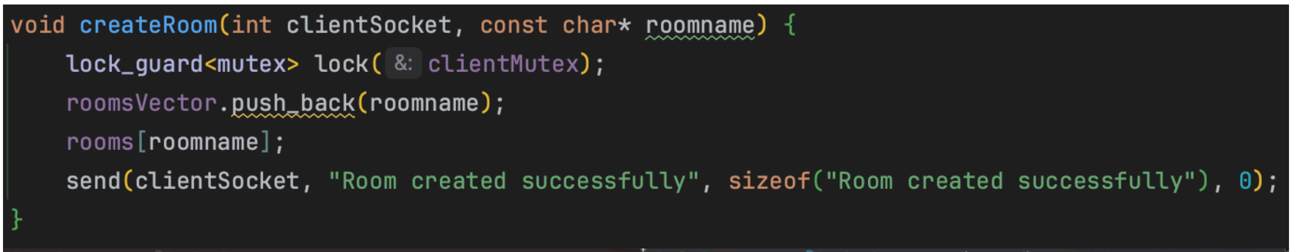
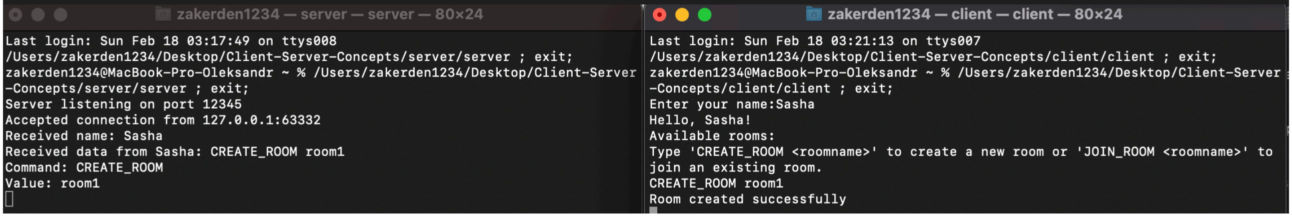


```
void createRoom(int clientSocket, const char* roomname) {
    lock_guard<mutex> lock( &: clientMutex);
    roomsVector.push_back(roomname);
    rooms[roomname];
    send(clientSocket, "Room created successfully", sizeof("Room created successfully"), 0);
}
```

