COMP 4621 Project Report

Server Based Multiparty chatroom with online and offline messages

Table of Contents

Introduction

Project Requirements

The project required me to implement a server-client based multiparty chatroom with online and offline messages. I have implemented this project using socket programming in C and the POSIX library. The POSIX library was required to implement concurrency such that the client could send and receive messages at the same time.

Other specific requirements include :

- Programming must be in C
- You must use poll() system call to handle multiple clients concurrently, instead of select() or epoll(). Otherwise, you will get no points.
- Each client must supply a unique nickname for registration and login. If the nickname supplied already exists, we regard it as a login case. Otherwise, regard it as a registration case.
- Clients must be able to send private messages to other online clients, leave private messages to offline clients as well as send public broadcast messages.
- Clients should be able to send and receive messages simultaneously.
- Both the client and the server must handle exceptions and errors accordingly

Project Scope

Since the chatroom was based on a client server model, the server acts as a proxy between different clients sending messages to each other. The server can also handle different commands, and responds to them appropriately

Some commands are as follows :

- **REGISTER** <nickname>: is sent in response to the server prompting the user to supply a nickname for registration or login.
- **WHO**: can be sent by the client to the server to obtain a tab-separated list of nicknames of all the users in the system (except itself), where the state of each user is also labelled.
- **EXIT**: can be sent by the client to the server to leave the chatroom

The clients could send two kinds of messages to each other:

- **Broadcast message** : The client sends a message to all online users in the chatroom
- **Direct Message** : The client sends a message to another client directly i.e. in private, only the recipient of the message is notified in this scenario. There can be two cases for direct messages :
    - Online user : The message is forwarded to the recipient in real time and displayed as a private message from the sender on the recipients screen
    - Offline user : The message is stored in the recipient's message box. The recipient can view all the offline messages when he/she is online again.

Implementation

Client Side Application

I started the project by programming the client side application as it did not involve supporting a lot of features. The skeleton code was provided to me as a starting point for the project.

I started off by creating a client socket and connecting it to the server. I used the socket() system call in C for the same. I referred to previous lab codes to implement the same. After creating the client socket and connecting it to the server, I connected the client socket to the server socket using the connect() system call. For a better user experience, I have also added error messages if something went wrong in this process based on the return values from the system calls.

According to the instructions in the skeleton code, I had to prepend REGISTER to the buffer when the client sends the first message to the server. This is used to handle the user registration and login feature in the chatroom. I used a character array for the same with the strcat() feature to concatenate the character array to the buffer before sending it to the server. My code segment for the same is shown below.

```
/*************************************/
/* Input the nickname and send a message to the server */
/* Note that we concatenate "REGISTER" before the name to notify the server it is the register/login message*/
/*****************************************/

bzero(buffer, sizeof(buffer));
printf("Enter the string: ");
n = 0;
while ((buffer[n++] = getchar()) != '\n')
    ;
char reg[8] = {'R','E','G','I','S','T','E','R'};
char *tmp = strdup(buffer);

strcpy(buffer, reg);
strcat(buffer, tmp);
free(tmp);

send(sockfd, buffer, strlen(buffer), 0);
```

To enable the feature of simultaneous sending and receiving I had to create another thread to receive messages from the server that runs concurrently with the thread that is sending messages to the server. The following code segment shows how I used the pthread_create() function from the POSIX thread library to create and run another thread concurrently that receives messages from the server.

```
/*****************************************************/
/* Create a thread to receive message from the server*/
/* pthread_t recv_server_msg_thread;*/
/*****************************************************/
pthread_t recv_server_msg_thread;
int* pclient = malloc(sizeof(int));
*pclient = sockfd;

if (pthread_create(&recv_server_msg_thread, NULL, recv_server_msg_handler, (int *) pclient) != 0)
{
    perror("pthread_create() error");
    exit(1);
}
```

```
void * recv_server_msg_handler(int* p_client_socket) {
    /*********************************/
    /* receive message from the server and desplay on the screen*/
    /*********************************/
    // printf("DEBUG - Inside recv_server_msg_handler\n");
    int sockfd = *p_client_socket;
    // printf("DEBUG - sockfd : %d\n", sockfd);

    char buffer[MAX];
    for (;;)
    {
        bzero(buffer, sizeof(buffer));
        if (recv(sockfd, buffer, sizeof(buffer), 0) > 0)
        {
            // printf("DEBUG : %d\n", strcmp(buffer, "Wrong Password. Try again"));
            printf("> %s", buffer);
        }
        else
        {
            break;
        }

    }
}
```

I added a “>” character when we display incoming messages from the server for a better user experience. It is clear to the user when we receive a message from the server as compared to when we are writing messages in the buffer.

The rest of the client side application was an infinite loop that identified the different types of commands or messages and sent them to the server. Please find the code for the same below. The EXIT command exits the created thread and closes the client side program.

```c
// chat with the server
for (;;) {
    bzero(buffer, sizeof(buffer));
    n = 0;
    while ((buffer[n++] = getchar()) != '\n')
        ;
    if ((strncmp(buffer, "EXIT", 4)) == 0) {
        printf("Client Exit...\n");
        /*****************************************/
        /* Send exit message to the server and exit */
        /* Remember to terminate the thread and close the socket */
        /*****************************************/
        strcpy(buffer, "EXIT");
        send(sockfd, buffer, strlen(buffer), 0);
        pthread_exit(NULL);
        close(sockfd);
        break;
        // ============================================================================================
    }
    else if (strncmp(buffer, "WHO", 3) == 0) {
        printf("Getting user list, pls hold on...\n");
        if (send(sockfd, buffer, sizeof(buffer), 0)<0){
            puts("Sending MSG_WHO failed");
            exit(1);
        }
        printf("If you want to send a message to one of the users, pls send with the format: '#username:message'\n");
    }
    else if (strncmp(buffer, "#", 1) == 0) {
        // If the user want to send a direct message to another user, e.g., aa wants to send direct message "Hello" to bb, aa needs to input "#bb:Hello"
        if (send(sockfd, buffer, sizeof(buffer), 0)<0){
            printf("Sending direct message failed...");
            exit(1);
        }
    }
    else {
        /*************************************/
        /* Sending broadcast message. The send message should be of the format "username: message"*/
        /*************************************/
        // printf("DEBUG - name  : %s\n", name);

        if (send(sockfd, buffer, sizeof(buffer), 0)<0){
            printf("Sending broadcast message failed...");
            exit(1);
        }

    }
}
```

Server Side Application

I started coding the server side code by implementing the helper functions that were mentioned in the skeleton code.
Since we had to read and write user information and maintain their state, I used a mutex lock to implement mutual exclusivity and not face the reader's writer's problem.

