**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



**BÀI TẬP LỚN**

MÔN HỌC: LẬP TRÌNH MẠNG

**ĐỀ TÀI:** XÂY DỰNG CHƯƠNG TRÌNH TRẮC NGHIỆM TRỰC TUYẾN TRÊN MẠNG CỤC BỘ

Giảng viên hướng dẫn : **Bùi Trọng Tùng**

Sinh viên thực hiện : **Tô Quang Ngọc**

**Phan Tiến Dũng**

Lớp : **AS1 – K52**

HÀ NỘI, 11/2010

**MỤC LỤC**

[LỜI NÓI ĐẦU 1](#_Toc277883834)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 2](#_Toc277883835)

[Lập trình Socket trên Linux. 3](#_Toc277883837)

[CHƯƠNG 2. XÂY DỰNG CHƯƠNG TRÌNH 5](#_Toc277883838)

I. Mô tả yêu cầu…… …………………………………………………………...5

[I. Xử lý truyền thông 5](#_Toc277883839)

[II. Cấu trúc chương trình 8](#_Toc277883840)

[CHƯƠNG 3. CÀI ĐẶT VÀ KIỂM THỬ 10](#_Toc277883841)

[I. Cài đặt chương trình 10](#_Toc277883842)

[II. Hướng dẫn chạy chương trình 10](#_Toc277883843)

[III. Kiểm thử chương trình 10](#_Toc277883844)

[CHƯƠNG 4. KẾT LUẬN 11](#_Toc277883845)

[I. Về chức năng so với yêu cầu bài toán 11](#_Toc277883846)

[II. Về hiệu năng chương trình 11](#_Toc277883847)

[TÀI LIỆU THAM KHẢO 12](#_Toc277883849)

# LỜI NÓI ĐẦU

1. Giới thiệu đề tài :

Ngày nay với việc công nghệ thông tin bùng nổ thì các kỳ thi cũng dần dần có những đổi thay. Việc áp dụng CNTT vào học tập, thi cử đã dần trở nên phổ biến. Dựa vào những thực tế mà chúng em đã từng trải qua hoặc biết đến như các kỳ thi TOEFL iBT hay các kỳ thi lấy chứng chỉ của IBM đều đã được đẩy lên thi qua mạng. Bởi vậy chúng em đã quyết định lựa chọn đề tài **“Xây dựng chương trình trắc nghiệm trực tuyến trên mạng cục bộ”** này với mong muốn có thể xây dựng được một chương trình nhỏ giúp ích phần nào cho một số hoạt động thi cử tại các trường học.

Trong quá trình thực hiện, chúng em xin chân thành cảm ơn thầy giáo Bùi Trọng Tùng đã nhiệt tình giúp đỡ, giải đáp những thắc mắc, xin cảm ơn các bạn cùng lớp đã có những góp ý để chương trình được hoàn thiện một cách nhanh và chính xác nhất.

1. Phân công công việc :

* Tô Quang Ngọc :
  + Tìm hiểu mô hình client – server trên Linux
  + Phân tích thiết kế cấu trúc dữ liệu, giải thuật cho chương trình
  + Code phần comunication giữa Client và Server
  + Code phần xử lý dữ liệu trên Server với file
  + Kiểm thử, chỉnh sửa code
* Phan Tiến Dũng :
  + Tìm hiểu mô hình client – server
  + Phân tích thiết kế , giải thuật cho chương trình
  + Code phần comunication giữa Client và Server
  + Kiểm thử và thu gom rác, chỉnh sửa lại code

1. Tóm tắt nội dung báo cáo.

Báo cáo gồm có 4 chương:

+ Chương 1: Cơ sở lý thuyết

+ Chương 2: Xây dựng chương trình

+ Chương 3: Cài đặt và kiểm thử

+ Chương 4: Kết luận

Cụ thể nội dung từng chương như sau:

|  |  |
| --- | --- |
| **Chương** | **Nội dung trình bày** |
| Cơ sở lý thuyết | Lập trình Socket trên Linux |
| Xây dựng chương trình | 1. Mô tả yêu cầu 2. Xử lí truyền thông 3. Cấu trúc chương trình |
| Cài đặt và kiểm thử | 1. Cài đặt chương trình 2. Hướng dẫn chạy chương trình 3. Kiểm thử |
| Kết luận | 1. Về chức năng so với yêu cầu bài toán 2. Về hiệu năng chương trình |

# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

## Lập trình Socket trên Linux

* 1. **Socket :**

Socket là một phương pháp để thiết lập kết nối truyền thông giữa một chương trình yêu cầu dịch vụ ( client) và một chương trình cung cấp dịch vụ (server) trên mạng LAN, WAN hay Internet và có thể là giữa những quá trình ngay bên trong máy tính. Mỗi socket có thể được xem như một điểm cuối trong một kết nối. Một socket trên máy yêu cầu dịch vụ có địa chỉ mạng được cấp sẵn để “gọi” một socket trên máy cung cấp dịch vụ. Một khi socket đã được thiết lập phù hợp, hai máy tính có thể trao đổi dịch vụ và dữ liệu.

Một Socket Address bao gồm: host name và port. Ví dụ: socket(127.0.0.1; 80) //cổng dịch vụ của HTTP

Trên Windows, thư viện winsock được nhiều người sử dụng để lập trình mạng trên Windows. Tuy nhiên trong yêu cầu của chương trình là làm việc trong môi trường mã nguồn mở. Chúng em xin được trình bày những nội dung về lập trình C Socket trên UNIX.

* 1. **Các loại Socket :**

Có hai loại socket: stream sockets(SOCK\_STREAM) và datagram sockets(SOCK\_DGRAM).

Stream sockets sử dụng giao thức TCP(Transmission Control Protocol) nên rất đáng tin cậy. Stream thường được dùng để chuyển các dữ liệu lớn trên mạng hoặc khi cần chuyển các gói dữ liệu nối tiếp nhau(thứ tự của các gói dữ liệu(packets) là điều quan trọng nhất). Ví dụ, bạn có 3 gói dữ liệu "A,B,C", stream sockets sẽ chuyển theo đúng thứ tự "A,B,C".

*Stream sockets*(TCP)‏

Datagram sockets cung cấp một phương pháp chuyển dữ liệu đơn giản hơn và nhanh hơn stream sockets. Datagram sockets dựa trên UDP(User Datagram Protocol), không đáng tin cậy so với TCP. Dữ liệu có thể không đến được đích, không nối tiếp nhau(packets được chuyển đi không theo đúng thứ tự) hoặc bị lặp lại nhiều lần(socket đích có thể nhận được một hoặc nhiều packets trùng nhau). HTTP, FTP và một số protocols khác sử dụng stream sockets, trong khi broadcast protocols dùng datagram sockets.

Datagram Socket (UDP)‏

# CHƯƠNG 2. XÂY DỰNG CHƯƠNG TRÌNH

1. **Mô tả yêu cầu**

Yêu cầu của bài toán là xây dựng một mô phỏng cho chương trình trắc nghiệm trực tuyến trên mạng cục bộ .

1. **Đối với phía Client :**
   * Người dùng sử dụng hệ thống lựa chọn thao tác mình muốn thực hiện tại Menu chính, cần phải đăng nhập để bắt đầu vào Test.
   * Nếu đã có tài khoản thì có thể đăng nhập ngay, hoặc nếu chưa có tài khoản có thể đăng ký một tài khoản mới.
   * Gửi câu trả lời tới Server.
   * Thời gian cho mỗi câu hỏi là cố định, ở đây mặc định là 15s (Có thể thay đổi). Khi hết giờ tự động nhảy tới câu tiếp theo.
   * Có thể tải về đề thi và đáp án khi kết thúc.
2. **Đối với phía Server :**
   * Xử lý các yêu cầu từ phía Client như đăng nhập, đăng ký rồi trả về kết quả
   * Đọc câu hỏi từ ngân hàng đề thi và gửi cho phía Client
   * Nhận câu trả lời từ phía Client
   * Tính toán đưa ra kết quả bài thi và gửi lại cho Client
   * Phục vụ xử lý với nhiều Client thi cùng lúc

## Xử lý truyền thông :

Trong bài tập lớn này chúng em sử dụng stream sockets (SOCK\_STREAM) vì các lý do sau:

+) Cần phải đảm bảo tính tin cậy của các gói tin được truyền (các câu hỏi gửi từ server và câu trả lời từ client phải được đảm bảo tính tin cây,đầy đủ và toàn vẹn để không mất mát thông tin)

+) Mặt khác dung lượng các thông tin truyền không quá lớn, không đòi hỏi cao về tốc độ.