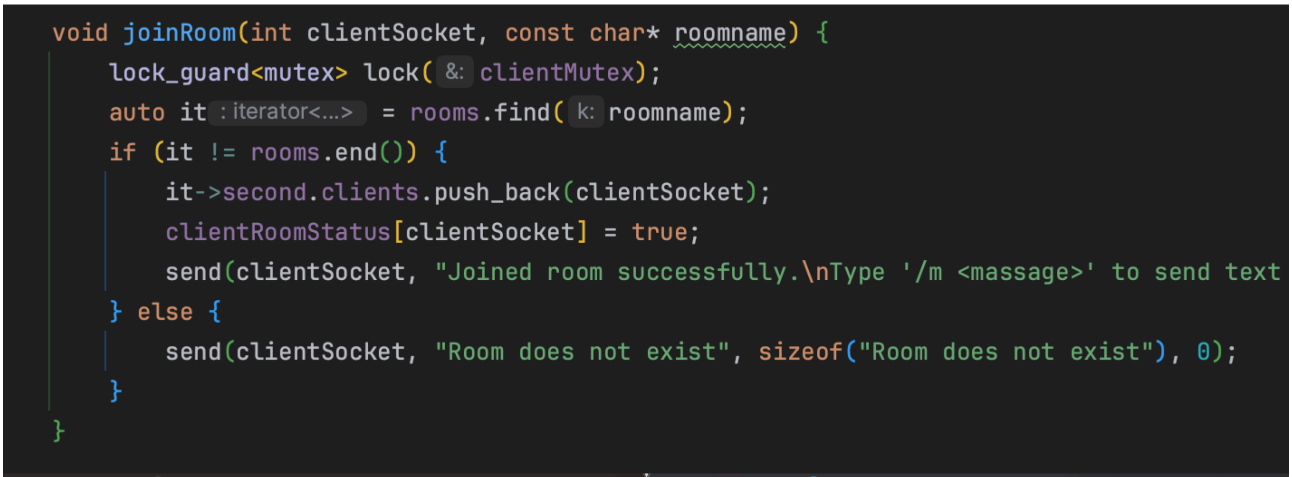


zakerden1234 — server — server — 80×24

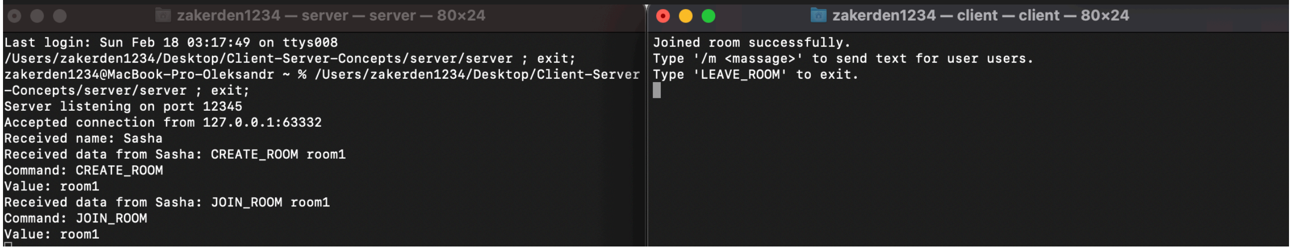
```
Last login: Sun Feb 18 03:17:49 on ttys008
/Users/zakerden1234/Desktop/Client-Server-Concepts/server/server ; exit;
zakerden1234@MacBook-Pro-Oleksandr ~ % /Users/zakerden1234/Desktop/Client-Server
-Concepts/server/server ; exit;
Server listening on port 12345
Accepted connection from 127.0.0.1:63332
Received name: Sasha
Received data from Sasha: CREATE_ROOM room1
Command: CREATE_ROOM
Value: room1
```

zakerden1234 — client — client — 80×24

```
Last login: Sun Feb 18 03:21:13 on ttys007
/Users/zakerden1234/Desktop/Client-Server-Concepts/client/client ; exit;
zakerden1234@MacBook-Pro-Oleksandr ~ % /Users/zakerden1234/Desktop/Client-Server
-Concepts/client/client ; exit;
Enter your name:Sasha
Hello, Sasha!
Available rooms:
Type 'CREATE_ROOM <roomname>' to create a new room or 'JOIN_ROOM <roomname>' to
join an existing room.
CREATE_ROOM room1
Room created successfully
```



```
void joinRoom(int clientSocket, const char* roomname) {
    lock_guard<mutex> lock( &: clientMutex);
    auto it : iterator<...> = rooms.find( k: roomname);
    if (it != rooms.end()) {
        it->second.clients.push_back(clientSocket);
        clientRoomStatus[clientSocket] = true;
        send(clientSocket, "Joined room successfully.\nType '/m <massage>' to send text
    } else {
        send(clientSocket, "Room does not exist", sizeof("Room does not exist"), 0);
    }
}
```

