

**Bộ Công Thương**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HCM**

**----🙣🕮🙡----**



**Đồ Án**

**Môn: Phát Triển Ứng Dụng**

**Đề Tài: Xây dựng hệ thống chat room**



***Giáo Viên Hướng Dẫn:***

**Huỳnh Thái Học**

|  |  |
| --- | --- |
| ***Sinh viên thực hiện:***  **Họ và tên:** | **MSSV:** |
| **Nguyễn Công Luật** | **15076361** |
| **Mai Quốc Khánh** | **15074761** |
| **Nguyễn Nghĩa Minh Nam** | **15071451** |

**Link GitHub: https://github.com/luat4568/Topic1\_Ca1\_NguyenCongLuat**

**Mục Lục**

[**Mục 1: Tổng Quan** 3](#_Toc494748938)

[**1.1 ) Tên đề Tài và nhiệm vụ** 3](#_Toc494748939)

[**1.2 ) Tiến độ thực hiện của tất cả các tuần.** 3](#_Toc494748940)

[**Mục 2: Chi tiết xây dựng hệ thống Chat Room.** 6](#_Toc494748941)

[**2.1) Mô Hình Tổng Thể Hệ Thống Chat** 6](#_Toc494748942)

[**2.1.1:Bộ giao thức TCP/IP** 6](#_Toc494748943)

[**2.1.2: Mô hình Client/Server** 7](#_Toc494748944)

[**2.1.3: Cơ chế Socket trong Java** 10](#_Toc494748945)

[**2.2) Thao tác chat ban đầu giữa Client và Server** 12](#_Toc494748946)

[**2.2.1: Client** 12](#_Toc494748947)

[**2.2.2: Server** 12](#_Toc494748948)

[**2.2.3: ThreadSocket** 13](#_Toc494748949)

[**2.2.4: Kết quả** 14](#_Toc494748950)

[**Mục 3: Kết Luận** 16](#_Toc494748951)

[**Tài Liệu Tham Khảo** 17](#_Toc494748952)

**Mục 1: Tổng Quan**

**1.1 ) Tên đề Tài và nhiệm vụ**

Đề tài: Xây dựng chương trình Chat Room theo mô hình Client-sever

Nhiệm vụ: hoàn thành xây dựng chương trình Chat Room theo mô hình Client-Sever. Đảm bảo cho người dùng có thể dễ dàng sử dụng với giao diện đã thiết đơn giản, gần gũi. Cải tiến thêm các chức năng cho người dùng nếu có thể.

