CS 594: Internetworking Protocols – FALL 2023

Internet Relay Chat Protocol

Team Members:

Vamsi Myla vmyla@pdx.edu
Srihari Tanmay Karthik Tadala stadala@pdx.edu
Raghuram Gudemaranahalli Nataraja raghuram@pdx.edu

Status of the memo:

This is a project for the Internetworking Protocol subject. messages used in this project are based on a thorough understanding of the course concepts. This document has been created for examination and review purposes, and we welcome any feedback to help us improve the model. We are committed to delivering a high-quality project and appreciate your support in making it even better.

Abstract:

Internet Chat Relay is a user-friendly and efficient client-server architecture that enables seamless network communication between clients and servers using socket programming. This text-based protocol allows clients to exchange private messages with ease, while multiple clients can connect to a server concurrently. With the ability to create, enter, exit, and broadcast messages to all other clients in a chat room, this protocol offers a highly interactive and engaging experience.

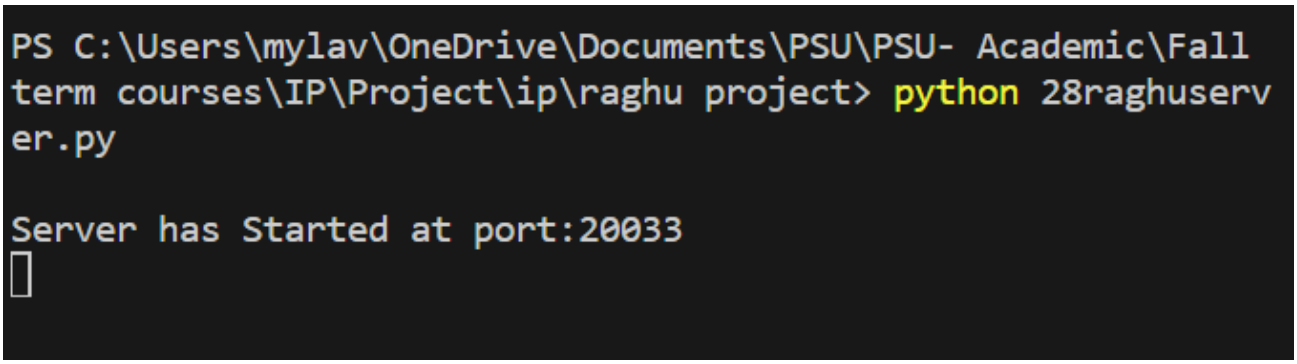
Table of Contents:

1. Introduction

Socket Programming enables server client connection. Once the server is up, the server listens for client connection. When the server receives a client request , it accepts the request. Multiple clients on the server can share relays of messages.

1.1 Servers

Socket Programming enables connection to multiple clients hosted on a specific port. The server address is a combination of the IP address and port address, which can be specified by the user or default to port 1024. Whenever a client connects to the specified port, it is listed on the server window. Also, when a room is created, the server keeps track of all commands that connected clients execute. To establish server connections, Socket Programming uses a bind method to add addresses, a listen method to read client requests, and an accept method to accept the connection request.



1.2 Clients

Any user who connects to the server is referred to as a client. Each client is given a unique name to distinguish them from other clients. Multiple or single threads can attempt to connect to a port by either specifying the same port as the server or, if nothing is specified, the port starts at 1024. Once the connection has been established, clients can provide their selected username, which is used to keep track of messages and threads.

1.3 Rooms

A client has the ability to create a room which can have multiple members. This feature allows clients to communicate in a group using the room. When a room is created, the client who initiated the creation process is automatically added to the room. Other clients can view the available rooms on the server and join any room they want. Moreover, clients also have the option to leave any room they are a part of. Additionally, they can view the members of the room they are currently in and broadcast messages to all the rooms they are a part of.