Please find below my code for the user_add() helper function.

```
/* Add user to userList */
void user_add(user_info_t *user){

    pthread_mutex_lock(&clients_mutex); //********************

    if(users_count ==  MAX_USERS){
        printf("sorry the system is full, please try again later\n");
        return;
    }
    /***************************/
    /* add the user to the list */
    /***************************/
    else
    {
        listOfUsers[users_count] = user;
        users_count++;
    }

    pthread_mutex_unlock(&clients_mutex); //********************

}
```

I used a linear search algorithm for the get_username() and get_sockfd() helper functions. The code for the functions are shown below.

After implementing the helper functions, I created a listener socket to accept incoming TCP connections and I binded it with a port number.
In the main loop in the server side code, I used the poll() function to identify file descriptors where we are looking for data to read. The two helper functions provided add_to_pdfs() and del_from_pdfs() helped in keeping track of the state of the file descriptors of all the active connections.

There are two cases when a data is read from the poll() function:

1. It is a new connection at the listener
2. It is not the listener socket and it is just a regular client sending data

Case 1 : Event on listener socket i.e. handling a new client connection

```
if (pfds[i].fd == listener)
{
  /***************************/
  /* we are the listener and we need to handle new connections from clients */
  /***************************/
    addr_size = sizeof(client_addr);
    newfd = accept(listener, (struct sockaddr*)& client_addr, &addr_size);
    printf("DEBUG - newfd : %d\n", newfd);
    // send welcome message
    if (newfd == -1)
    {
        perror("accept");
    }
    else
    {
        add_to_pfds(&pfds, newfd, &fd_count, &fd_size);
        printf("pollserver: new connection from %s on socket %d\n",inet_ntoa(client_addr.sin_addr),newfd);
    }
    bzero(buffer, sizeof(buffer));
    strcpy(buffer, "Welcome to the chat room!\nPlease enter a nickname.\n");

    if (send(newfd, buffer, sizeof(buffer), 0) == -1)
    {
        perror("send");
    }
}
```

I accepted new TCP connections at the server side using the accept() system call; the new file descriptor of the socket where the client's connection has been accepted is stored in the newfd variable. I am adding the new file descriptor of the incoming connection to the pdfs list to keep track of the client file descriptors and sending a welcome message to the incoming connection's file descriptor/socket.

Case 2 : Event not on listener socket i.e. Handling a client sending data

I started off this case by writing code to handle errors from the clients. If there is an error such as if a client quit unexpectedly, I wrote a log message to print out the error, close the socket and delete the client from the pdfs list.

```
// handle data from a client
bzero(buffer, sizeof(buffer));
if ((nbytes = recv(pfds[i].fd, buffer, sizeof(buffer), 0)) <= 0) {
  // got error or connection closed by client
  if (nbytes == 0) {
    // connection closed
    printf("pollserver: socket %d hung up\n", pfds[i].fd);
  } else {
    perror("recv");
  }
  close(pfds[i].fd); // Bye!
  del_from_pfds(pfds, i, &fd_count);
```

If there is no error from the data read from the socket, the data is legitimate data from the client that needs to be processed.

There are different cases for incoming client data that needs to be handled by the server, they are listed as follows:

1. REGISTER/LOGIN case: We need to handle a new user registration or login of an existing user in the chatroom. The procedure for each case is listed down as follows:
    a. Register Case :
        i. Create a new user
        ii. Store new user's socket file descriptor, username and state
        iii. Create a message box for offline messages (a .txt file named after the username of the new user)
        iv. Broadcast welcome message to online users (iterative process using for loop)
    b. Login Case :
        i. Update the user's state
        ii. Read the user's message box (text file based on username)
        iii. Send offline messages to the user
        iv. Update user's state
        v. Broadcast welcome back message to online users (iterative process using for loop)

2. EXIT Case: We need to handle a disconnection from the client, i.e. handle the case when the client has left the chatroom. The procedure for handling this case is listed below:
        i. Get the username from the socket file descriptor
        ii. Broadcast the leave message to online users in the chatroom
        iii. Change the state of the user to offline
        iv. Close the socket and remove it from the pfds list

3. WHO Case: Case where we need to send a list of users in the chatroom and their state information (online or offline). The procedure for handling the WHO case is shown below:
        i. Get the socket descriptor of the incoming request (pfds[i].fd)
        ii. Get the username from the socket descriptor
        iii. Go through all users in list of user's, concatenate their usernames in a string, mark "*" with the username if the state of the user is online
        iv. Send the string of concatenated usernames to the client

4. Direct Message Case : Case where one user sends a message directly to another user, this message is handled in private between the two users and is not visible to other people in the chatroom. The procedure for the same is shown below:

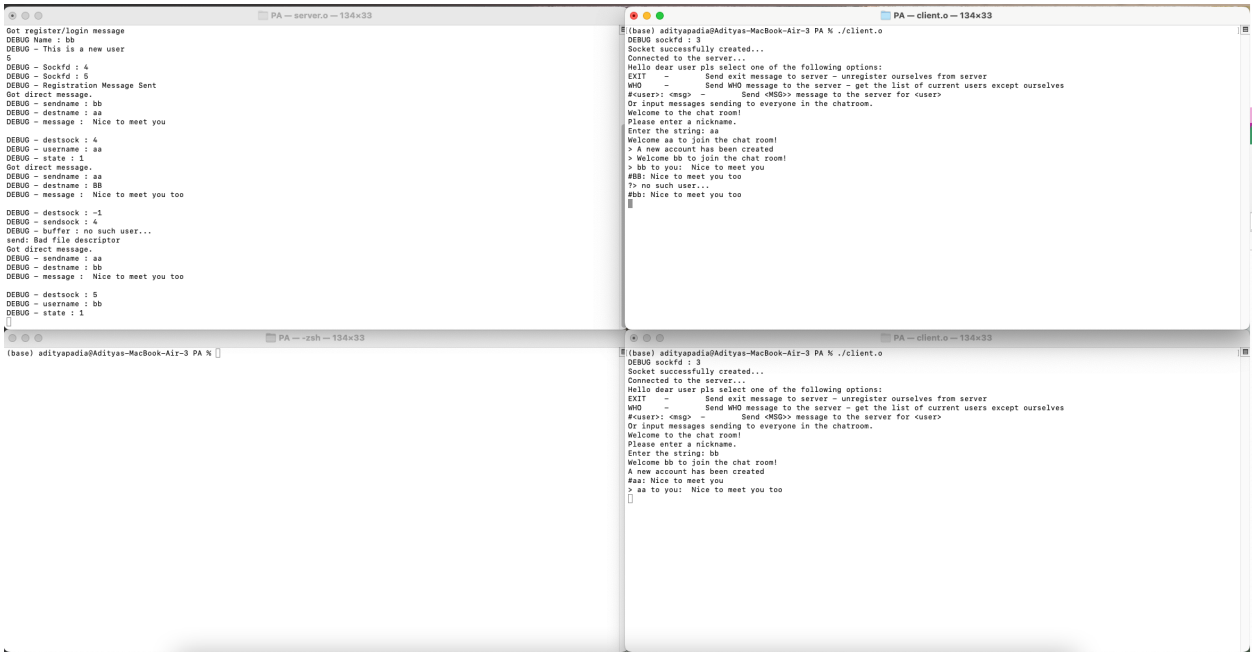
i. Get the username of the sender using the get_username() function with pfds[i].fd
ii. Get the recipient's username by scanning the buffer up to the ":" character
iii. Get the recipient's socket file descriptor using the get_sockfd() function with the recipient's username
iv. Handle error if target user is not found send error message to sender
v. If target user is found, append the sender information and send it to the recipient
vi. If the recipient is offline, i.e. the state is set to 0, then write the direct message into the recipient's mailbox or text file using the fprintf() function

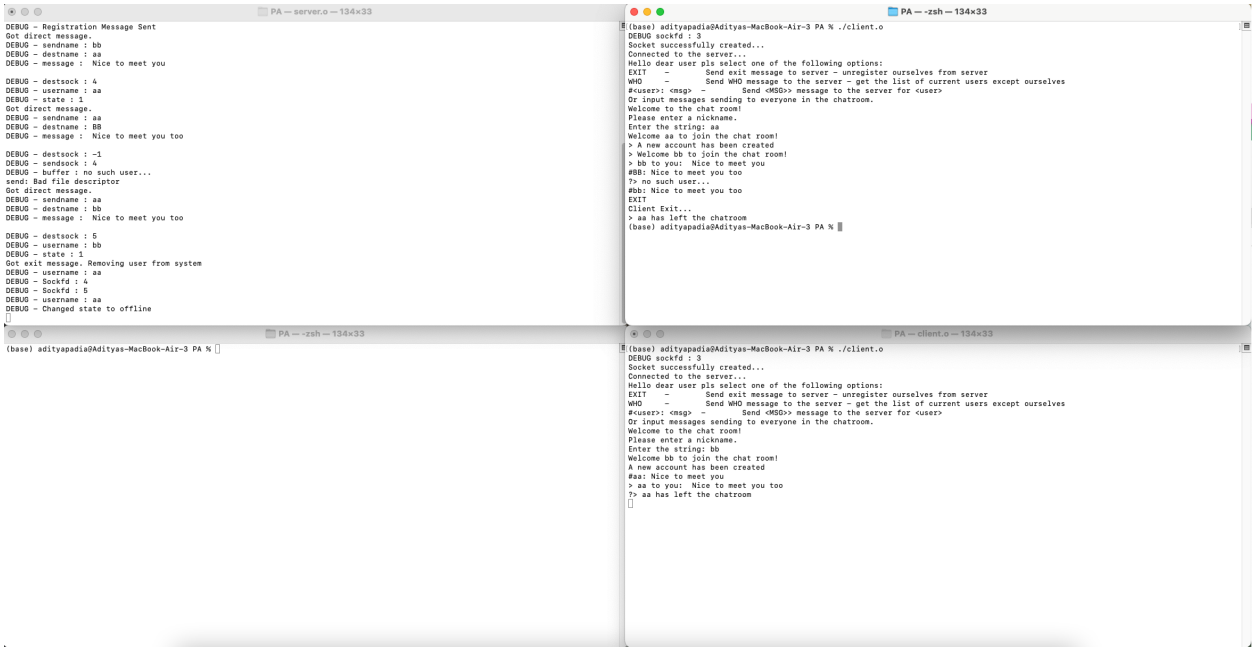
5. Broadcast Message Case : Case when one user sends a message to the entire chatroom, i.e. all the online user's receive the message immediately, offline messages are not written in this case. The procedure for handling the broadcast message case is shown below:
    i. Get the username from the socket file descriptor using the get_sockfd() function with pfds[i].fd
    ii. Extract the message using string functions and store it in the buffer
    iii. Iterate through all the users, if the user is not the listener socket or if the user is not the sender then send the buffer/message to all users

