



LẬP TRÌNH MINICHAT

Bài giảng 3



MiniChat: Chức Năng

- **Chức năng**
 - Cho phép nhiều user đăng ký vào các nhóm để trò chuyện với nhau.
- **Hình thức trò chuyện**
 - Một user với một user (1-1)
 - Một user với một nhóm các user (1-n)
 - Một user với tất cả các user (1-n)

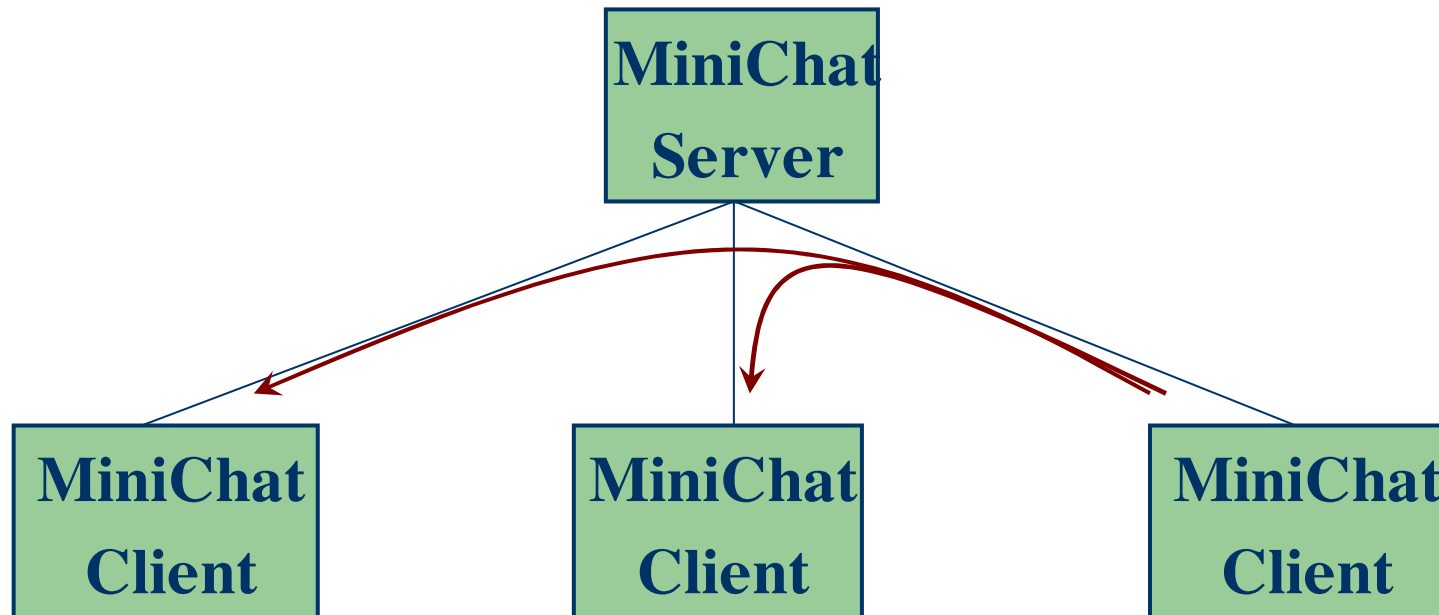


MiniChat: Kiến Trúc (1)

- **Mô hình lựa chọn**
 - Client/server
- **Server**
 - Quản lý các nhóm và các user của từng nhóm.
 - Phân phối chuỗi thông tin từ một user đến các user khác.
- **Client**
 - Giao tiếp với các user.
 - Cho phép họ tạo nhóm, đăng ký tham gia vào nhóm; gửi/nhận thông tin cho nhau.