2. Conventions used in this document-

This communication system operates on a text-based client-server model where command prompt is used for all requests. Once a client is connected to the server, they are provided with a selection of functionalities that can be triggered using cmd. To execute a command, simply start with $command_name and add any necessary parameters. The following is a comprehensive list of commands and functionalities that clients are presented with upon a successful connection to the server.

```
                          IRC COMMANDS
Below are the list of commands
$listAll                            : To list all the available rooms and clients
$list room_name                     : To list members of a room
$create room_name                   : To create a new room
$join room_name                     : To join a room
$switch                             : To switch rooms
$leave                              : To leave current room
$private_message username message    : To send a direct personal message
$send_room room_name message        : To send a message to specified rooms
$broadcast_everyone message         : To broadcast message to everyone
$menu                               : To print menu
$bye                                : To exit
```

3. Basic Information

This system utilizes the TCP/IP three-way communication protocol for establishing connections. Clients connect to the server using the server's IP address and port number. Once connected, the client can send messages and requests to the server via the open channel and the server can respond via the same channel. All communication is conducted over TCP/IP, and the server listens for incoming connections on a designated port. Clients can connect to this port to maintain a persistent connection with the server. The client-server model is asynchronous, meaning the client sends requests to the server as needed and the server immediately processes them. This messaging protocol is also asynchronous in nature.

4. Messaging Infrastructure

A client can use the following command formats for sending messages:

```
                          IRC COMMANDS
Below are the list of commands
$listAll                            : To list all the available rooms and clients
$list room_name                     : To list members of a room
$create room_name                   : To create a new room
$join room_name                     : To join a room
$switch                             : To switch rooms
$leave                              : To leave current room
$private_message username message   : To send a direct personal message
$send_room room_name message        : To send a message to specified rooms
$broadcast_everyone message         : To broadcast message to everyone
$menu                               : To print menu
$bye                                : To exit
```

5. Client and Server Messages

5.1 Establishing the connection

In order to establish a connection, the host must initiate the server application and run it. Once an authentic connection is established, multiple clients can connect to the server. Clients who are connected have the ability to send broadcast messages to all other clients.

```
PS C:\Users\mylav\OneDrive\Documents\PSU\PSU- Academic\Fall term courses\IP\Project\ip\raghu pro
ject> python3 28raghuclient.py
Enter your first name: vamsi
Waiting for connection response
Hi vamsi! You have successfully connected to the server!
```

5.2 Creating Room

After the connection is established, the user can create a room by using the command below. The client who creates a room automatically becomes a member of that room and does not need to join it separately.

$create new_room_name

```
PS C:\Users\mylav\OneDrive\Documents\PSU\PSU-
Academic\Fall term courses\IP\Project\ip\raghu
 project> python 28raghuserver.py

Server has Started at port:20033
Vamsi

Client connected on ('127.0.0.1', 59222) with
username:VamsiThread /Client Number: 1
Vamsi has created room roomA
```

```
Enter your first name: Vamsi
Waiting for connection response
Hi Vamsi! You have successfully connected to the server!


                                IRC COMMANDS
Below are the list of commands
$listAll                          : To list all the available rooms and
clients
$list room_name                   : To list members of a room
$create room_name                 : To create a new room
$join room_name                   : To join a room
$switch                           : To switch rooms
$leave                            : To leave current room
$private_message username message : To send a direct personal message
$send_room room_name message      : To send a message to specified rooms

$broadcast_everyone message       : To broadcast message to everyone
$menu                             : To print menu
$bye                              : To exit

If you are in any room you can send message directly
Enter $command_name along with the paramerters mentioned.

$create roomA
roomA created
```

```
$create roomA
roomA created
```

5.3 Joining the Room

Once a room is created, additional clients can join the room by using the following command. Clients who are already members of the room will receive a confirmation message upon someone joining.