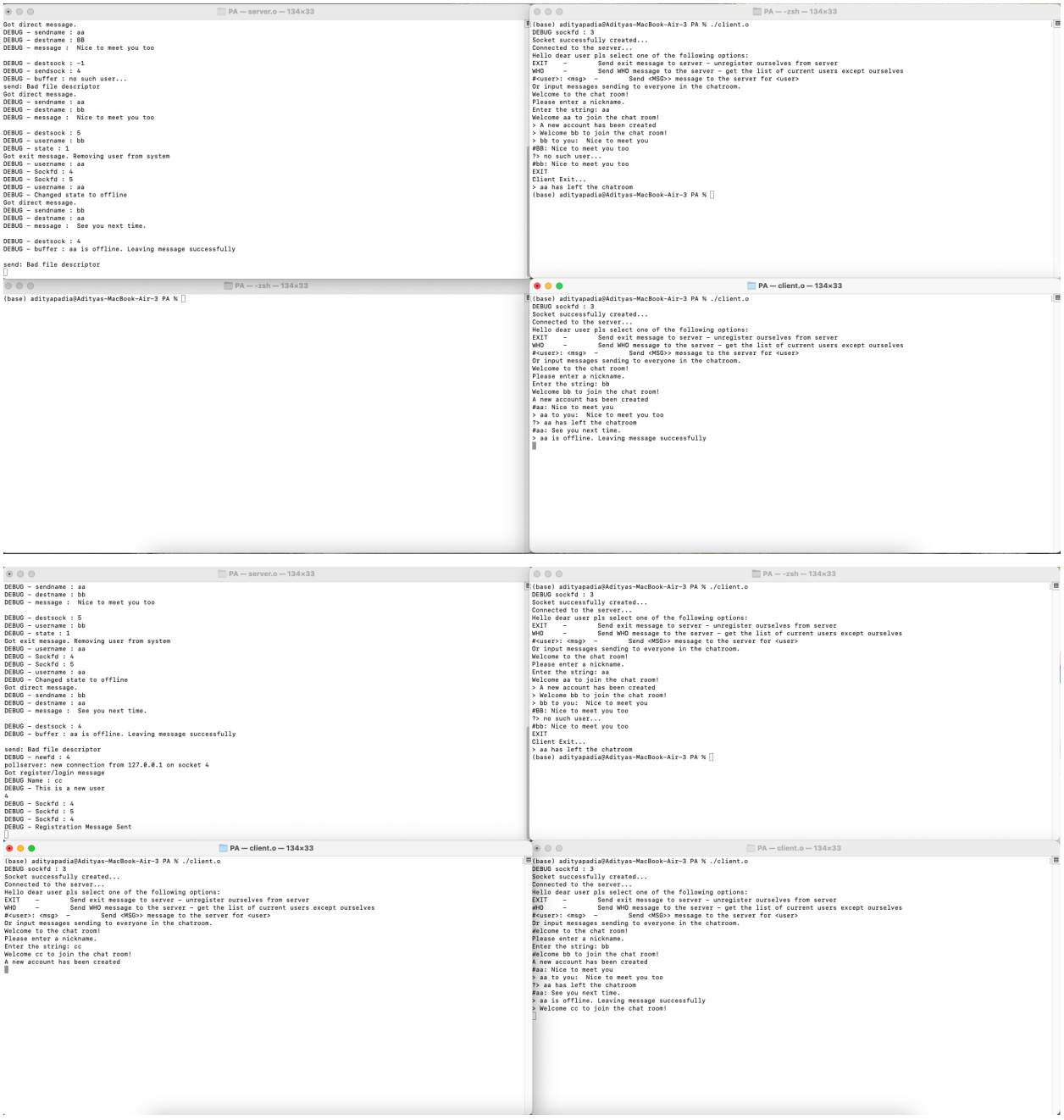
Screenshots of Test Cases

Please find below the screenshots of the test cases provided in the marking scheme document. For reference, top left is the server application, top right is the first client (aa), bottom right is the second client (bb) and bottom left is the third client (cc)

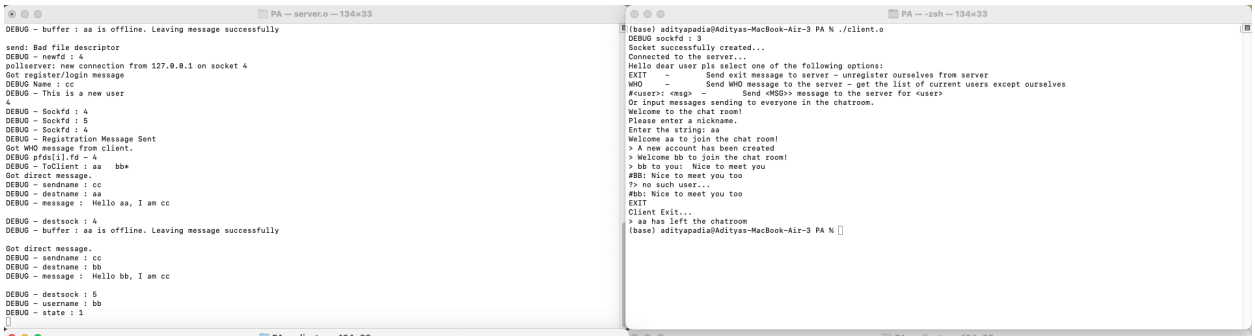

PA — server.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./server.o
Socket successfully created..
Socket successfully binded..
Server listening..
DEBUG - newfd : 4
pollserver: new connection from 127.0.0.1 on socket 4
Got register/login message
DEBUG Name : aa
DEBUG - This is a new user
4
DEBUG - Sockfd : 4
DEBUG - Registration Message Sent
DEBUG - newfd : 5
pollserver: new connection from 127.0.0.1 on socket 5
Got register/login message
DEBUG Name : bb
DEBUG - This is a new user
5
DEBUG - Sockfd : 4
DEBUG - Sockfd : 5
DEBUG - Registration Message Sent
Got direct message.
DEBUG - sendname : bb
DEBUG - destname : aa
DEBUG - message : Nice to meet you
DEBUG - destsock : 4
DEBUG - username : aa
DEBUG - state : 1
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT - Send exit message to server - unregister ourselves from server
WHO - Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> - Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: aa
Welcome aa to join the chat room!
> A new account has been created
> Welcome bb to join the chat room!
> bb to you: Nice to meet you
PA — -zsh — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA %
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT - Send exit message to server - unregister ourselves from server
WHO - Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> - Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: bb
Welcome bb to join the chat room!
A new account has been created
#aa: Nice to meet you
PA — server.o — 134×33
Got register/login message
DEBUG Name : aa
DEBUG - This is a new user
4
DEBUG - Sockfd : 4
DEBUG - Registration Message Sent
DEBUG - newfd : 5
pollserver: new connection from 127.0.0.1 on socket 5
Got register/login message
DEBUG Name : bb
DEBUG - This is a new user
5
DEBUG - Sockfd : 4
DEBUG - Sockfd : 5
DEBUG - Registration Message Sent
Got direct message.
DEBUG - sendname : bb
DEBUG - destname : aa
DEBUG - message : Nice to meet you
DEBUG - destsock : 4
DEBUG - username : aa
DEBUG - state : 1
Got direct message.
DEBUG - sendname : aa
DEBUG - destname : BB
DEBUG - message : Nice to meet you too
DEBUG - destsock : -1
DEBUG - sendsock : 4
DEBUG - buffer : no such user...
send: Bad file descriptor
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT - Send exit message to server - unregister ourselves from server
WHO - Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> - Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: aa
Welcome aa to join the chat room!
> A new account has been created
> Welcome bb to join the chat room!
> bb to you: Nice to meet you
#BB: Nice to meet you too
?> no such user...
PA — -zsh — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA %
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT - Send exit message to server - unregister ourselves from server
WHO - Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> - Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: bb
Welcome bb to join the chat room!
A new account has been created
#aa: Nice to meet you