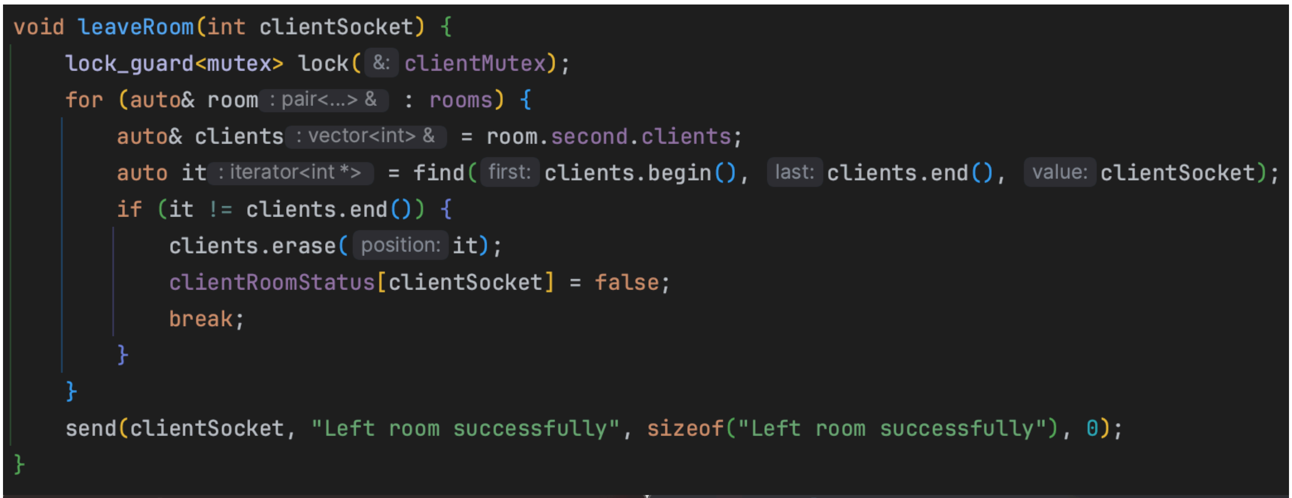


zakerden1234 — server — server — 80×24

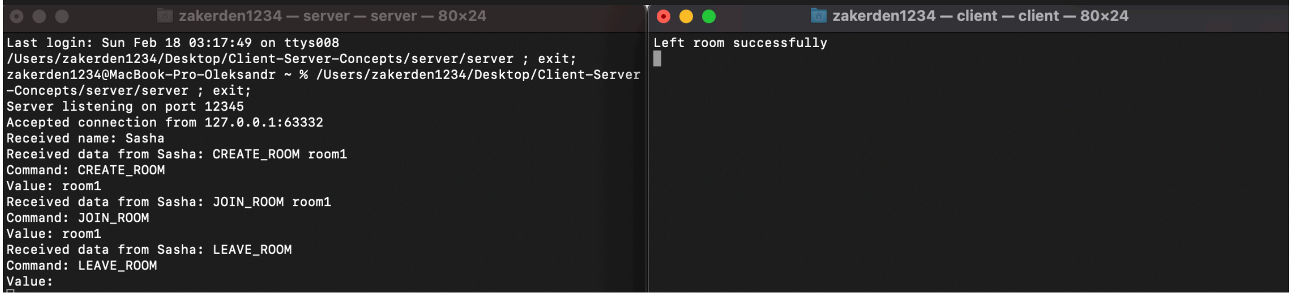
```
Last login: Sun Feb 18 03:17:49 on ttys008
/Users/zakerden1234/Desktop/Client-Server-Concepts/server/server ; exit;
zakerden1234@MacBook-Pro-Oleksandr ~ % /Users/zakerden1234/Desktop/Client-Server
-Concepts/server/server ; exit;
Server listening on port 12345
Accepted connection from 127.0.0.1:63332
Received name: Sasha
Received data from Sasha: CREATE_ROOM room1
Command: CREATE_ROOM
Value: room1
Received data from Sasha: JOIN_ROOM room1
Command: JOIN_ROOM
Value: room1
```

zakerden1234 — client — client — 80×24

```
Joined room successfully.
Type '/m <massage>' to send text for user users.
Type 'LEAVE_ROOM' to exit.
```



```
void leaveRoom(int clientSocket) {
    lock_guard<mutex> lock( &: clientMutex);
    for (auto& room : pair<...> & : rooms) {
        auto& clients : vector<int> & = room.second.clients;
        auto it : iterator<int *> = find( first: clients.begin(), last: clients.end(), value: clientSocket);
        if (it != clients.end()) {
            clients.erase( position: it);
            clientRoomStatus[clientSocket] = false;
            break;
        }
    }
    send(clientSocket, "Left room successfully", sizeof("Left room successfully"), 0);
}
```

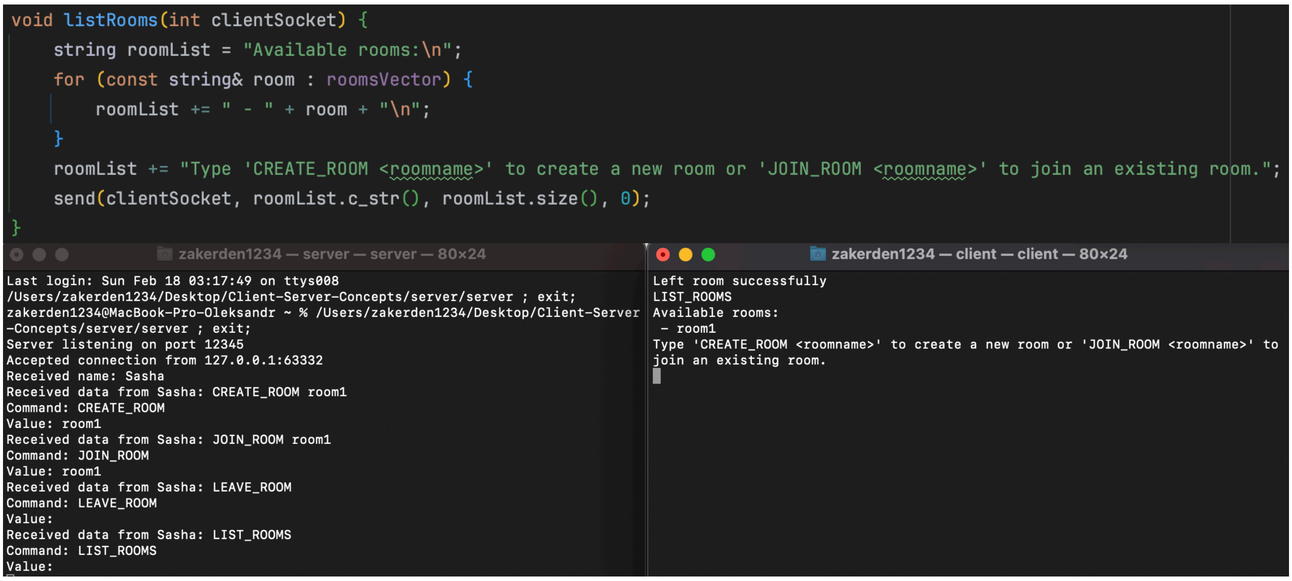


zakerden1234 — server — server — 80×24

```
Last login: Sun Feb 18 03:17:49 on ttys008
/Users/zakerden1234/Desktop/Client-Server-Concepts/server/server ; exit;
zakerden1234@MacBook-Pro-Oleksandr ~ % /Users/zakerden1234/Desktop/Client-Server
-Concepts/server/server ; exit;
Server listening on port 12345
Accepted connection from 127.0.0.1:63332
Received name: Sasha
Received data from Sasha: CREATE_ROOM room1
Command: CREATE_ROOM
Value: room1
Received data from Sasha: JOIN_ROOM room1
Command: JOIN_ROOM
Value: room1
Received data from Sasha: LEAVE_ROOM
Command: LEAVE_ROOM
Value:
```

