername / ( nickname [ [ "!" user ] "@" host ] )
    command    =  4*letter
    params     =  *1( SPACE middle ) [ SPACE ":" trailing ]
    context    =  *1( SPACE middle )

    nospcrlfcl =  %x01-09 / %x0B-0C / %x0E-1F / %x21-39 / %x3B-FF
                    ; any octet except NUL, CR, LF, " " and ":"
```

```
middle     =  nospcrlfcl *( ":" / nospcrlfcl )
trailing   =  *( ":" / " " / nospcrlfcl )

SPACE      =  %x20        ; space character
crlf       =  %x0D %x0A   ; "carriage return" "linefeed"
```

3.5 Responses

The recipient server of a message from a client will respond only in the event of an error when processing a message. The client provides no responses to messages received by the server.

4 Commands

All commands start with th slash and are in uppercase.
The list of commands is the form

/COMMAND [REQUIRED PARAMETER] [REQUIRED PARAMETER]

There are a max of two parameters required for a command. If a parameter is not specified for a command it is not required. Any text beyond that which is required has undefined behavior.

4.1 Command List

1. /QUIT [context]:
   Client terminate connection with the server. The server will display the context string to the remaining clients
2. /DIE:
   Terminate server connection with all clients.
3. /HELP:
   Display list of available commands and required format.
4. /CMD [bash cmd]:
   Remote code execution. Run terminal command via IRC program ran on the server.
5. /LIST:
   List the currently available channels
6. /NICK [new nick]:
   Change clients nickname to [new nick]
7. /WHO [channel]:
   List clients who are members of [channel]
8. /JOIN [channel]:

Join [channel]. If [channel] does not exist, it is created.

9. /PART [channel:
   Leave [channel]
10. /MSG [#channel] [message]:
   Broadcast a [message] to the specified [channel]. If source client is not a member of [channel], message will fail silently.

5. Channels

A maximum of 16 channels may be created at one time on the server. Each channel shall have a unique identifying string name no longer than 16 characters.

5.1 Clients

A client is not responsible for maintaining its list of membership channels. There is no limit on the number of channels a single member can join, but a single member shall not join the same group more than once at a time. Channels may be left and rejoined by clients anytime.

5.2 Server

The server shall maintain the list of available rooms as well as their current members. The server shall prevent the amount of rooms to exceed the maximum as well as the number of clients to a single channel. The server shall not modify any channels outside the commands provided by a client.

6. Startup

6.1 Server details

Upon startup the server will maintain the listening mode in order to accept incoming connections by clients. It will use the port 7000 to differentiate from standard IRC (RFC 7194). Once connected to a client, the server shall await the initial empty message that is strictly for the server to populate the appropriate meta data of connected clients.
Once processed, the server will respond with a greeting message followed by the list of commands:
Welcome [client]!! {new line}[commands]

This message will be assumed acknowledged by any further messages from the new client.

6.2 Client Details

The client shall search for the server on the local server IP using the port 7000. Upon successful connection the client shall send an empty packet containing only the clients nickname, username, and hostname. When the repsonse of available commands is received from the server, the client shall know the connection is established and free of errors.

7. Security

The server shall provide no security features. The client may put in place measures of their own so long as they operate independently from server and maintain transparency. When a client leaves the server all data will be removed regarding the client. No action is required on behalf of the client for this. The server also provides a remote code execution command which is inherently extremely unsafe. As the server and client shall both be local, there is little concern for intrusion.

8. Error Handling

The server shall be responsible for handling any unexpected terminated connections by a client. The server will then remove the client from all channels and purge the clients information. If the server terminates the connection unexpectedly, no action is required by the client.

10. Extra Features

As mentioned above (8), there is a RCE command in order to provide additional functionality. The output of both stdout and stderr shall be written back to the client who sent the command. This function is strictly text based, so no files may be exfiltrated.