$join room_name

```
$join roomA
[roomA]  Karthik joined the room
```

The below figure shows confirmation to other users joins the room.

```
$join roomA
[roomA]  Karthik joined the room
```

```
$join roomA
[roomA]  Raghu joined the room
```

5.4 Switch room

Clients have the ability to switch from one room to another using the $switch command. This command is useful when a client needs to broadcast messages to a particular group or leave a group. If the client is already a member of the room they are attempting to switch to, they will receive a message indicating that they are already part of that room.

```
$switch roomB
Switched to roomB

$switch roomB
youve been switched successfully to the room
```

5.5 List Members of a room

Clients have the ability to view the list of members in a given room by using the $list command followed by the room name. This command will display the names of all rooms and the members currently present in each room.

```
Members of Room 'roomA':

Vamsi
Karthik
Raghu
```

5.6 Multiple clients can connect to a server:

```
Documents\PSU\PSU- Academic\Fall term courses\IP\Project\ip\raghu project> python 28raghuse
rver.py

Server has Started at port:20033
Vamsi

Client connected on ('127.0.0.1', 59259) with username:VamsiThread /Client Number: 1
Vamsi has created room roomA
Karthik

Client connected on ('127.0.0.1', 59277) with username:Karthik
Thread /Client Number: 2
Karthik has joined room roomA
Raghu

Client connected on ('127.0.0.1', 59279) with username:Raghu
Thread /Client Number: 3
Raghu has joined room roomA
```

5.7 List all members and room names:

Clients can view a list of all members connected to the server, as well as all available room names, by using the $listAll command. This functionality allows clients to easily join a room or message anyone on the server.

```
Client connected on ('127.0.0.1', 59279) with username
:Raghu
Thread /Client Number: 3
Raghu has joined room roomA
Vamsi has created room roomB
Karthik has joined room roomB
roomA
['Vamsi', 'Karthik', 'Raghu']
roomB
['Vamsi', 'Karthik']
```

```
$listAll

Rooms:

roomA
roomB



List of Users:

Vamsi
Karthik
Raghu
```

```
$listAll

Rooms:

roomA
roomB
roomC



List of Users:

Vamsi
Karthik
Raghu
```

5.8 Leave a Room

Clients have the option to leave a room by first switching to that specific room using the $switch command followed by the room name. Once in the room, clients can leave the room by using the $leave command.

```
$leave roomA
You left the room
```

5.9 Client can join multiple (selected) rooms -

Clients are able to join multiple rooms using $join room_name command

```
Client connected on ('127.0.0.1', 527
84) with username:raghu
Thread /Client Number: 3
mC
vamsi has joined room roomCroomA
['vamsi', 'karthik', 'raghu']
roomB
['karthik', 'vamsi', 'raghu']
roomC
['raghu', 'karthik', 'vamsi']
```

```
$create roomA
roomA created

[roomA]  karthik joined t
he room

$join roomB
[roomB]  vamsi joined the
 room

[roomB]  raghu joined the
 room

[roomA]  raghu joined the
 room

$join roomC
[roomC]  vamsi joined the
 room
```

```
$create roomB
roomB created

$join roomA
[roomA]  karthik join
ed the room

[roomB]  vamsi joined
 the room

[roomB]  raghu joined
 the room

[roomA]  raghu joined
 the room

$join roomC
[roomC]  karthik join
ed the room
```

```
$create roomC
roomC created

$join roomB
[roomB]  raghu joined the
room

$join roomA
[roomA]  raghu joined the
room

[roomC]  karthik joined th
e room

[roomC]  vamsi joined the
room
```

5.10 Send distinct message to multiple rooms

Clients can send individualized messages to different rooms by using the $broadcast_message command followed by the room name and message. This command sends the designated message to all members of the specified room. To send distinct messages to a specific room, use the $broadcast_message roomname message command.

```
$send_room roomA hello
[roomA]  [Vamsi] $send_room roomA hello
```

5.11 Send Private Message

This application allows for one-to-one communication where a client can send a message directly to another client without it being broadcasted to other members or rooms. To send a private message, the client can use the $private_message command followed by the intended recipient's name and the message. If the specified user does not exist, an error message will be displayed indicating that the user was not found.

```
$private_message Karthik Hi Karthik
[private message] Vamsi: Hi Karthik
```

```
[private message] Vamsi: Hi Karthik
```

```
$private_message Raghu Hello Raghu
[private message] Karthik: Hello Raghu
```

```
[private message] Karthik: Hello Raghu
```

5.12 Send a broadcast message to all the clients

A broadcast message is sent to all clients currently connected to the server. This message is received by all active clients, regardless of whether they are in a particular room or not. All clients who receive the message will know which client delivered it.