zakerden1234 — client — client — 80×24

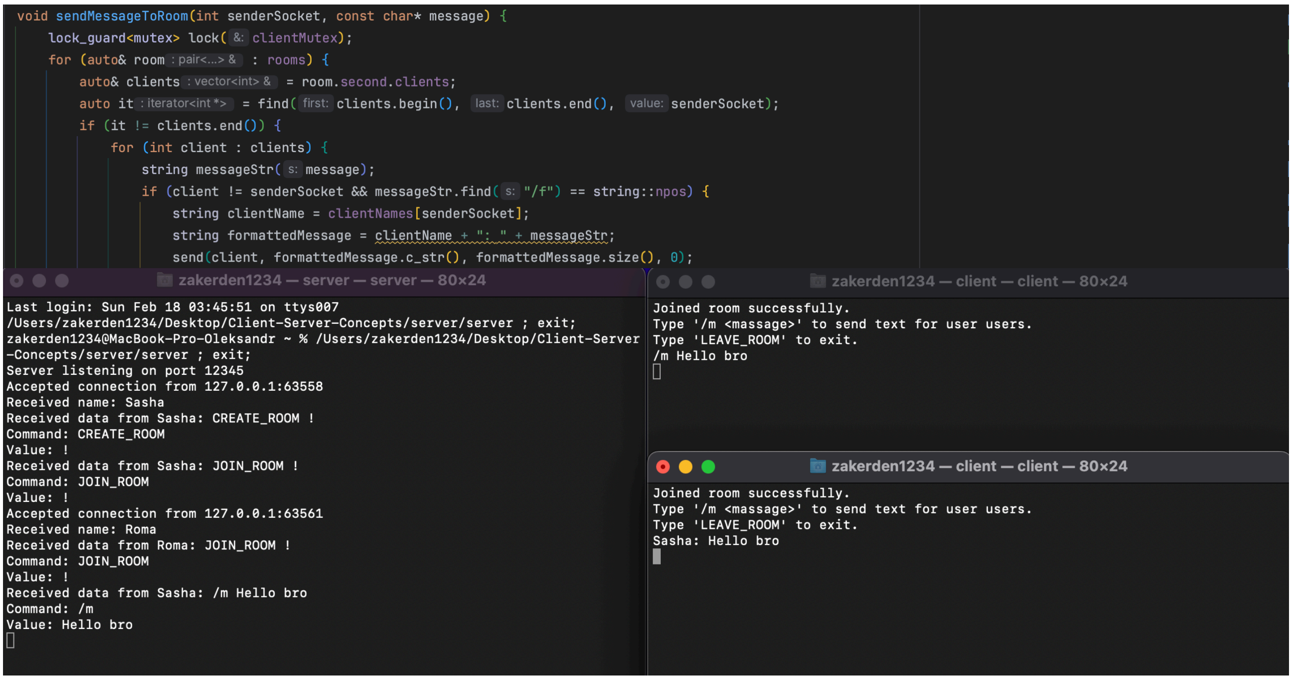
```
Left room successfully
```



```cpp
void listRooms(int clientSocket) {
    string roomList = "Available rooms:\n";
    for (const string& room : roomsVector) {
        roomList += " - " + room + "\n";
    }
    roomList += "Type 'CREATE_ROOM <roomname>' to create a new room or 'JOIN_ROOM <roomname>' to join an existing room.";
    send(clientSocket, roomList.c_str(), roomList.size(), 0);
}
```

```
Last login: Sun Feb 18 03:17:49 on ttys008
/Users/zakerden1234/Desktop/Client-Server-Concepts/server/server ; exit;
zakerden1234@MacBook-Pro-Oleksandr ~ % /Users/zakerden1234/Desktop/Client-Server
-Concepts/server/server ; exit;
Server listening on port 12345
Accepted connection from 127.0.0.1:63332
Received name: Sasha
Received data from Sasha: CREATE_ROOM room1
Command: CREATE_ROOM
Value: room1
Received data from Sasha: JOIN_ROOM room1
Command: JOIN_ROOM
Value: room1
Received data from Sasha: LEAVE_ROOM
Command: LEAVE_ROOM
Value:
Received data from Sasha: LIST_ROOMS
Command: LIST_ROOMS
Value:
```

```
Left room successfully
LIST_ROOMS
Available rooms:
 - room1
Type 'CREATE_ROOM <roomname>' to create a new room or 'JOIN_ROOM <roomname>' to
join an existing room.
```

