TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────



**BÀI TẬP LỚN**

Môn: Các hệ phân tán – IT4371

Đề tài 1: MySSH

**Giáo viên hướng dẫn: TS. Nguyễn Bình Minh**

**Sinh viên thực hiện**

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **MSSV** | **Lớp** |
| Bùi Tiến Thành | 20144052 | CNTT2.2-K59 |
| Võ Văn Tài | 20143927 | CNTT2.2-K59 |
| Nguyễn Thị Dung | 20140698 | CNTT2.1-K59 |

***Hà Nội, tháng 5 năm 2017***

Lời mở đầu

Cách phổ biến nhất để kết nối với máy chủ Linux từ xa là thông qua SSH. SSH là viết tắt của Secure Shell và cung cấp một cách an toàn và an toàn để thực hiện các lệnh, thay đổi và cấu hình các dịch vụ từ xa.

Vì hứng thú và muốn tìm hiểu về SSH nên nhóm chúng em chọn đề tài viết chương trình MySSH sử dụng Java Socket để mô phỏng chương trình SSH chạy trên Windown.

Chúng em xin gửi lời cảm ơn đến thầy Nguyễn Bình Minh đã giải đáp thắc mắc và giúp đỡ chúng em trong quá trình thực hiện đề tài.

Đây là lần đầu chúng em tìm hiểu và làm việc với Java Socket, cũng như các khái niệm liên quan đến SSH nên chương trình sẽ còn nhiều thiếu sót. Nhóm chúng em xin tiếp thu những lời khuyên và cải thiện để chương trình hoàn thiện hơn.

MỤC LỤC

[**Phần 1 Giới thiệu về SSH và lập trình Socket** 4](#_Toc481943486)

[**I.** **Lập trình Socket** 4](#_Toc481943487)

[Giới thiệu chung về Socket 4](#_Toc481943488)

[Socket hoạt động như thế nào? 4](#_Toc481943489)

[**II.** **Giới thiệu SSH** 4](#_Toc481943490)

[**Phần 2 Xây dựng chương trình MySSH sử dụng Socket** 5](#_Toc481943491)

[**I.** **Giới thiệu về MySSH** 5](#_Toc481943492)

[**II.** **Thiết kế giải pháp** 5](#_Toc481943493)

[Phía Server: 5](#_Toc481943494)

[Phía Client: 6](#_Toc481943495)

[**III.** **Hướng dẫn cấu hình và sử dụng MySSH** 6](#_Toc481943496)

[Chạy Server 6](#_Toc481943497)

[Chạy Client 7](#_Toc481943498)

[**Phần 3 Phân công công việc trong nhóm để hoàn thành Project** 8](#_Toc481943499)

[**Phần 4 Tài liệu tham khảo** 8](#_Toc481943500)

# **Phần 1 Giới thiệu về SSH và lập trình Socket**

1. **Lập trình Socket**

### Giới thiệu chung về Socket

-Một socket là một điểm cuối của một giao tiếp 2 chiều giữa 2 chương trình chạy trên mạng. Socket được ràng buộc với một cổng (con số cụ thể) để các tầng TCP có thể định danh ứng dụng mà dư liệu sẽ được gửi tới

### Socket hoạt động như thế nào?

-Tại phía Server: Thông thường, một chương trình server chạy trên một máy tính cụ thể, chương trình này có một ổ cắm (Server socket), ổ cắm được ràng buộc bởi cổng (Port number) cụ thể. Các chương trình phục vụ (Server program) chỉ chờ đợi, lắng nghe tại ổ cắm (Server socket) các client để thực hiện một yêu cầu kết nối.

-Tại phía Client: Các Client biết tên máy của máy tính mà trên đó chương trình chủ (server) đang chạy và số cổng mà chương trình chủ lắng nghe. Để thực hiện một yêu cầu kết nối, Client cố gắng tạo ra cuộc gặp với máy chủ trên máy tính của chương trình chủ và cổng. Các Client cũng cần phải tự định danh chính nó với server để gắn với một cổng địa phương cái sẽ được sử dụng trong suốt quá trình kết nối này, thông thường nó được gán bởi hệ điều hành.

-Nếu mọi việc suôn sẻ, chương trình chủ (server program) chấp nhận kết nối của client. Khi chấp nhận, máy chủ có được một socket mới bị ràng buộc vào cùng "cổng địa phương" và thông tin đầu cuối (remote endpoint) của nó chính là địa chỉ và cổng của client. Nó đã tạo ra một socket mới để chăm sóc Client vừa được chấp nhận kết nối, và tiếp tục lắng nghe tại ổ cắm gốc ban đầu (ServerSocket) cho các yêu cầu kết nối khác. Về phía Client, nếu kết nối được chấp nhận, một ổ cắm được tạo thành công và Client có thể sử dụng ổ cắm để giao tiếp với chương trình chủ.

-Các Client và Server có thể giao tiếp bằng cách ghi hay đọc từ ổ cắm (Socket) của chúng.

1. **Giới thiệu SSH**

-SSH là một giao thức bảo mật được sử dụng làm phương tiện chính để kết nối đến các máy chủ từ xa. Nó cung cấp một giao diện dựa trên văn bản bằng cách sinh ra remote shell. Sau khi kết nối, tất cả các lệnh bạn nhập vào thiết bị đầu cuối cục bộ của bạn sẽ được gửi đến máy chủ từ xa và thực hiện ở đó.