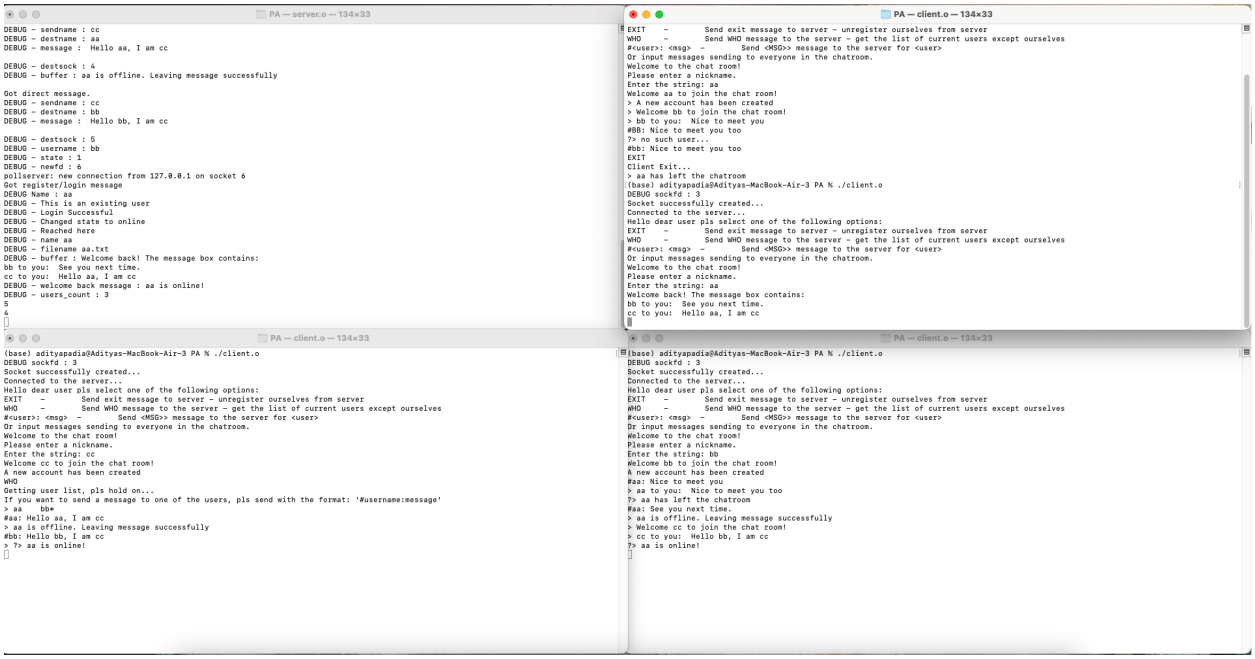

PA — server.o — 134×33
Got register/login message
DEBUG Name : bb
DEBUG – This is a new user
5
DEBUG – Sockfd : 4
DEBUG – Sockfd : 5
DEBUG – Registration Message Sent
Got direct message.
DEBUG – sendname : bb
DEBUG – destname : aa
DEBUG – message : Nice to meet you
DEBUG – destsock : 4
DEBUG – username : aa
DEBUG – state : 1
Got direct message.
DEBUG – sendname : aa
DEBUG – destname : BB
DEBUG – message : Nice to meet you too
DEBUG – destsock : -1
DEBUG – sendsock : 4
DEBUG – buffer : no such user...
send: Bad file descriptor
Got direct message.
DEBUG – sendname : aa
DEBUG – destname : bb
DEBUG – message : Nice to meet you too
DEBUG – destsock : 5
DEBUG – username : bb
DEBUG – state : 1
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created..
Connected to the server...
Hello dear user pls select one of the following options:
EXIT – Send exit message to server - unregister ourselves from server
WHO – Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> – Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: aa
Welcome aa to join the chat room!
> A new account has been created
> Welcome bb to join the chat room!
> bb to you: Nice to meet you
#BB: Nice to meet you too
?> no such user...
#bb: Nice to meet you too
PA — -zsh — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA %
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created..
Connected to the server...
Hello dear user pls select one of the following options:
EXIT – Send exit message to server - unregister ourselves from server
WHO – Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> – Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: bb
Welcome bb to join the chat room!
A new account has been created
#aa: Nice to meet you
> aa to you: Nice to meet you too