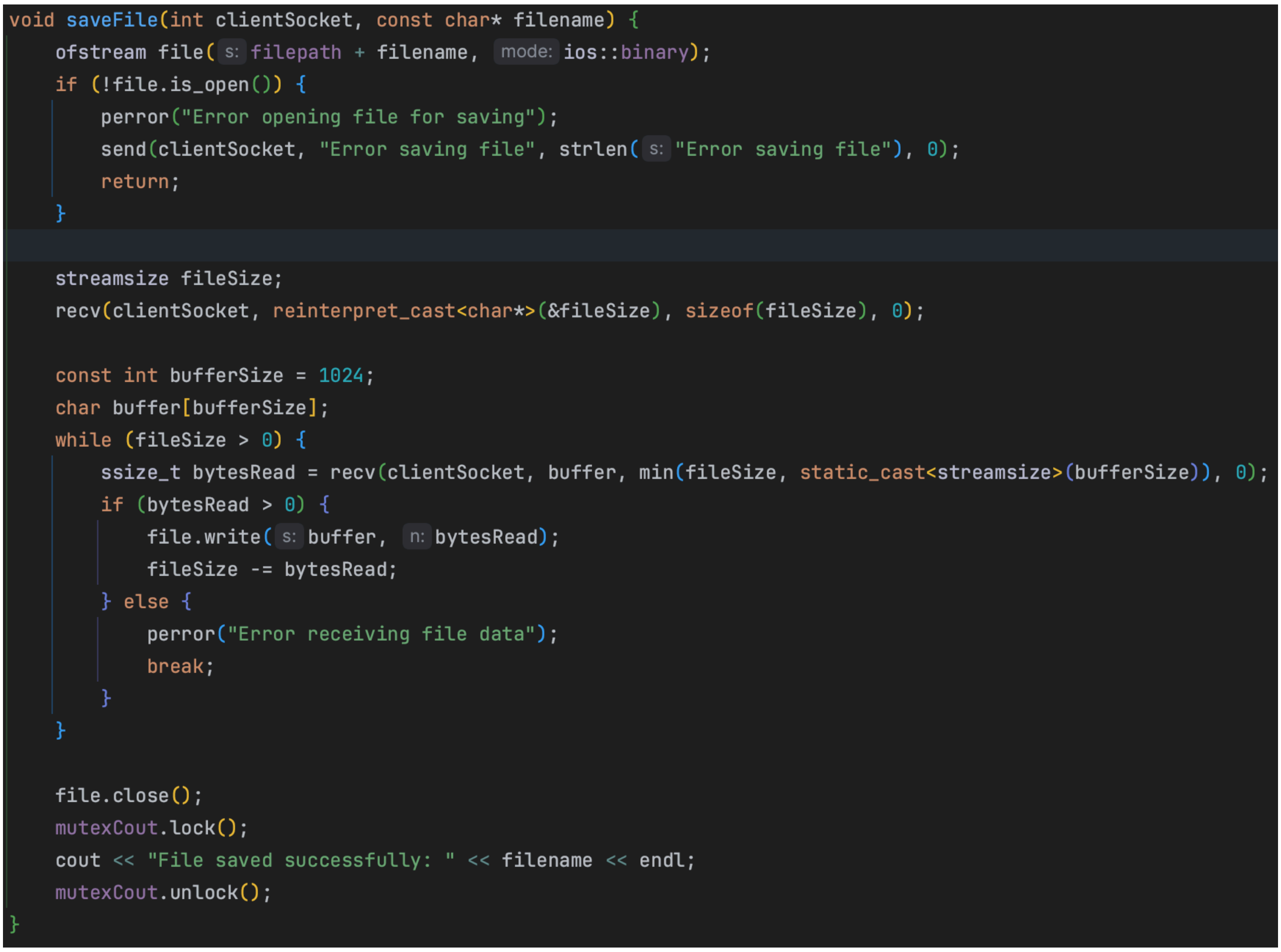


```cpp
void sendMessageToRoom(int senderSocket, const char* message) {
    lock_guard<mutex> lock(clientMutex);
    for (auto& room : rooms) {
        auto& clients = room.second.clients;
        auto it = find(clients.begin(), clients.end(), senderSocket);
        if (it != clients.end()) {
            for (int client : clients) {
                string messageStr(message);
                if (client != senderSocket && messageStr.find("/f") == string::npos) {
                    string clientName = clientNames[senderSocket];
                    string formattedMessage = clientName + ": " + messageStr;
                    send(client, formattedMessage.c_str(), formattedMessage.size(), 0);
```

```
Last login: Sun Feb 18 03:45:51 on ttys007
/Users/zakerden1234/Desktop/Client-Server-Concepts/server/server ; exit;
zakerden1234@MacBook-Pro-Oleksandr ~ % /Users/zakerden1234/Desktop/Client-Server
-Concepts/server/server ; exit;
Server listening on port 12345
Accepted connection from 127.0.0.1:63558
Received name: Sasha
Received data from Sasha: CREATE_ROOM !
Command: CREATE_ROOM
Value: !
Received data from Sasha: JOIN_ROOM !
Command: JOIN_ROOM
Value: !
Accepted connection from 127.0.0.1:63561
Received name: Roma
Received data from Roma: JOIN_ROOM !
Command: JOIN_ROOM
Value: !
Received data from Sasha: /m Hello bro
Command: /m
Value: Hello bro
```

```
Joined room successfully.
Type '/m <massage>' to send text for user users.
Type 'LEAVE_ROOM' to exit.
/m Hello bro
```

```
Joined room successfully.
Type '/m <massage>' to send text for user users.
Type 'LEAVE_ROOM' to exit.
Sasha: Hello bro
```