MiniChat: Kiến Trúc (2)





MiniChat: Qui Trình (1)

- MiniChat server luôn ở trạng thái sẵn sàng để chờ kết nối từ client
- Nếu một user muốn chat, người user phải kết nối đến MiniChat server, sử dụng MiniChat client
 - Mặc định, khi vừa kết nối, user chưa tham gia vào bất kỳ nhóm nào
- Sau khi kết nối, user có thể tham gia vào các nhóm có sẵn để chat, hoặc tạo ra những nhóm mới
- Sau khi tham gia vào một nhóm nào đấy, user có thể gửi một mẫu tin cho tất cả các user trong nhóm hoặc đến từng user cụ thể



MiniChat: Qui Trình (2)

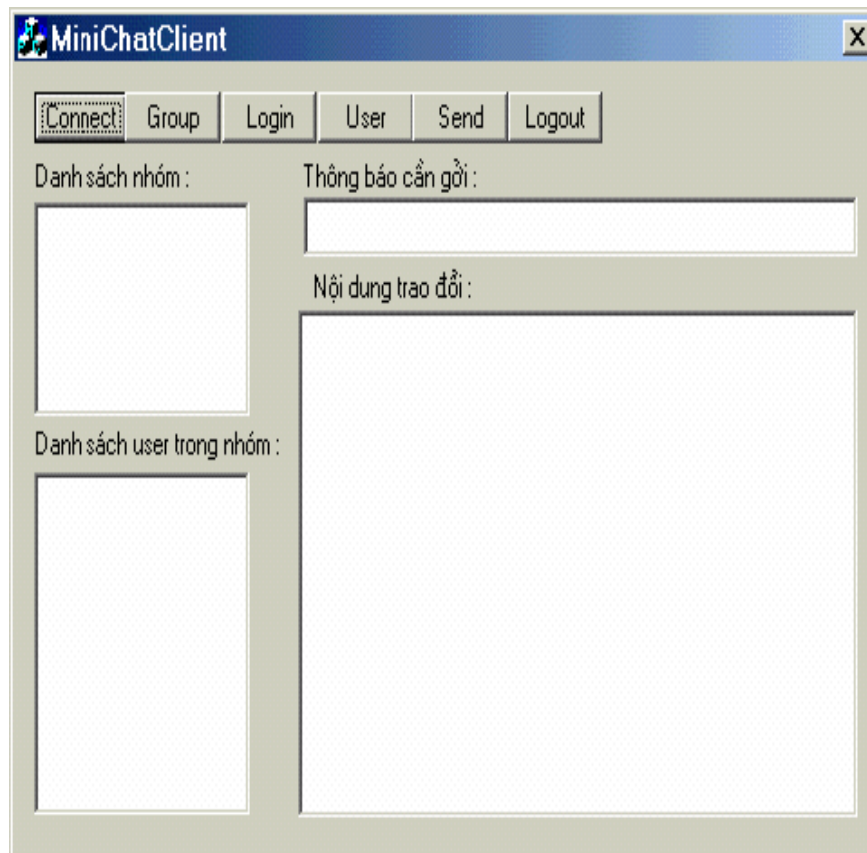
- MiniChat server có một danh sách các kết nối đến từ user và danh sách các nhóm đang tồn tại
- Khi nhận được yêu cầu kết nối từ user, server sẽ đưa kết nối (socket) vào danh sách
- Khi nhận được yêu cầu tạo nhóm từ user, server sẽ tạo ra một nhóm và thêm user đưa ra yêu cầu tạo nhóm vào nhóm mới được tạo, nhóm mới sẽ được đưa vào danh sách
- Khi nhận được yêu cầu chat từ user, server sẽ gửi mẫu tin đến tất cả các user trong nhóm (hoặc từng user cụ thể) dùng các kết nối đã có trong danh sách