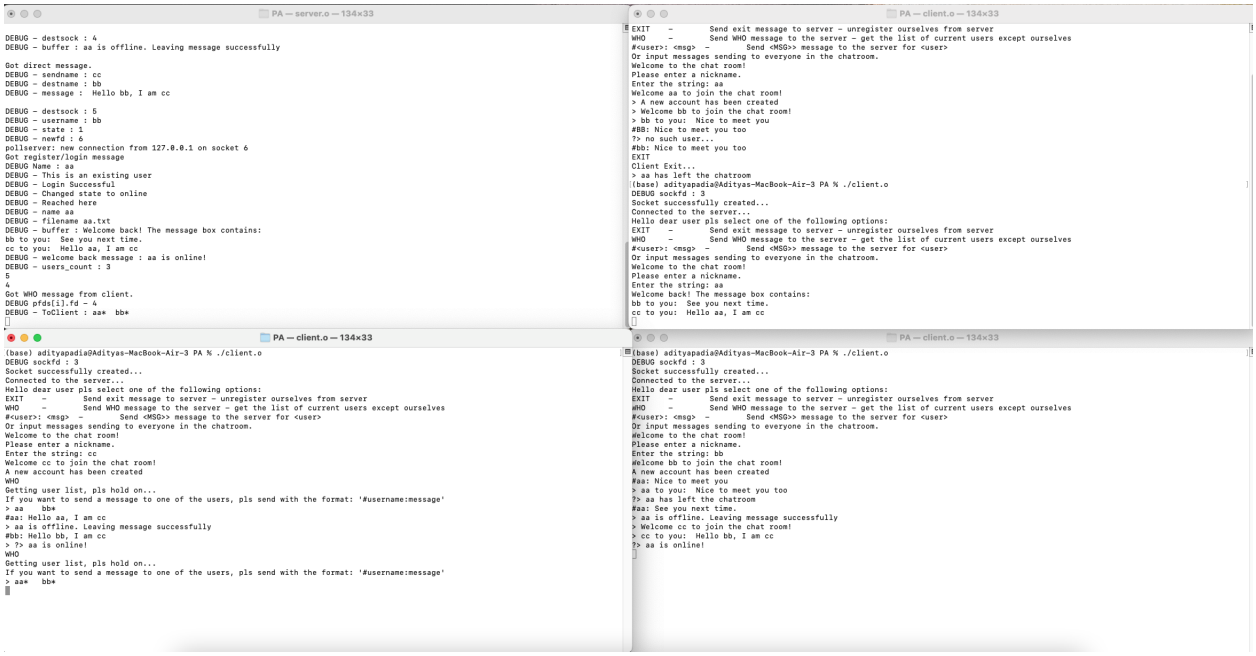

PA — server.o — 134×33
DEBUG – Registration Message Sent
Got direct message.
DEBUG – sendname : bb
DEBUG – destname : aa
DEBUG – message : Nice to meet you
DEBUG – destsock : 4
DEBUG – username : aa
DEBUG – state : 1
Got direct message.
DEBUG – sendname : aa
DEBUG – destname : BB
DEBUG – message : Nice to meet you too
DEBUG – destsock : -1
DEBUG – sendsock : 4
DEBUG – buffer : no such user...
send: Bad file descriptor
Got direct message.
DEBUG – sendname : aa
DEBUG – destname : bb
DEBUG – message : Nice to meet you too
DEBUG – destsock : 5
DEBUG – username : bb
DEBUG – state : 1
Got exit message. Removing user from system
DEBUG – username : aa
DEBUG – Sockfd : 4
DEBUG – Sockfd : 5
DEBUG – username : aa
DEBUG – Changed state to offline
PA — -zsh — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created..
Connected to the server...
Hello dear user pls select one of the following options:
EXIT – Send exit message to server - unregister ourselves from server
WHO – Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> – Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: aa
Welcome aa to join the chat room!
> A new account has been created
> Welcome bb to join the chat room!
> bb to you: Nice to meet you
#BB: Nice to meet you too
?> no such user...
#bb: Nice to meet you too
EXIT
Client Exit...
> aa has left the chatroom
(base) adityapadia@Adityas-MacBook-Air-3 PA %
PA — -zsh — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA %
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created..
Connected to the server...
Hello dear user pls select one of the following options:
EXIT – Send exit message to server - unregister ourselves from server
WHO – Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> – Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: bb
Welcome bb to join the chat room!
A new account has been created
#aa: Nice to meet you
> aa to you: Nice to meet you too
?> aa has left the chatroom


PA — server.o — 134×33
Got direct message.
DEBUG - sendname : aa
DEBUG - destname : BB
DEBUG - message : Nice to meet you too
DEBUG - destsock : -1
DEBUG - sendsock : 4
DEBUG - buffer : no such user...
send: Bad file descriptor
Got direct message.
DEBUG - sendname : aa
DEBUG - destname : bb
DEBUG - message : Nice to meet you too
DEBUG - destsock : 5
DEBUG - username : bb
DEBUG - state : 1
Got exit message. Removing user from system
DEBUG - username : aa
DEBUG - Sockfd : 4
DEBUG - Sockfd : 5
DEBUG - username : aa
DEBUG - Changed state to offline
Got direct message.
DEBUG - sendname : bb
DEBUG - destname : aa
DEBUG - message : See you next time.
DEBUG - destsock : 4
DEBUG - buffer : aa is offline. Leaving message successfully
send: Bad file descriptor
PA — -zsh — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA %
PA — -zsh — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT - Send exit message to server - unregister ourselves from server
WHO - Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> - Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: aa
Welcome aa to join the chat room!
> A new account has been created
> Welcome bb to join the chat room!
> bb to you: Nice to meet you
#BB: Nice to meet you too
?> no such user...
#bb: Nice to meet you too
EXIT
Client Exit...
> aa has left the chatroom
(base) adityapadia@Adityas-MacBook-Air-3 PA %
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT - Send exit message to server - unregister ourselves from server
WHO - Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> - Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: bb
Welcome bb to join the chat room!
A new account has been created
#aa: Nice to meet you
> aa to you: Nice to meet you too
?> aa has left the chatroom
#aa: See you next time.
> aa is offline. Leaving message successfully
PA — server.o — 134×33
DEBUG - sendname : aa
DEBUG - destname : bb
DEBUG - message : Nice to meet you too
DEBUG - destsock : 5
DEBUG - username : bb
DEBUG - state : 1
Got exit message. Removing user from system
DEBUG - username : aa
DEBUG - Sockfd : 4
DEBUG - Sockfd : 5
DEBUG - username : aa
DEBUG - Changed state to offline
Got direct message.
DEBUG - sendname : bb
DEBUG - destname : aa
DEBUG - message : See you next time.
DEBUG - destsock : 4
DEBUG - buffer : aa is offline. Leaving message successfully
send: Bad file descriptor
DEBUG - newfd : 4
pollserver: new connection from 127.0.0.1 on socket 4
Got register/login message
DEBUG Name : cc
DEBUG - This is a new user
4
DEBUG - Sockfd : 4
DEBUG - Sockfd : 5
DEBUG - Sockfd : 4
DEBUG - Registration Message Sent
PA — -zsh — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT - Send exit message to server - unregister ourselves from server
WHO - Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> - Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: aa
Welcome aa to join the chat room!
> A new account has been created
> Welcome bb to join the chat room!
> bb to you: Nice to meet you
#BB: Nice to meet you too
?> no such user...
#bb: Nice to meet you too
EXIT
Client Exit...
> aa has left the chatroom
(base) adityapadia@Adityas-MacBook-Air-3 PA %
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT - Send exit message to server - unregister ourselves from server
WHO - Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> - Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: cc
Welcome cc to join the chat room!
A new account has been created
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT - Send exit message to server - unregister ourselves from server
WHO - Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> - Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: bb
Welcome bb to join the chat room!
A new account has been created
#aa: Nice to meet you
> aa to you: Nice to meet you too
?> aa has left the chatroom
#aa: See you next time.
> aa is offline. Leaving message successfully
> Welcome cc to join the chat room!