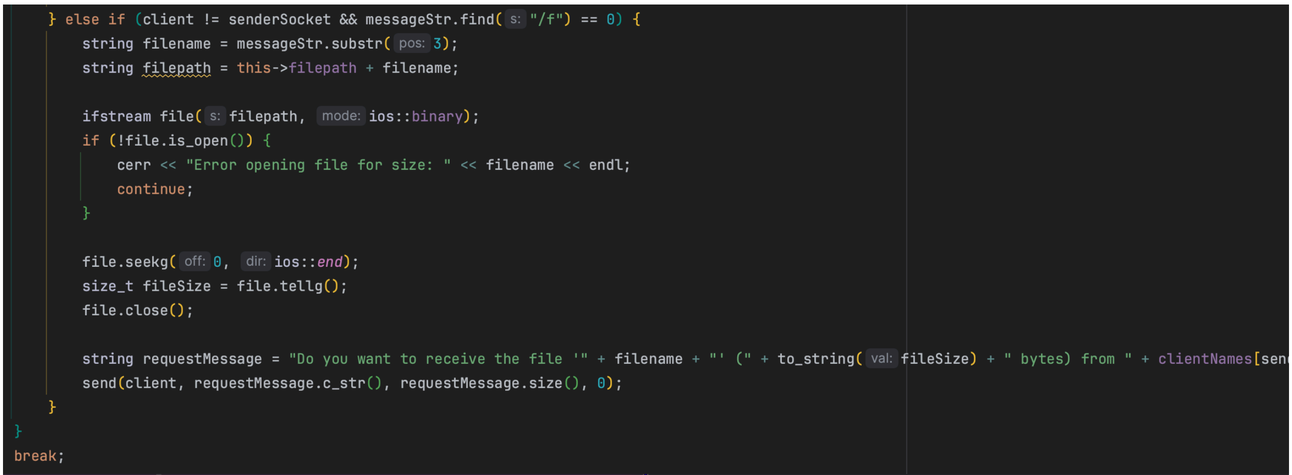


```
void saveFile(int clientSocket, const char* filename) {
    ofstream file( s: filepath + filename,  mode: ios::binary);
    if (!file.is_open()) {
        perror("Error opening file for saving");
        send(clientSocket, "Error saving file", strlen( s: "Error saving file"), 0);
        return;
    }

    streamsize fileSize;
    recv(clientSocket, reinterpret_cast<char*>(&fileSize), sizeof(fileSize), 0);

    const int bufferSize = 1024;
    char buffer[bufferSize];
    while (fileSize > 0) {
        ssize_t bytesRead = recv(clientSocket, buffer, min(fileSize, static_cast<streamsize>(bufferSize)), 0);
        if (bytesRead > 0) {
            file.write( s: buffer,  n: bytesRead);
            fileSize -= bytesRead;
        } else {
            perror("Error receiving file data");
            break;
        }
    }

    file.close();
    mutexCout.lock();
    cout << "File saved successfully: " << filename << endl;
    mutexCout.unlock();
}
```

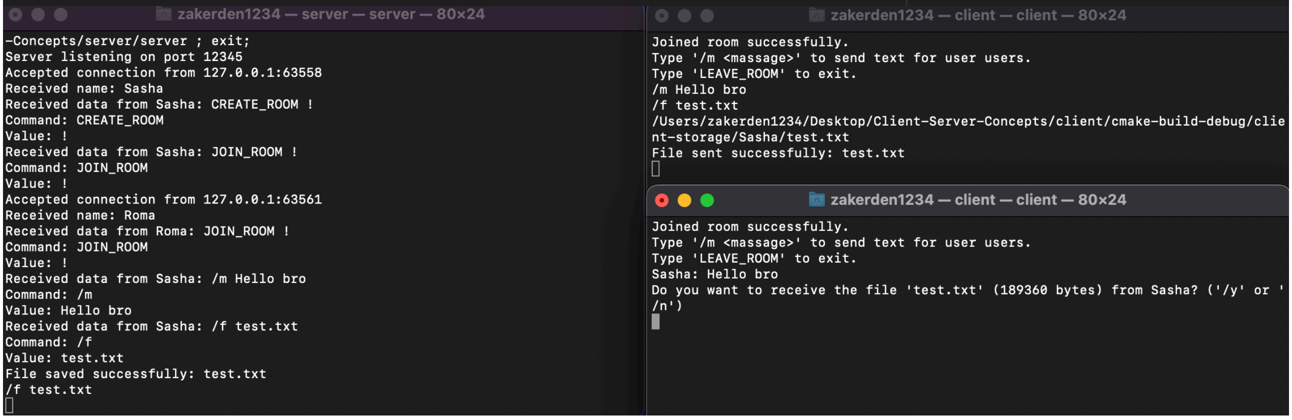


```
        } else if (client != senderSocket && messageStr.find( s: "/f") == 0) {
            string filename = messageStr.substr( pos: 3);
            string filepath = this->filepath + filename;

            ifstream file( s: filepath,  mode: ios::binary);
            if (!file.is_open()) {
                cerr << "Error opening file for size: " << filename << endl;
                continue;
            }

            file.seekg( off: 0,  dir: ios::end);
            size_t fileSize = file.tellg();
            file.close();

            string requestMessage = "Do you want to receive the file '" + filename + "' (" + to_string( val: fileSize) + " bytes) from " + clientNames[sen
            send(client, requestMessage.c_str(), requestMessage.size(), 0);
        }
    }
    break;
```