**1.2 ) Tiến độ thực hiện của tất cả các tuần.**

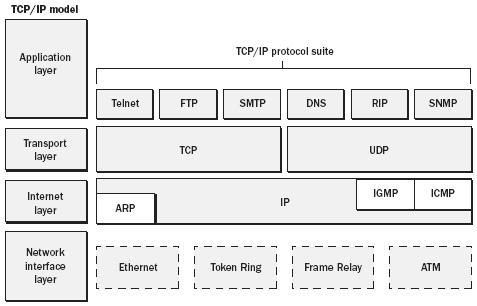
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thời Gian | Người thực hiện | Công việc thực hiện | Khó Khăn | Tiến độ |
| Ngày 1 (15/8/2017) | Thành viên nhóm | Tìm hiểu ý nghĩa các vấn đề sau:  -TCP/IP  -Socket:  +Khái quát về socket  +Cơ chế socket  +Mô hình truyền tin socket  +Một số hàm cơ bản trong  socket  -Thread  -Mô hình Client-Sever | Khái niệm mới nên vẫn chưa hiểu hết được. Nhất là về mô hình client-sever. | Tiến độ :100%  Hoàn thành việc nghiên cứu tài liệu. |
| Ngày 2 (22/8/2017) | Thành viên nhóm | Tìm hiểu hướng đối tượng trong JAVA:  +Học và thực hành các bài tập Java cơ bản trên môi trườn Eclips hoặc Netbean.  +Đọc tài liệu tìm hiểu được cách để viết chương trình chat room bằng ngôn ngữ Java  +Thực hành các dạng bài tập:  -Viết chương trình giải phương trình bậc 2  -Viết chương trình quản lý CD. | Một số thành viên chưa học qua lập trình Java nên tốn khoản thời gian khá nhiều cho việc đọc và học Ngôn ngữ Java. | Tiến độ: 100%  Hoàn thành được hơn một nữa các bài tập thực hành đã đề ra. |
| Ngày 3 (29/8/2017) | Nguyễn Công Luật(1)  Mai Quốc Khánh(2)  Nguyễn Nghĩa Minh Nam(3) | -Tiếp tục thực hành lập trình Java.  -Hoàn tất các bài của tuần trước  -Tham khảo, tìm hiểu các thực hiện chương trình chat client-sever cơ bản.  (3)  -Viết chương trình chat client-sever cơ bản(chưa có giao diện)  (1)(2) | -Khó khăn trong việc tiếp xúc các hàm, thuộc tính để viết các kết nối của chat. | Tiến độ: 100%  Hoàn thành các bài tập Java cơ bản còn lại.  Bắt đầu thiết kế chương trình chat cilent-sever cơ bản. |
| Ngày 4 (5/9/2017) | Nguyễn Công Luật(1)  Mai Quốc Khánh(2)  Nguyễn Nghĩa Minh Nam(3) | -Tiếp tục thiết kế chương trình chat client-sever cơ bản.  (1),(2).  -Học và ôn tập cách thiết kế giao diện(1,2,3):  +Học cách sử dụng các cấu trúc câu, thuộc tính.  +Thực hành các bài giao diện mẫu đơn giản.  -Phân tích cách thiết kế giao diện chat room(1):  +Thiết kế mẫu giao diện chat room bằng hình ảnh | Phần thiết kế giao diện gặp phải khá nhiều thuộc tính mới, nên việc làm quen củng như thực hành gặp nhìu khó khăn, tốn nhiều thời gian | Tiến độ: 80%  -Phần thiết kế giao diện mẫu bằng hình ảnh hoàn thành.  -Phần thiết kế chương trình chat , và học cách thiết kế giao diện vẫn chưa hoàn thành.  \*thời gian thực hiện công việc bị chậm trễ |
| Ngày 5 (12/9/2017) | Nguyễn Công Luật(1)  Mai Quốc Khánh(2)  Nguyễn Nghĩa Minh Nam(3) | -Thiết kế giao diện chat.(1,2)  -Nghiên cứu thêm tài liệu về giao diện hổ trợ cho 2 thành viên còn lại(3). | Thiết kế theo mẫu đã vẽ sẵn nhưng vẫn chưa đc hoàn thiện. | Tiến độ: hoàn thành 100% của tuần 4. Và 50% tiến độ của tuần 5. |
| Ngày 6 (19/9/2017) | Nguyễn Công Luật(1)  Mai Quốc Khánh(2)  Nguyễn Nghĩa Minh Nam(3) | Tiếp tục thiết diện hoàn chỉnh(1,2)  -Tham khảo tài liệu liên kết giao diện, làm liên kết giao diện chat với client-sever(3) | Phần giao diện xác định được bố cục, nhưng khi chạy thì các button bị lệch.  Phần tài liệu liên kết giao diện chat phải tham khảo các nguồn tài liệu nước ngoài nên gặp phải một chút trở ngại. | Tiến độ: 70%  -Phần thiết kế giao diện hoàn thành. Nhưng liên kết vẫn chưa hoàn thành.  \*Tiến độ bị chậm trễ. |
| Ngày 7 (26/9/2017) | Nguyễn Công Luật(1)  Mai Quốc Khánh(2)  Nguyễn Nghĩa Minh Nam(3) | Tiếp tục code phần kết nối client-sever(3) |  | Chưa thực hiện |
| Ngày 8 (3/10/2017) | Thành viên nhóm | Hợp phần code thiết kế với code kết nối client-sever lại, điều chỉnh để có thể chạy được giao diện hoàn chỉnh có thể kết nối các máy với nhau. |  | Chưa thực hiện |
| Ngày 9 (10/10/2017) | Thành viên nhóm | Kiểm tra chương trình chat room, ghi nhận các lỗi sự cố xảy ra và tìm các khắc phục. |  | Chưa thực hiện |
| Ngày 10 (17/10/2017) | Thành viên nhóm | Hoàn tất việc xây dựng chương trình chat room. Ngoài ra có thể đề xuất xây dựng thêm:  -Gửi file.  -Chat Camera, Icon. |  | Chưa thực hiện |

**Mục 2: Chi tiết xây dựng hệ thống Chat Room.**

**2.1) Mô Hình Tổng Thể Hệ Thống Chat**

**2.1.1:Bộ giao thức TCP/IP**

* TCP/IP là một chuẩn phổ biến mà các mạng nội bộ và diện rộng có thể giao tiếp, cho phép các máy tính kết nối với nhau và cho các ứng dụng để gửi dữ liệu đi và về.
* Có 4 lớp trừu tượng khác nhau trong TCP/IP bao gồm:
  + Application
  + Transport
  + Internet
  + Network
* Các giao thức tầng Application
  + Trong TCP/IP, các giao thức tầng application hoạt động trong 3 tầng của mô hình OSI : session, presentation, application
  + Các giao thức : TELNET, DNS, HTTP, FTP, SMTP, …
* Các giao thức tầng Transport bao gồm : TCP, UDP
* Các giao thức tầng Internet (Network) bao gồm : IP, ARP, RARP, ICMP, IGMP
* Các giao thức tầng Network
  + TCP/IP không định nghĩa giao thức riêng ở tầng này.
  + Sử dụng những giao thức có sẵn: DSL, SONET, 802.11, Ethernet