PA — server.o — 134×33
DEBUG - destsock : 5
DEBUG - username : bb
DEBUG - state : 1
Got exit message. Removing user from system
DEBUG - username : aa
DEBUG - Sockfd : 4
DEBUG - Sockfd : 5
DEBUG - username : aa
DEBUG - Changed state to offline
Got direct message.
DEBUG - sendname : bb
DEBUG - destname : aa
DEBUG - message : See you next time.
DEBUG - destsock : 4
DEBUG - buffer : aa is offline. Leaving message successfully
send: Bad file descriptor
DEBUG - newfd : 4
pollserver: new connection from 127.0.0.1 on socket 4
Got register/login message
DEBUG Name : cc
DEBUG - This is a new user
4
DEBUG - Sockfd : 4
DEBUG - Sockfd : 5
DEBUG - Sockfd : 4
DEBUG - Registration Message Sent
Got WHO message from client.
DEBUG pfds[i].fd - 4
DEBUG - ToClient : aa bb*
PA — -zsh — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT - Send exit message to server - unregister ourselves from server
WHO - Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> - Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: aa
Welcome aa to join the chat room!
> A new account has been created
> Welcome bb to join the chat room!
> bb to you: Nice to meet you
#BB: Nice to meet you too
?> no such user...
#bb: Nice to meet you too
EXIT
Client Exit...
> aa has left the chatroom
(base) adityapadia@Adityas-MacBook-Air-3 PA %
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT - Send exit message to server - unregister ourselves from server
WHO - Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> - Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: cc
Welcome cc to join the chat room!
A new account has been created
WHO
Getting user list, pls hold on...
If you want to send a message to one of the users, pls send with the format: '#username:message'
> aa bb*
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT - Send exit message to server - unregister ourselves from server
WHO - Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> - Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: bb
Welcome bb to join the chat room!
A new account has been created
#aa: Nice to meet you
> aa to you: Nice to meet you too
?> aa has left the chatroom
#aa: See you next time.
> aa is offline. Leaving message successfully
> Welcome cc to join the chat room!
PA — server.o — 134×33
DEBUG - username : aa
DEBUG - Changed state to offline
Got direct message.
DEBUG - sendname : bb
DEBUG - destname : aa
DEBUG - message : See you next time.
DEBUG - destsock : 4
DEBUG - buffer : aa is offline. Leaving message successfully
send: Bad file descriptor
DEBUG - newfd : 4
pollserver: new connection from 127.0.0.1 on socket 4
Got register/login message
DEBUG Name : cc
DEBUG - This is a new user
4
DEBUG - Sockfd : 4
DEBUG - Sockfd : 5
DEBUG - Sockfd : 4
DEBUG - Registration Message Sent
Got WHO message from client.
DEBUG pfds[i].fd - 4
DEBUG - ToClient : aa bb*
Got direct message.
DEBUG - sendname : cc
DEBUG - destname : aa
DEBUG - message : Hello aa, I am cc
DEBUG - destsock : 4
DEBUG - buffer : aa is offline. Leaving message successfully
PA — -zsh — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT - Send exit message to server - unregister ourselves from server
WHO - Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> - Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: aa
Welcome aa to join the chat room!
> A new account has been created
> Welcome bb to join the chat room!
> bb to you: Nice to meet you
#BB: Nice to meet you too
?> no such user...
#bb: Nice to meet you too
EXIT
Client Exit...
> aa has left the chatroom
(base) adityapadia@Adityas-MacBook-Air-3 PA %
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT - Send exit message to server - unregister ourselves from server
WHO - Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> - Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: cc
Welcome cc to join the chat room!
A new account has been created
WHO
Getting user list, pls hold on...
If you want to send a message to one of the users, pls send with the format: '#username:message'
> aa bb*
#aa: Hello aa, I am cc
> aa is offline. Leaving message successfully
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT - Send exit message to server - unregister ourselves from server
WHO - Send WHO message to the server - get the list of current users except ourselves
#<user>: <msg> - Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: bb
Welcome bb to join the chat room!
A new account has been created
#aa: Nice to meet you
> aa to you: Nice to meet you too
?> aa has left the chatroom
#aa: See you next time.
> aa is offline. Leaving message successfully
> Welcome cc to join the chat room!


PA — server.o — 134×33
DEBUG – buffer : aa is offline. Leaving message successfully
send: Bad file descriptor
DEBUG – newfd : 4
pollserver: new connection from 127.0.0.1 on socket 4
Got register/login message
DEBUG Name : cc
DEBUG – This is a new user
4
DEBUG – Sockfd : 4
DEBUG – Sockfd : 5
DEBUG – Sockfd : 4
DEBUG – Registration Message Sent
Got WHO message from client.
DEBUG pfds[i].fd – 4
DEBUG – ToClient : aa bb*
Got direct message.
DEBUG – sendname : cc
DEBUG – destname : aa
DEBUG – message : Hello aa, I am cc
DEBUG – destsock : 4
DEBUG – buffer : aa is offline. Leaving message successfully
Got direct message.
DEBUG – sendname : cc
DEBUG – destname : bb
DEBUG – message : Hello bb, I am cc
DEBUG – destsock : 5
DEBUG – username : bb
DEBUG – state : 1
PA — -zsh — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT – Send exit message to server – unregister ourselves from server
WHO – Send WHO message to the server – get the list of current users except ourselves
#<user>: <msg> – Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: aa
Welcome aa to join the chat room!
> A new account has been created
> Welcome bb to join the chat room!
> bb to you: Nice to meet you
#BB: Nice to meet you too
?> no such user...
#bb: Nice to meet you too
EXIT
Client Exit...
> aa has left the chatroom
(base) adityapadia@Adityas-MacBook-Air-3 PA %
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT – Send exit message to server – unregister ourselves from server
WHO – Send WHO message to the server – get the list of current users except ourselves
#<user>: <msg> – Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: cc
Welcome cc to join the chat room!
A new account has been created
WHO
Getting user list, pls hold on...
If you want to send a message to one of the users, pls send with the format: '#username:message'
> aa bb*
#aa: Hello aa, I am cc
> aa is offline. Leaving message successfully
#bb: Hello bb, I am cc
> ?
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT – Send exit message to server – unregister ourselves from server
WHO – Send WHO message to the server – get the list of current users except ourselves
#<user>: <msg> – Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: bb
Welcome bb to join the chat room!
A new account has been created
#aa: Nice to meet you
> aa to you: Nice to meet you too
?> aa has left the chatroom
#aa: See you next time.
> aa is offline. Leaving message successfully
> Welcome cc to join the chat room!
> cc to you: Hello bb, I am cc