zakerden1234 — server — server — 80×24

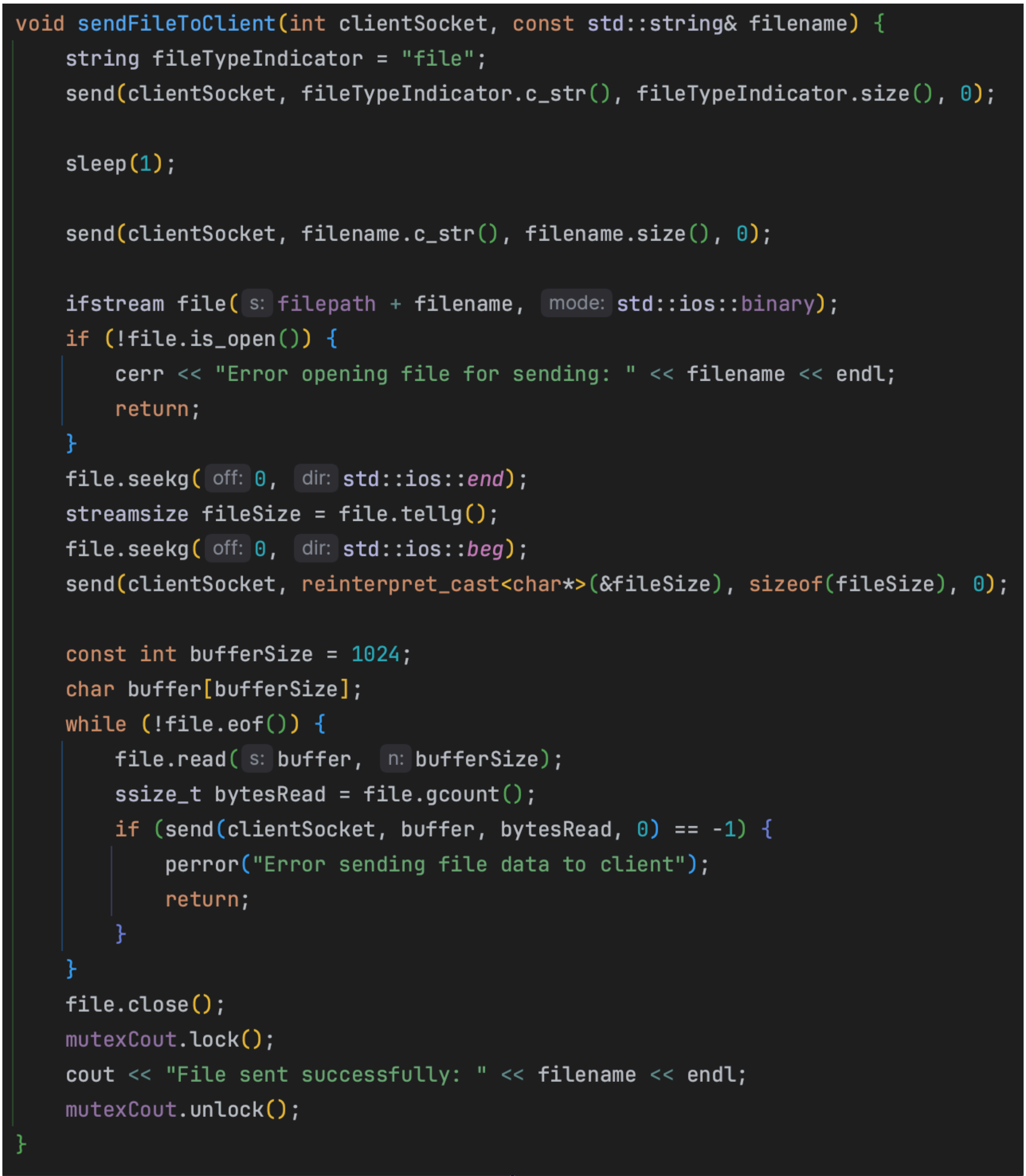
```
-Concepts/server/server ; exit;
Server listening on port 12345
Accepted connection from 127.0.0.1:63558
Received name: Sasha
Received data from Sasha: CREATE_ROOM !
Command: CREATE_ROOM
Value: !
Received data from Sasha: JOIN_ROOM !
Command: JOIN_ROOM
Value: !
Accepted connection from 127.0.0.1:63561
Received name: Roma
Received data from Roma: JOIN_ROOM !
Command: JOIN_ROOM
Value: !
Received data from Sasha: /m Hello bro
Command: /m
Value: Hello bro
Received data from Sasha: /f test.txt
Command: /f
Value: test.txt
File saved successfully: test.txt
/f test.txt
```

zakerden1234 — client — client — 80×24

```
Joined room successfully.
Type '/m <massage>' to send text for user users.
Type 'LEAVE_ROOM' to exit.
/m Hello bro
/f test.txt
/Users/zakerden1234/Desktop/Client-Server-Concepts/client/cmake-build-debug/clie
nt-storage/Sasha/test.txt
File sent successfully: test.txt
```

zakerden1234 — client — client — 80×24

```
Joined room successfully.
Type '/m <massage>' to send text for user users.
Type 'LEAVE_ROOM' to exit.
Sasha: Hello bro
Do you want to receive the file 'test.txt' (189360 bytes) from Sasha? ('/y' or '
/n')
```



