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: Xây dựng MySSH

**GV Hướng Dẫn:** TS. Nguyễn Bình Minh

**Nhóm sinh viên thực hiện:**

Lương Tuấn Dũng 20140793

Võ Mạnh Cường 20140635

Vũ Trọng Hiệu 20141683

Mục Lục

1. **Mô tả bài toán**

* Ứng dụng mySSH mô phỏng chương trình “ssh” để Client kết nối từ xa đến Server thực hiện các lệnh hệ thống trên Server dùng Socket, giao thức TCP.
* Server cho phép nhiều Clients kết nối đồng thời (giới hạn N clients, vượt quá N thì không cho phép kết nối )
* Mỗi Clients kết nối tới phải đăng nhập (username / password). CSDL người dùng là của chương trình không phải của hệ thống.
* Server thông báo kết nối thành công hay không khi các Clients kết nối tới.
* Khi kết nối thành công thì client có thể thực hiện các lệnh trên hệ thống của server
* Các lệnh có thể có tham số hoặc không
* Một số lệnh cơ bản: hiển thị thư mực hiện thời, in danh sách các tệp, thư mục trong thư mục hiện thời, xóa, di chuyển tệp, thư mục, hiển thị ngày giờ hệ thống, v.v...
* Server phân tích lệnh nhận được từ phía client, thực hiện và trả kết quả cho client
* Nếu các lệnh thực hiện không được phép chạy thì phải thông báo lỗi.
* Lệnh chạy trên server phụ thuộc vào hệ điều hành mà server được cài đặt lên (Windows, Linux)
* Kết thúc phiên làm việc client gửi lệnh ngắt kết nối tới server và chấm dứt kết nối.

1. Phân chia công việc

|  |  |
| --- | --- |
| Thành viên | Công việc |
| Vũ Trọng Hiệu | Xây dựng client, giao diện, quản lý người dùng; lệnh: userctl |
| Lương Tuấn Dũng | Xử lý gửi, nhận thông điệp giữa client, server; các lệnh : cp, mv, rm |
| Võ Mạnh Cường | Xử lý command; các lệnh: time, echo, ls, mkdir, pwd, cd |

1. Công nghệ sử dụng:

 Sử dụng IDE: NetBean

 Ngôn ngữ lập trình : Java

1. Lập trình Socket với Java

* Socket cung cấp một cơ chế giao tiếp giữa 2 máy tính sử dụng giao thức TCP - Transmission Control Protocol ( giao thức kết nối tin cậy giữa hai ứng dụng). Khi giao tiếp, một máy client sẽ tạo ra một socket ở đầu của nó trong giao tiếp và cố gắng kết nối socket đó tới server. Khi một kết nối được khởi tạo, server sẽ tạo ra một đối tượng socket ở đầu của mình trong giao tiếp, từ đó client và server có thể giao tiếp với nhau thông qua việc ghi và đọc từ socket.
* Lớp chức năng trong java:
  + class ServerSocket sẽ nhận đầu vào là một số hiệu cổng (có thể thêm địa chỉ IP local), sau đó sẽ tạo ra một thể hiện của lớp ServerSocket sẽ gắn với địa chỉ và cổng truyền vào, cung cấp các phương thức, thuộc tính để lắng nghe và thực hiện kết nối với các kết nối tới, gửi và truyền thông điệp tới máy client.
  + class Socket sẽ nhận đầu vào là một địa chỉ IP và số hiệu cổng bất kì, một thể hiện của lớp sẽ được gắn với địa chỉ và cổng đó, đóng vai trò như một endpoint giao tiếp với server tại địa chỉ định sẵn, cung cấp các phương thức, thuộc tính để gửi và truyền thông điệp tới server.
* Các bước thực hiện để khởi tạo một kết nối TCP giữa hai máy sử dụng socket:
  + Server khởi tạo một thể hiện của lớp ServerSocket gắn với một cổng cụ thể sẽ diễn ra giao tiếp
  + Server sẽ gọi đến phương thức accept() của đối tượng lớp ServerSocket. Phương thức này sẽ đợi cho đến khi một client kết nối đến cổng định sẵn
  + Sau khi Server sẵn sang chấp nhận kết nối, Client sẽ khởi tạo một thể hiện của lớp Socket, gắn với một server name và một cổng kết nối tới
  + Client sẽ cố gắng kết nối tới server và port định sẵn, nếu kết nối thành công thì thể hiện Socket đó sẽ được dùng để giao tiếp với server
  + Phía server, khi có kết nối đến thành công, phương thức accept() sẽ trả về một tham chiếu đến một thể hiện socket mới trên server mà được kết nối tới thể hiện socket của Client.
  + Giao tiếp được thực hiện thông qua I/O stream, mỗi socket có 2 đối tượng InputStream và OutputStream. Input của socket ở server là output của socket ở client và ngược lại.
* Như vậy công việc còn lại phần server sẽ là lấy thông tin từ client gửi lên, thực hiện các chức năng tương ứng và trả về thông điệp cho client, cùng với đó là việc quản lý người dùng, số lượng kết nối cũng như quyền hạn.

1. Hướng dẫn cài đặt, chạy chương chình

* Mã nguồn có thể tải về từ github tại địa chỉ: <https://github.com/tset0401/distributed-systems.git>
* Sau khi tải về mã nguồn, mở lên bằng Netbeans IDE, thực hiện build project
* Thay đổi cấu hình trong file server.info để sửa thông tin về cổng server cũng như số lượng client nhiều nhất được phục vụ
* Chạy file DSServer để khởi tạo server lắng nghe tại cổng đã cấu hình, sau đó chạy file TerminalUI để khởi tạo client và vào giao diện dòng lệnh
* Chương trình sẽ mặc định yêu cầu người dùng đăng nhập tài khoản, nếu đăng nhập thành công thì client có thể sử dụng các chức năng của server, ngược lại sẽ không thể dùng
* Để giao tiếp với server, người dùng nhập vào lệnh rồi ấn Enter, lệnh sẽ được gửi đi và chờ nhận về kết quả
* Để thoát chương trình, gõ : exit

1. Tài liệu tham khảo

<http://o7planning.org/vi/10393/huong-dan-lap-trinh-java-socket>

<https://www.tutorialspoint.com/java/java_networking.htm>