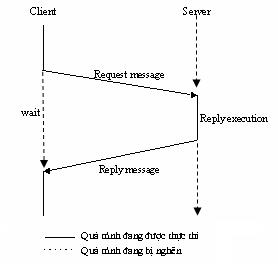
  
*Hình 1: Sơ đồ TCP/IP (http://hus.edu.vn)*

**2.1.2: Mô hình Client/Server**

-Trong mô hình này, chương trình ứng dụng được chia thành 2 thành phần:

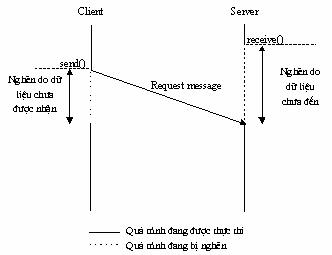
* Quá trình  chuyên cung cấp một số phục vụ nào đó, chẳng hạn: phục vụ tập tin, phục vụ máy in, phục vụ thư điện tử, phục vụ Web... Các quá trình này được gọi là các trình phục vụ hay **Server**.
* Một số quá trình khác có yêu cầu sử dụng các dịch vụ do các server cung cấp được gọi là các quá trình khách hàng hay **Client**.

-Việc giao tiếp giữa client và server được thực hiện dưới hình thức trao đổi các thông điệp (Message). Để được phục vụ, client sẽ gởi một thông điệp yêu cầu (Request Message) mô tả về công việc muốn server thực hiện. Khi nhận được thông điệp yêu cầu, server tiến hành phân tích để xác định công việc cần phải thực thi. Nếu việc thực hiện yêu cầu này có sinh ra kết quả trả về, server sẽ gởi nó cho client trong một thông điệp trả lời (Reply Message). Dạng thức (format) và ý nghĩa của các thông điệp trao đổi giữa client và server được qui định rõ bởi giao thức (protocol) của ứng dụng.



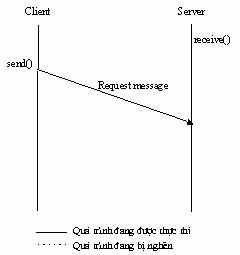
*Hình 2: Mô hình TCP/IP (https://voer.edu.vn )*

* Các chế độ trong giao tiếp: Quá trình giao tiếp giữa client và server có thể diễn ra theo hai chế độ là nghẽn (blocked) hay không nghẽn (Non blocked).
  + Chế độ nghẽn: Trong chế độ này, khi quá trình client hay server phát ra lệnh gởi dữ liệu,  (thông thường bằng lệnh send) , sự thực thi của nó sẽ bị tạm dừng cho đến khi quá trình nhận  phát ra lệnh nhận số dữ liệu đó (thường là lệnh receive). Tương tự cho trường hợp nhận dữ liệu, nếu  quá trình nào đó, client hay server, phát ra lệnh nhận dữ liệu, mà ở thời điểm đó chưa có dữ liệu gởi đến, sự thực thi của nó cũng tạm dừng cho đến khi có dữ liệu gởi đến.



*Hình 3: Chế độ nghẽn (voer.edu.vn)*

* + Chế độ không nghẽn: Trong chế độ này, khi quá trình client hay server phát ra lệnh gởi dữ liệu, sự thực thi của nó vẫn được tiếp tục mà không quan tâm đến việc có quá trình nào phát ra lệnh nhận số dữ liệu đó hay không. Tượng tự cho trường hợp nhận dữ liệu, khi quá trình phát ra lệnh nhận dữ liệu, nó sẽ nhận được số lượng dữ liệu hiện có (bằng 0 nếu chưa có quá trình nào gởi dữ liệu đến). Sự thực thi của quá trình vẫn được tiếp tục. Trong thực tế cần chú ý đến chế độ giao tiếp nghẽn khi lập trình, vì nó có thể dẫn đến trường hợp chương trình bị "treo" do số lần gởi và nhận giữ liệu không bằng nhau giữa hai bên giao tiếp.