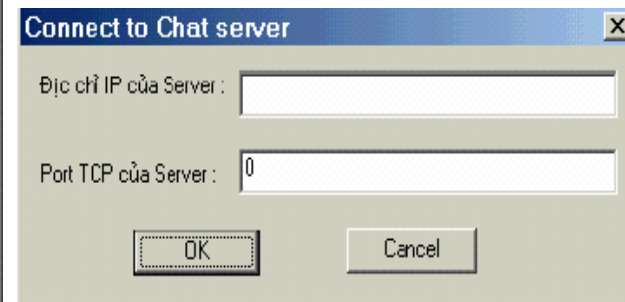
MiniChat: Giao Thức

- **Giao thức cơ bản dùng cho hệ thống MiniChat**
 - Lệnh GLIST <CRLF>
 - Lệnh ULIST <CRLF>
 - Lệnh LOGIN <tên group>,<tên user> <CRLF>
 - Lệnh SEND <string> <CRLF>
 - Lệnh LOGOU <CRLF>
- **Dạng reply cho tất cả các request**
 - N <chuỗi dữ liệu phụ kèm theo>
 - N = 1: Thành công, N = 0: Thất bại

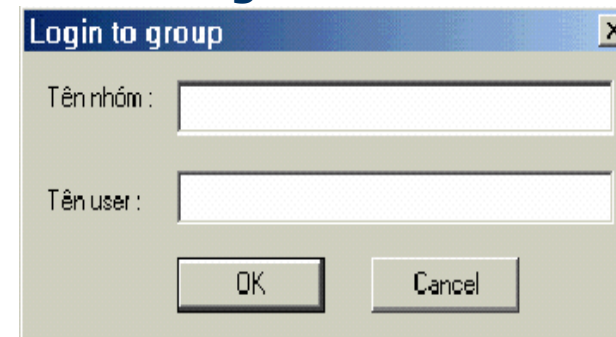
MiniChat: Giao Diện (Ví dụ)