PA — server.o — 134×33
DEBUG – sendname : cc
DEBUG – destname : aa
DEBUG – message : Hello aa, I am cc
DEBUG – destsock : 4
DEBUG – buffer : aa is offline. Leaving message successfully
Got direct message.
DEBUG – sendname : cc
DEBUG – destname : bb
DEBUG – message : Hello bb, I am cc
DEBUG – destsock : 5
DEBUG – username : bb
DEBUG – state : 1
DEBUG – newfd : 6
pollserver: new connection from 127.0.0.1 on socket 6
Got register/login message
DEBUG Name : aa
DEBUG – This is an existing user
DEBUG – Login Successful
DEBUG – Changed state to online
DEBUG – Reached here
DEBUG – name aa
DEBUG – filename aa.txt
DEBUG – buffer : Welcome back! The message box contains:
bb to you: See you next time.
cc to you: Hello aa, I am cc
DEBUG – welcome back message : aa is online!
DEBUG – users_count : 3
5
4
PA — client.o — 134×33
EXIT – Send exit message to server – unregister ourselves from server
WHO – Send WHO message to the server – get the list of current users except ourselves
#<user>: <msg> – Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: aa
Welcome aa to join the chat room!
> A new account has been created
> Welcome bb to join the chat room!
> bb to you: Nice to meet you
#BB: Nice to meet you too
?> no such user...
#bb: Nice to meet you too
EXIT
Client Exit...
> aa has left the chatroom
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT – Send exit message to server – unregister ourselves from server
WHO – Send WHO message to the server – get the list of current users except ourselves
#<user>: <msg> – Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: aa
Welcome back! The message box contains:
bb to you: See you next time.
cc to you: Hello aa, I am cc
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT – Send exit message to server – unregister ourselves from server
WHO – Send WHO message to the server – get the list of current users except ourselves
#<user>: <msg> – Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: cc
Welcome cc to join the chat room!
A new account has been created
WHO
Getting user list, pls hold on...
If you want to send a message to one of the users, pls send with the format: '#username:message'
> aa bb*
#aa: Hello aa, I am cc
> aa is offline. Leaving message successfully
#bb: Hello bb, I am cc
> ?> aa is online!
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT – Send exit message to server – unregister ourselves from server
WHO – Send WHO message to the server – get the list of current users except ourselves
#<user>: <msg> – Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: bb
Welcome bb to join the chat room!
A new account has been created
#aa: Nice to meet you
> aa to you: Nice to meet you too
?> aa has left the chatroom
#aa: See you next time.
> aa is offline. Leaving message successfully
> Welcome cc to join the chat room!
> cc to you: Hello bb, I am cc
?> aa is online!


PA — server.o — 134×33
DEBUG – destsock : 4
DEBUG – buffer : aa is offline. Leaving message successfully
Got direct message.
DEBUG – sendname : cc
DEBUG – destname : bb
DEBUG – message : Hello bb, I am cc
DEBUG – destsock : 5
DEBUG – username : bb
DEBUG – state : 1
DEBUG – newfd : 6
pollserver: new connection from 127.0.0.1 on socket 6
Got register/login message
DEBUG Name : aa
DEBUG – This is an existing user
DEBUG – Login Successful
DEBUG – Changed state to online
DEBUG – Reached here
DEBUG – name aa
DEBUG – filename aa.txt
DEBUG – buffer : Welcome back! The message box contains:
bb to you: See you next time.
cc to you: Hello aa, I am cc
DEBUG – welcome back message : aa is online!
DEBUG – users_count : 3
5
4
Got WHO message from client.
DEBUG pfds[i].fd – 4
DEBUG – ToClient : aa* bb*
PA — client.o — 134×33
EXIT – Send exit message to server – unregister ourselves from server
WHO – Send WHO message to the server – get the list of current users except ourselves
#<user>: <msg> – Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: aa
Welcome aa to join the chat room!
> A new account has been created
> Welcome bb to join the chat room!
> bb to you: Nice to meet you
#BB: Nice to meet you too
?> no such user...
#bb: Nice to meet you too
EXIT
Client Exit...
> aa has left the chatroom
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT – Send exit message to server – unregister ourselves from server
WHO – Send WHO message to the server – get the list of current users except ourselves
#<user>: <msg> – Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: aa
Welcome back! The message box contains:
bb to you: See you next time.
cc to you: Hello aa, I am cc
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT – Send exit message to server – unregister ourselves from server
WHO – Send WHO message to the server – get the list of current users except ourselves
#<user>: <msg> – Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: cc
Welcome cc to join the chat room!
A new account has been created
WHO
Getting user list, pls hold on...
If you want to send a message to one of the users, pls send with the format: '#username:message'
> aa bb*
#aa: Hello aa, I am cc
> aa is offline. Leaving message successfully
#bb: Hello bb, I am cc
> ?> aa is online!
WHO
Getting user list, pls hold on...
If you want to send a message to one of the users, pls send with the format: '#username:message'
> aa* bb*
PA — client.o — 134×33
(base) adityapadia@Adityas-MacBook-Air-3 PA % ./client.o
DEBUG sockfd : 3
Socket successfully created...
Connected to the server...
Hello dear user pls select one of the following options:
EXIT – Send exit message to server – unregister ourselves from server
WHO – Send WHO message to the server – get the list of current users except ourselves
#<user>: <msg> – Send <MSG>> message to the server for <user>
Or input messages sending to everyone in the chatroom.
Welcome to the chat room!
Please enter a nickname.
Enter the string: bb
Welcome bb to join the chat room!
A new account has been created
#aa: Nice to meet you
> aa to you: Nice to meet you too
?> aa has left the chatroom
#aa: See you next time.
> aa is offline. Leaving message successfully
> Welcome cc to join the chat room!
> cc to you: Hello bb, I am cc
?> aa is online!

The last screenshot shows simultaneous sending and receiving, it is clearly seen that while aa was typing hello, cc send a broadcast message saying hello aa, and aa received the message immediately.