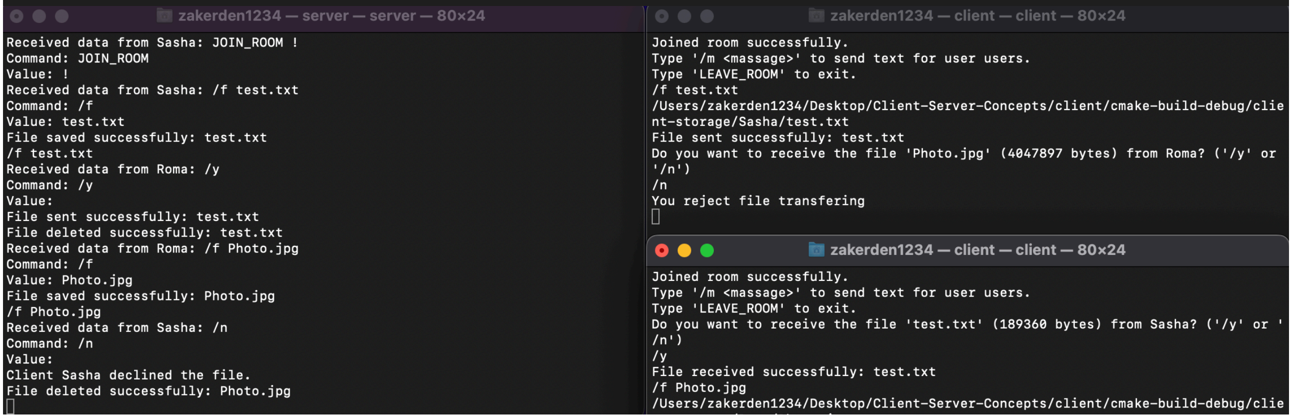
```
void sendFileToClient(int clientSocket, const std::string& filename) {
    string fileTypeIndicator = "file";
    send(clientSocket, fileTypeIndicator.c_str(), fileTypeIndicator.size(), 0);

    sleep(1);

    send(clientSocket, filename.c_str(), filename.size(), 0);

    ifstream file( s: filepath + filename,  mode: std::ios::binary);
    if (!file.is_open()) {
        cerr << "Error opening file for sending: " << filename << endl;
        return;
    }
    file.seekg( off: 0,  dir: std::ios::end);
    streamsize fileSize = file.tellg();
    file.seekg( off: 0,  dir: std::ios::beg);
    send(clientSocket, reinterpret_cast<char*>(&fileSize), sizeof(fileSize), 0);

    const int bufferSize = 1024;
    char buffer[bufferSize];
    while (!file.eof()) {
        file.read( s: buffer,  n: bufferSize);
        ssize_t bytesRead = file.gcount();
        if (send(clientSocket, buffer, bytesRead, 0) == -1) {
            perror("Error sending file data to client");
            return;
        }
    }
    file.close();
    mutexCout.lock();
    cout << "File sent successfully: " << filename << endl;
    mutexCout.unlock();
}
```



```
Received data from Sasha: JOIN_ROOM !
Command: JOIN_ROOM
Value: !
Received data from Sasha: /f test.txt
Command: /f
Value: test.txt
File saved successfully: test.txt
/f test.txt
Received data from Roma: /y
Command: /y
Value:
File sent successfully: test.txt
File deleted successfully: test.txt
Received data from Roma: /f Photo.jpg
Command: /f
Value: Photo.jpg
File saved successfully: Photo.jpg
/f Photo.jpg
Received data from Sasha: /n
Command: /n
Value:
Client Sasha declined the file.
File deleted successfully: Photo.jpg
```

```
Joined room successfully.
Type '/m <massage>' to send text for user users.
Type 'LEAVE_ROOM' to exit.
/f test.txt
/Users/zakerden1234/Desktop/Client-Server-Concepts/client/cmake-build-debug/clie
nt-storage/Sasha/test.txt
File sent successfully: test.txt
Do you want to receive the file 'Photo.jpg' (4047897 bytes) from Roma? ('/y' or
'/n')
/n
You reject file transfering
```

```
Joined room successfully.
Type '/m <massage>' to send text for user users.
Type 'LEAVE_ROOM' to exit.
Do you want to receive the file 'test.txt' (189360 bytes) from Sasha? ('/y' or '
/n')
/y
File received successfully: test.txt
/f Photo.jpg
/Users/zakerden1234/Desktop/Client-Server-Concepts/client/cmake-build-debug/clie
```