Cửa sổ nhập thông tin của button **"Connect"**



Cửa sổ nhập thông tin của button **"login"**



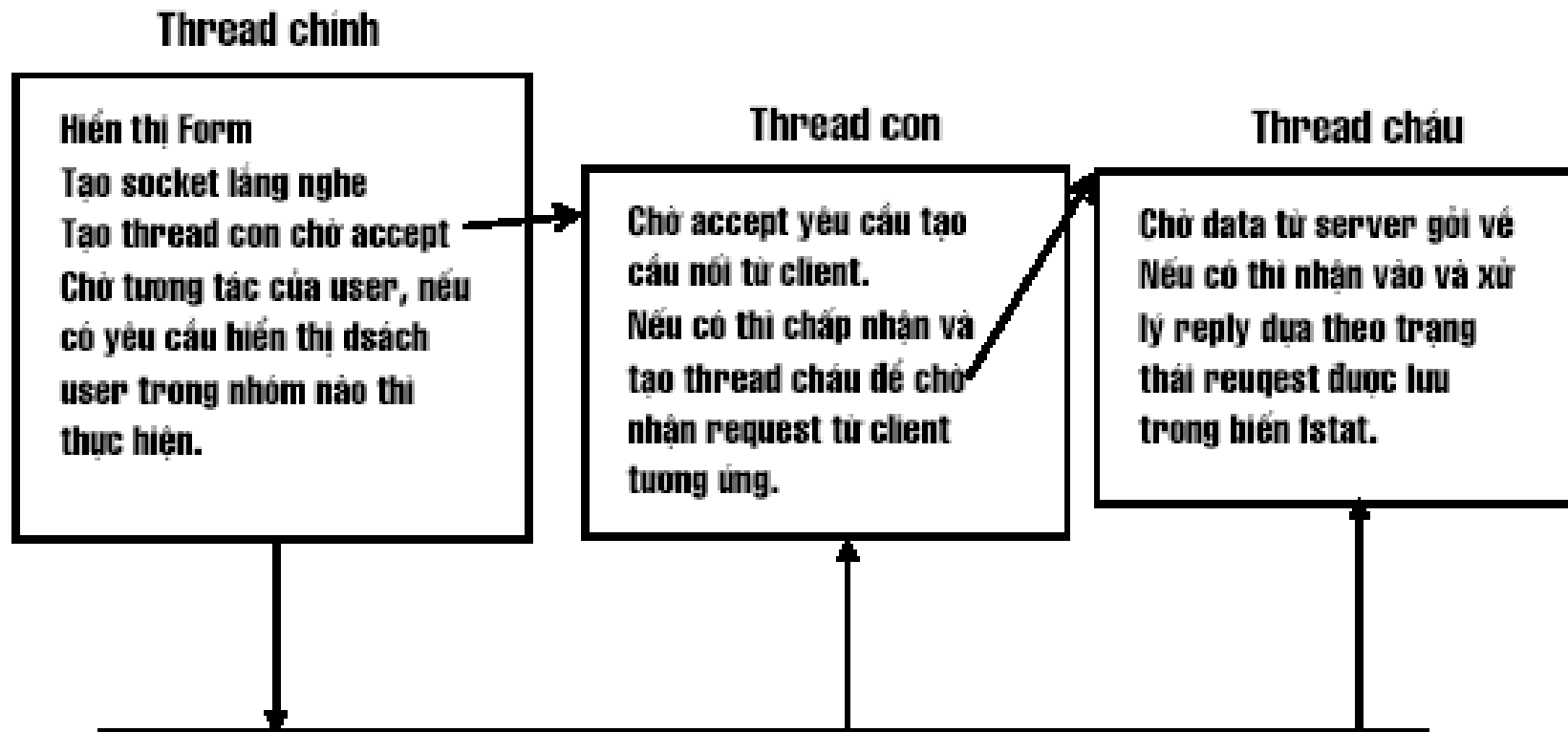


Kỹ Thuật Java Multithread

```
1. class PrimeRun implements Runnable {
2.     long minPrime;
3.     PrimeRun ( long minPrime ) {
4.         this.minPrime = minPrime;
5.     }
6.     public void run() {
7.         // compute primes larger than minPrime
8.         ...
9.     }
10. }
11. PrimeRun p = new PrimeRun(143);
12. new Thread(p).start();
```

MiniChat Server Sử Dụng Java

Mô hình các thread ở Server :





MiniChat Server Sử Dụng Java

```
//Constructor của frame
public MiniChatServerDlg() {
    ...
// Tao sersersocket lang nghe cho server
    try {
        serverSocket = new ServerSocket( SERVER_PORT, 100 );
        DefaultListModel lmContent = (DefaultListModel)jlbContent.getModel();
        lmContent.addElement("Server listening on port " + SERVER_PORT + "
        ...");
// tạo thread con để chờ
        new ServerAcceptThread(this,serverSocket).start();
    } // end try
// handle exception creating server and connecting clients
    catch ( IOException ioException ) {
        ioException.printStackTrace();
    }
    ...
}
```



MiniChat Server Sử Dụng Java

```
public class ServerAcceptThread extends Thread {
    ServerSocket serverSocket;
    MiniChatServerDlg serverChat;
    public ServerAcceptThread(MiniChatServerDlg server, ServerSocket sock) {
        serverSocket = sock;
        serverChat = server;
    }
    public void run() {
        T_UserRec puser;
        try {
            // listen for clients constantly
            while (true) {
                // accept new client connection
                Socket clientSocket = serverSocket.accept();
                puser = new T_UserRec();
                puser.sock = clientSocket;
                puser.next = serverChat.m_sock_no_user;
                serverChat.m_sock_no_user = puser;
            }
        }
    }
}
```



MiniChat Server Sử Dụng Java

```
// create new ReceivingThread for receiving messages from client
new ReceivingThread(serverChat, clientSocket).start();
// print connection information
DefaultListModel lmContent =
(DefaultListModel)serverChat.jlbContent.getModel();
lmContent.addElement("Connection received from: " +

        clientSocket.getInetAddress());
serverChat.SendMessage(clientSocket,"Request accepted");
} // end while
}
// handle exception creating server and connecting clients
catch ( IOException ioException ) {
    ioException.printStackTrace();
}
}
}
```



TỔNG KẾT

- Những yêu cầu cơ bản của chương trình MiniChat
- Khung chương trình MiniChat Server sử dụng Java