*Hình 4: Chế độ giao tiếp không nghẽn (voer.edu.vn)*

**2.1.3: Cơ chế Socket trong Java**

Một socket là một điểm cuối của thông tin hai chiều liên kết giữa hai chương trình đang chạy trên mạng. Những lớp socket được dùng để đại diện cho kết nối giữa một chương trình client và một chương trình server. Trong Java gói Java.net cung cấp hai lớp Socket và ServerSocket để thực hiện kết nối giữa client và server.

Thông thường thì server sẽ chạy trên một máy đặc biệt và có một socket giới hạn trong 1 Portnumber đặc biệt.

Phía client: client được biết hostname của máy mà server đang chạy và port number mà server đang lắng nghe. Để tạo một yêu cầu kết nối client sẽ thử hẹn gặp server ở trên máy của server thông qua port number. Client cũng cần xác định chính nó với server thông qua local port number.

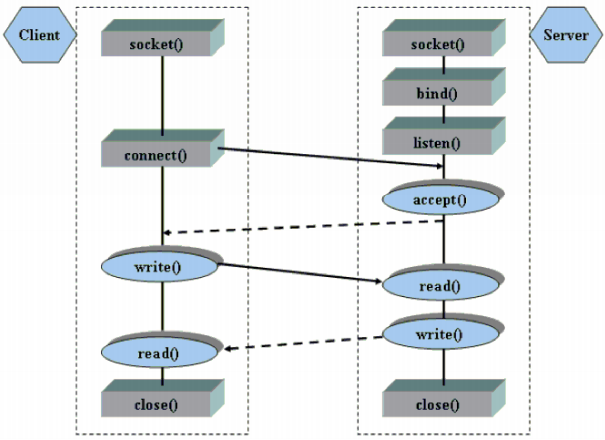


*Hình 5: Client gửi yêu cầu kết nối tới Server*

Nếu mọi thứ tốt đẹp thì server sẽ đồng ý kết nối. khi đồng ý kết nối thì server sẽ tạo ra một socket mới để nói chuyện với client và cũng tạo ra một socket khác để tiếp tục lắng nghe.



*Hình 6 : Server đồng ý kết nối và tiếp tục lắng nghe.*

****

*Hình 7: Mô hình truyền tin Socket (http://www.cit.ctu.edu.vn)*

* Một socket có thể thực hiện bảy thao tác cơ bản:
  + Kết nối với một máy ở xa
  + Gửi dữ liệu Nhận dữ liệu
  + Ngắt liên kêt
  + Gán cổng
  + Nghe dữ liệu đến
  + Chấp nhận liên kết từ các máy ở xa trên cổng đã được gán

**2.2) Thao tác chat ban đầu giữa Client và Server**

**2.2.1: Client**

public class Client {

public static void main(String args[])

{

try {

String loigui;

String ketQua;

BufferedReader inFormUser= new BufferedReader(new InputStreamReader(System.*in*));

Socket clientSocket= new Socket("127.0.0.1",9999);

DataOutputStream sendToServer= new DataOutputStream(clientSocket.getOutputStream());

BufferedReader inFromServer = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

while(true)

{

loigui = inFormUser.readLine();

sendToServer.writeBytes(loigui+'\n');

if (loigui.equalsIgnoreCase("quit"))

break;

ketQua = inFromServer.readLine();

System.*out*.println("FROM SERVER: "+ketQua);

}

clientSocket.close();

} catch (IOException e) {

System.*out*.println("Exception Client: "+e.getMessage());

}

}

}

**2.2.2: Server**

public class Server {

public static void main(String args[])

{

try {

ServerSocket ss = new ServerSocket(9999);

while (true)

{

new ThreadSocket(ss.accept()).start();

}

} catch (IOException e) {

System.*out*.println("Exception: " + e.getMessage());

} }}

**2.2.3: ThreadSocket**

public class ThreadSocket extends Thread{

Socket socket= null;

public ThreadSocket(Socket socket)

{

this.socket=socket;

}

public void run()

{

try {

DataOutputStream sendToClient= new DataOutputStream(socket.getOutputStream());

BufferedReader fromClient= new BufferedReader(new InputStreamReader(socket.getInputStream()));

while (true)

{

String sentence=fromClient.readLine();

System.*out*.println("FROM CLIENT: " + sentence);

if (sentence.equalsIgnoreCase("quit"))

break;

String reverseSentence= reverse(sentence);

sendToClient.writeBytes(reverseSentence+'\n');

}

} catch (Exception e) {

e.printStackTrace();

}

}

public String reverse(String input) throws InterruptedException

{

String output="";

StringBuilder strBuilder= new StringBuilder(input);

