

Computer Architecture and Operating Systems

Assignment 2

Implementation

```
int main() {
    int sockfd, newsockfd, len;
    struct sockaddr_un local, remote;
    pthread_t thread;

    // Create a socket file descriptor

    sockfd = socket(AF_UNIX, SOCK_STREAM, 0);

    if (sockfd == -1) {
        perror("socket");
    }

    // Add information about the Unix domain address

    local.sun_family = AF_UNIX;
    strcpy(local.sun_path, SOCK_PATH);
    unlink(local.sun_path);
    len = strlen(local.sun_path) + sizeof(local.sun_family);

    // Bind the socket to an address

    if (bind(sockfd, (struct sockaddr *)&local, len) == -1) {
        perror("bind");
        exit(1);
    }

    // Listen to incoming connections from client programs

    if (listen(sockfd, 5) == -1) {
        perror("listen");
        exit(1);
    }

    printf("\n\n__WELCOME TO THE CHAT__\n\n");

    while (1) {
        socklen_t remotelen = sizeof(remote);

        // We create a new socked fd which is connected to the client

        newsockfd = accept(sockfd, (struct sockaddr *)&remote, &remotelen);
```

```

while (1) {
    socklen_t remotelen = sizeof(remote);

    // We create a new socked fd which is connected to the client

    newsockfd = accept(sockfd, (struct sockaddr *)&remote, &remotelen);

    if(newsockfd == -1) {
        perror("accept");
        exit(1);
    }

    // Check for maximum users

    if (user_count >= MAX_USERS) {
        printf("FAILED: Maximum users connected\n");
        close(newsockfd);
        continue;
    }

    // Set user details

    user_t *new_user = (user_t *) malloc(sizeof(user_t));
    new_user->userid = ++user_count;
    new_user->sockfd = newsockfd;
    new_user->addr = remote;

    // Add user to list of users

    for (int i = 0; i < MAX_USERS; i++) {
        if (list_of_users[i] == NULL) {
            list_of_users[i] = new_user;
            break;
        }
    }

    // Create a new thread for the user

    pthread_create(&thread, NULL, &welcome_user, (void *) new_user);
}
}

```

Errors

Error checking added for:

1. `send()`
2. `recv()`
3. `socket()`
4. `bind()`
5. `connect()`
6. Invalid usernames
7. Maximum users

How to compile and test the program

1. make compile to compile the files
2. ./server to run the server
3. ./client to run the client
4. Open new terminal windows to have multiple clients
5. Start chatting
6. Remember to follow the instructions that are displayed when connected

Inputs the user should give

```
anmolgupta at Anmols-MacBook-Air-2 in ~/Desktop/Semester3/CAOS/Chatroom-in-C/final
$ ./server

___WELCOME TO THE CHAT___
```

No input is required to get the server running. The program asks for username after running the client.

The chat then begins normally.

```
anmolgupta at Anmols-MacBook-Air-2 in ~/Desktop/Semester3/CAOS/Chatroom-in-C/final
$ ./client
Enter your name (3 - 10 characters):
```

Expected output

```
final — server — 56x19
anmolgupta at Anmols-MacBook-Air-2 in ~/Desktop/Semester
3/CAOS/Chatroom-in-C/final
$ make compile
rm socketfil
gcc -o server server.c
gcc -o client client.c

anmolgupta at Anmols-MacBook-Air-2 in ~/Desktop/Semester
3/CAOS/Chatroom-in-C/final
$ ./server

___WELCOME TO THE CHAT___

anmol has joined the chatroom.
naman has joined the chatroom.
ojas has joined the chatroom.
[]

final — client — 51x20
$ ./client
Enter your name (3 - 10 characters): anmol

Connecting to server...

Connected to server

___WELCOME TO THE CHAT___

Do '@[name] [your message]' to send to one person
Do '@all [your message]' to send to everyone
Enter 'quit' to exit chatroom

> naman has joined the chatroom.
> ojas has joined the chatroom.
> []

final — client — 58x19
$ ./client
Enter your name (3 - 10 characters): naman

Connecting to server...

Connected to server

___WELCOME TO THE CHAT___

Do '@[name] [your message]' to send to one person
Do '@all [your message]' to send to everyone
Enter 'quit' to exit chatroom

> ojas has joined the chatroom.
> []

final — client ojas — 52x19
ster3/CAOS/Chatroom-in-C/final
$ ./client ojas
Enter your name (3 - 10 characters): ojas

Connecting to server...

Connected to server

___WELCOME TO THE CHAT___

Do '@[name] [your message]' to send to one person
Do '@all [your message]' to send to everyone
Enter 'quit' to exit chatroom

> []
```

Expected output

```
final — server — 56x19
3/CAOS/Chatroom-in-C/final
$ make compile
rm socketfil
gcc -o server server.c
gcc -o client client.c

anmolgupta at Anmols-MacBook-Air-2 in ~/Desktop/Semester
3/CAOS/Chatroom-in-C/final
$ ./server

___WELCOME TO THE CHAT___

anmol has joined the chatroom.
naman has joined the chatroom.
ojas has joined the chatroom.
anmol: @all hey
naman: @ojas how r u ojas

```

```
final — client — 51x20
Enter your name (3 - 10 characters): anmol
Connecting to server...
Connected to server

___WELCOME TO THE CHAT___

Do '@[name] [your message]' to send to one person
Do '@all [your message]' to send to everyone
Enter 'quit' to exit chatroom

> naman has joined the chatroom.
> ojas has joined the chatroom.
> @all hey
> 
```

```
final — client — 58x19
Connecting to server...
Connected to server

___WELCOME TO THE CHAT___

Do '@[name] [your message]' to send to one person
Do '@all [your message]' to send to everyone
Enter 'quit' to exit chatroom

> ojas has joined the chatroom.
> anmol: @all hey
> @ojas how r u ojas
> 
```

```
final — client ojas — 52x19
Enter your name (3 - 10 characters): ojas
Connecting to server...
Connected to server

___WELCOME TO THE CHAT___

Do '@[name] [your message]' to send to one person
Do '@all [your message]' to send to everyone
Enter 'quit' to exit chatroom

> anmol: @all hey
> naman: @ojas how r u ojas
> 
```