$broadcast_everyone message

```
$broadcast_everyone Hello all
[Broadcast message from Vamsi] Hello all
```

5.13 Menu

Clients have the ability to view a list of available commands at any time by using the $menu command on the command prompt.

```
$menu
                            IRC COMMANDS
Below are the list of commands
$listAll                            : To list all the available rooms and clients
$list room_name                     : To list members of a room
$create room_name                   : To create a new room
$join room_name                     : To join a room
$switch                             : To switch rooms
$leave                              : To leave current room
$private_message username message    : To send a direct personal message
$send_room room_name message        : To send a message to specified rooms
$broadcast_everyone message         : To broadcast message to everyone
$menu                               : To print menu
$bye                                : To exit
```

5.14 Disconnect from server

Clients have the option to disconnect from the server by using the $bye command. Once the client has disconnected, the server will display a message indicating that the client has left the room. Clients who have disconnected will no longer receive any further communication from the server or other clients connected to that server. Additionally, the room members who were in the room with the disconnected client will receive a message notifying them that the client has left the room.

6. Error Handling:

All negative cases are handled using error handling. If a user encounters an error, an exception will be thrown. Below are a few examples of when error handling is utilized: -

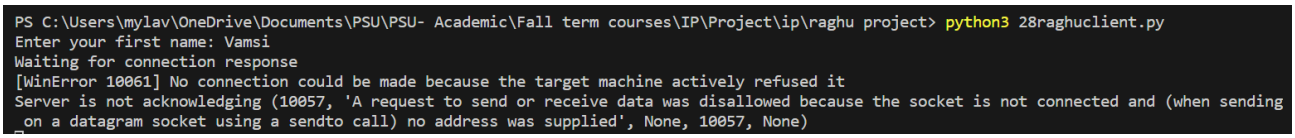
When a user attempts to switch to a room they are not a member of.

```
$switch roomA
Members only authorized, in order to switch the room please join the room
```

If the client is trying to send message to user that doesn't exists, user gets user not found error

```
$private_message unknown_user Hello
User not found
```

If the server is not available to connect



7. Additional Functionalities

Under extra credits we have implemented sending a private message to clients.

8. Security Considerations

The purpose of this standard is to establish a fundamental framework for communication between multiple clients through a centralized forwarding server and TCP. The standard defines a text-based communication scheme that clients can use to create their own protocols without altering the specification. This document presents the basic specification of the Internet Relay Chat (IRC) Project, which is a framework designed to enable multiple clients to interact with each other via a server, tailored to the specific requirements of the application.

9. Conclusion

The current implementation of the IRC protocol has several limitations. Unlike the RFC 1459 standard, this protocol does not employ multiple servers operating concurrently. Additionally, the private messaging feature could be improved to ensure that encrypted messages are only viewable by the client end. In the future, there is potential for the development of login credentials for joining a channel, as well as file or media sharing capabilities.

10. Future work:

The IRC protocol has several limitations. Unlike the RFC 1459 standard, it does not make use of multiple servers operating concurrently. Additionally, the private messaging feature of this protocol could be improved to ensure that only the client end can access encrypted messages. In the future, there is potential for the development of login credentials for joining channels, as well as the inclusion of file or media sharing functionalities.

11.Reference

Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

https://datatracker.ietf.org/doc/html/rfc2119
https://en.wikipedia.org/wiki/Internet_Relay_Chat

12. Acknowledge

For these project deliverables, we used Google resources and the Google product suite. For the design, we also researched Wikipedia and other sources.