

ChatRoom in C

OverView

ChatRoom in C with Socket Programming using threads and using select system call. Both Threads and Select system call allows to handle multiple clients in socket programming simultaneously. Private messages, group chats and file transfer are supported. Messages and files transfer are end to end encrypted. No one outside of the chat, not even server, can read or listen to them. Check the [Features](#) section for more.

Table of Contents

- [Overview](#)
- [Using Threads](#)
- [Using Select](#)
- [Features](#)
- [Commands Supported](#)
- [Developer Protocols followed](#)
- [Github Link](#)
- [Contributors](#)

Using Threads

serverThread.c and clientThread.c uses threads to achieve Client Server Architecture.

Command to start serverThread.c

```
gcc serverThread.c -o serverThread
./serverThread <port_no>
```

Command to start clientThread.c

```
gcc clientThread.c -o clientThread
./clientThread <IP_Address> <port_no> <username>
```

Using Select

server.c and client.c uses select() system call to achieve Client Server Architecture.

Command to start server.c

```
gcc server.c -o server
./server <port_no>
```

Command to start client.c

```
gcc client.c -o client
./client <IP_Address> <port_no> <username>
```

Features

- Handling Multiple Client
- **Unique** Client can join and disconnect
- Private Message between two Clients
- Broadcasting to all Alive Clients
- File transfer between two Clients
- Reporting a Client using username
- Client removal power to server
- Messages and files are **end-to-end encrypted**
- Time-out after 60 seconds (subjected to change)
- Group chats supported
- Features of group chats:
 - Create a group
 - Join a group
 - Leave a group
- Groups with no members automatically gets deleted
- Group removal power to server

Commands Supported

Client side

- <message> - for direct broadcasting the message to all client
- @<username> <message> - enter the username of client to send the private message
- @<username> @file <fileName> - to send file fileName
- #<username> - to report a sus client with username

- `EXIT` - to disconnect from the chat server

Group Chat

- `$CREATE <groupName>` - to create a group
- `$JOIN <groupName>` - to join a group
- `$LEAVE <groupName>` - to leave a group
- `$<groupName> <message>` - to send message in the group
- `$<groupName> @file <fileName>` - to send file `fileName` in the group

Server Side

- `REMOVE <username>` - to kick out client with username
- `DELETE <groupName>` - to delete the group with `groupName`
- `CLOSE` - to close the server

Developer Protocols followed

- response from server to client should be of format `>> <server_message>...`

Github Link

GITHUB LINK OF THE PROJECT (<https://github.com/pntu007/ChatRoom/tree/main>)

Contributors

- Saurav Kumar Singh (22CS01010)
- Suprit Naik (22CS01018)
- Om Prakash Behera (22CS01041)
- Harsh Maurya (22CS01046)