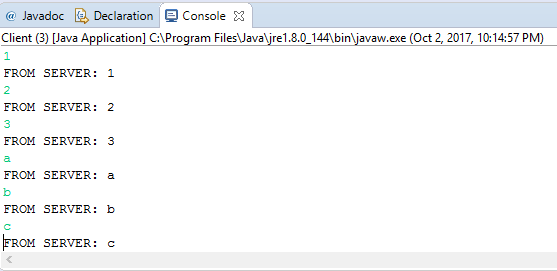
output=strBuilder.reverse().toString();

return output;

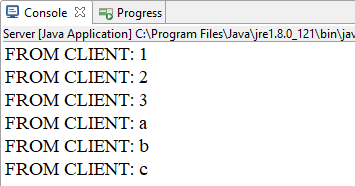
}

}

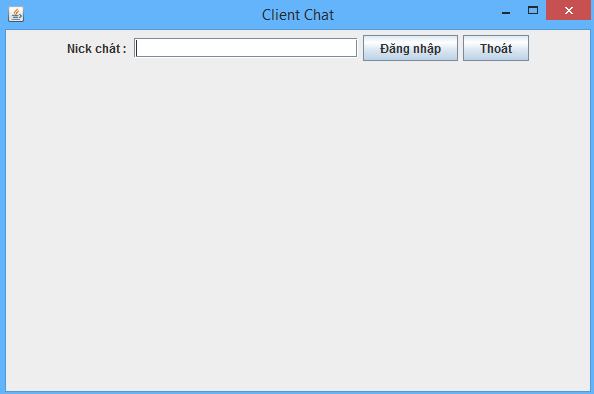
**2.2.4: Kết quả**



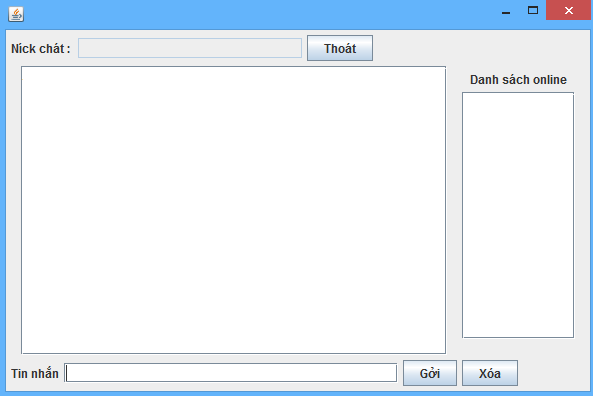
Hình Client Gửi nội dung chat lên Sever



Hình Sever nhận tin chat từ phía Client gửi lên



Hình giao diện đăng nhập tên chat demo



Hình giao diện chat demo

**Mục 3: Kết Luận**

● Công việc làm được:

* + - Xây dựng chương trình chat room cơ bản, chương trình chat room có sử dụng TCP/IP, Threads và Socket.
    - Viết được giao diện chat room.

● Tiến độ công việc:

* + - Hoàn thành được 60% tiến độ đề ra.

● Công việc chưa hoàn thành:

* + - Code phần thiết kế giao diện chưa bắt được sự kiện của code phần kết nối client – server.
    - Server chưa nhận được địa chỉ ip trong 1 mạng LAN.

● Công việc cần làm:

* + - Xem và làm lại bài tập về code phần thiết kế giao diện bắt sự kiện với code phần xử lý.
    - Hoàn thành code phần kết nối client-sever
    - Hợp phần code thiết kế với code kết nối client-sever lại, điều chỉnh để có thể chạy được giao diện hoàn chỉnh có thể kết nối các máy với nhau.

**Tài Liệu Tham Khảo**

**https://sites.google.com/site/huynhhoc/distributed-object**

[**http://www.toancau.vn/vietnam/4/5/32/giao-thuc-tcpip-la-gi**](http://www.toancau.vn/vietnam/4/5/32/giao-thuc-tcpip-la-gi)

[**http://hus.edu.vn/thcs/bai10.html**](http://hus.edu.vn/thcs/bai10.html)

[**https://voer.edu.vn/c/mo-hinh-client-server/761b0302/eedbe7c9**](https://voer.edu.vn/c/mo-hinh-client-server/761b0302/eedbe7c9)

[**http://www.cit.ctu.edu.vn/~dtnghi/netprog/4a-Socket.pdf**](http://www.cit.ctu.edu.vn/~dtnghi/netprog/4a-Socket.pdf)

[**http://www.cit.ctu.edu.vn/~dtnghi/netprog/4a-Socket.pdf**](http://www.cit.ctu.edu.vn/~dtnghi/netprog/4a-Socket.pdf)