# **Phần 2 Xây dựng chương trình MySSH sử dụng Socket**

1. **Giới thiệu về MySSH**

-Chương trình MySSH sử dụng Java Socket để giao tiếp giữa Clients và Server, lập trình và chạy thử trên hệ điều hành Windown. Client sẽ gửi yêu cầu kết nối tới Server và gửi lệnh cho Server phân tích và thực hiện trên máy chủ. Sau khi thực hiện lệnh Server sẽ gửi phản hồi lại cho Client, có thể là một thông báo hoặc báo lỗi.

-Tất cả dịch vụ và các hàm xử lý yêu cầu được lập trình ở phía Server, Client chỉ có nhiệm vụ gửi yêu cầu đến Server và nhận phản hồi từ Server. Chi tiết được trình bày ở các phần sau.

1. **Thiết kế giải pháp**

### Phía Server:

-Server sẽ tạo một ServerSocket tại cổng mặc định (số hiệu cổng có thể thay đổi trong file ServerController.Configs.java). ServerSocket sẽ lằng nghe kết nối và chấp nhận. Khi đó một Socket mới tạo ra kết nối với Socket tại Client, Socket đó được quản lý bởi nested class ServiceThread. Đối tượng ServiceThread cũng được tạo ra để quản lý mỗi khi một Socket mới được tạo ra.

-Khí đó Server có thể phục vụ được nhiều Client bằng cách tạo và quản lý nhiều Socket và ServiceThread mới với mỗi yêu cầu kết nối. Server chỉ tạo tối đa một số lượng Socket tương ứng với số Client (số lượng tối đa Socket có thể thay đổi trong file ServerController.Configs.java).

-Khi Client kết nối tới sẽ được yêu cầu đăng nhập (thông tin đăng nhập được lưu ở file UserInformation\user-information.txt ở máy Server). Chỉ khi đăng nhập thành công Client mới được phép yêu cầu các lệnh trên Server.

-Với mỗi Client kết nối đến, Server sẽ tạo 2 luồng chạy song song với nhau để giao tiếp với Client với Socket là điểm đầu cuối: Luồng input stream để nhận request của Client và luồng output stream để gửi response đến Client.

-Với mỗi Client, Server sẽ tạo một đối tượng ServerService để thực hiện và đưa ra result để gửi lại cho Client. Như vây nhiều Client có thể song song gửi yêu cầu và nhận kết quả trả về từ Server.

-Để thuận tiện trong việc lập trình và giúp chương trình dễ mở rộng, lớp QueryIO.java được tạo ra để lưu thông tin yêu cầu từ Client, bao gồm tên lệnh, danh sách tham số và kết quả trả về (kết quả trả về có 2 loại, dạng String cho thông báo từ Server và dạng bytes[] phục vụ việc gửi file).

-Khi nhận được request từ Client, ServiceThread sẽ phân tích để lấy ra tên lệnh và danh sách tham số lưu vào đối tượng QueryIO.java. Đối tượng ServerService sẽ thực hiện lệnh và lưu kết quả trả về dựa trên tên lệnh và danh sách tham số (nếu có).

### Phía Client:

-Client sẽ tạo một Socket để kết nối tới Server, với điều kiện biết địa chỉ ip của Server trong mạng và số hiệu cổng kết nối (Số hiệu cổng kết nối bên server thường sẽ cố định nên chỉ có địa chỉ ip là được nhập bằng tay, có thế thay đổi số hiệu cổng kết nối ở file ClientController.Client.java).

-Tương tự bên Server, Client sẽ tạo 2 luồng chạy song song: Luồng input stream để nhận response của Server và luồng output stream để gửi request đến Server.

1. **Hướng dẫn cấu hình và sử dụng MySSH**

-Cần chạy Server trước khi chạy Client, chạy bằng IDE(Khuyến khích sử dụng NetBeans)

### Chạy Server

-Chạy file ServerController.Server.java ở máy Server. Khi màn hình consol hiện dòng chữ “Watting for connection...” là đã chạy thành công, Server sẵn sàng để nhận kết nối.

-Mỗi khi có một Client kết nối đến, màn hình consol của Server sẽ hiển thị tên và địa chỉ ip của Client. Khi có Client ngắt kết nối hoặc mất kết nối, một thông báo cũng sẽ hiển thị.

### Chạy Client

-Chạy file ClientController.Client.java. Nhập địa chỉ ip của máy Server khi được yêu cầu. Sau khi kết nối thành công sẽ tiến hành đăng nhập, có thể dùng taikhoan là admin, matkhau là admin và nhập theo cú pháp yêu cầu.

Sau khi có thông báo đăng nhập thành công, Client có thể gửi yêu cầu tới Server dưới dạnh lệnh. Danh sách lệnh xem bằng lệnh help.

-Khi muốn ngắt kết nối tới Client, gõ lệnh QUIT

# **Phần 3 Phân công công việc trong nhóm để hoàn thành Project**

|  |  |
| --- | --- |
| Bùi Tiến Thành | Thiết kể các lớp, tạo luồng giao tiếp giữa client và server, viết lệnh hiển thị path(getpath, removepath), lệnh gửi file (get), viết báo cáo. |
| Võ Văn Tài | Viết các lệnh còn lại, hoàn thiện lệnh hiển thị path, hoàn thiện lệnh gửi file khi chia nhỏ file. |
| Nguyễn Thị Dung | Hoàn thiện luồng giao tiếp bên client, tìm và sửa lỗi phát sinh, hoàn thiện chương trình để có thể chạy được trên 2 máy. |

# **Phần 4 Tài liệu tham khảo**

-Tham khảo source code lập trình java socket cơ bản và các khái niệm về socket: <http://o7planning.org/vi/10393/huong-dan-lap-trinh-java-socket>

-Khái niệm về SSH: <https://www.digitalocean.com/community/tutorials/ssh-essentials-working-with-ssh-servers-clients-and-keys>