**Biểu đồ truyền thông và thực hiện của chương trình**

1. **Chức năng đăng nhập và test :**

false

true

Exit

Recv()

Send()

Recv()

Send()

Send()

Exit

Download answer

Recv()

Compute Result

Send()

Recv()

Recv()

false

Send()

true

Check username and passsword

Recv()

Send()

1. **Chức năng đăng ký :**

Send()

Recv()

Check username is existed?

true

Send()

false

write username and password into file user.txt

Recv()

Send()

false

Check message receieved success

true

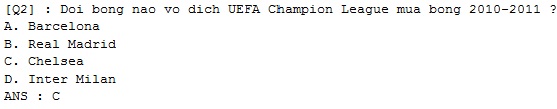
Exit registion

1. **Cấu trúc chương trình :**

Chương trình gồm 2 file chính dùng ở 2 phía Client và Server là **client.c** và **server.c ,** ngoài ra còn một số file dùng để lưu trữ dữ liệu khácnhư **user.txt**, **dethi.txt**, **dapan.txt**

1. Tổ chức lưu trữ dữ liệu :

* Thông tin về người dùng được lưu trữ trong file **user.txt** trên Server. Ví dụ : tqnst 123456 với username = tqnst và password = “123456”
* Ngân hàng câu hỏi được lưu trữ trong file **dethi.txt** trên Server . Ví dụ :

****

* Các đáp án dùng cho việc tính điểm trên Server được lưu trữ trong file **dapan.txt**
* Cấu trúc : struct question{

char question[1024]; //Cau hoi

char a[1024]; //Dap an A

char b[1024]; //Dap an B

char c[1024]; // Dap an C

char d[1024]; // Dap an D

char answer[10];//Dap an dung

};

trong file server.c trên phía Server dùng để lưu trữ câu hỏi khi đọc từ file dethi.txt ra , nó dùng để gửi các câu hỏi cho phía Client.

* Cấu trúc : struct question{

char quest[1024];

};

trong file client.c tại phía Client dùng để lưu trữ các câu hỏi khi nhận được từ phía Server, dùng để in ra màn hình các câu hỏi

* Thư viện liên kết **fields.h** hỗ trợ các thao tác đọc file

1. **Một số hàm xử lý trên Client và Server :**
2. **Ở phía Server :**
   * + void readfile(); //Doc file
     + void sendquestion(); // Gui cau hoi toi Client
     + int checklogin(char \*username,int sock\_client); // Kiem tra tai khoan dang nhap
     + int checksignup(char \*username,int sock\_client);// Kiem tra tai khoan dang ky
     + void chamdiem(char \*dapan); // Cham diem khi Client gui dap an len Server
     + void sendsubject(int sock\_client); // Gui dap an va cau hoi toi Client neu co yeu cau
   * Khi Client gửi thông tin đăng nhập lên phía Server, Server nhận thông tin đăng nhập và gọi tới hàm checklogin() . Hàm này trả về 0 nếu đăng nhập thất bại, trả về 1 nếu đăng nhập thành công.
   * Khi Client gửi thông tin đăng ký tới Server, Server nhận thông tin và gọi tới hàm checksignup() để kiểm tra xem tài khoản đã tồn tại hay chưa. Nếu tồn tại thì gửi lỗi về cho Client, nếu chưa tồn tại tiến hành ghi file và gửi thông báo thành công tới Client.
   * Sau khi đăng nhập thành công, Server gọi hàm sendquestion() để gửi câu hỏi tới Client.
   * Sau khi nhận câu trả lời từ phía Client, Server gọi hàm chamdiem() để chấm thi và gửi trả lại Client kết quả số câu đúng, số câu sai và số câu bỏ qua.
   * Nếu có yêu cầu nhận đề và đáp án từ Client, Server gọi hàm sendsubject() để gửi đề kèm đáp án và kết quả bài thi của Client
3. **Phía Client :**
   * + void login(char username[],char \*passwd,int sock); // Ham dang nhap tai khoan
     + void signup(int sock); //Ham dang ky tai khoan
     + void sendanswer(int sock); // Gui cau tra loi len phia Server
     + int answerinput(char s[]); // Tra loi cac cau hoi,moi cau gioi han trong 15s
   * Client gửi thông tin đăng nhập và đăng ký lên phía Server , thông tin này được nhập vào từ bàn phím bằng cách gọi các hàm login() hay signup()
   * Client sau khi đăng nhập thành công bắt đầu nhận thông tin đề thi từ Server, ghi nó vào cấu trúc

struct question{

char quest[1024];

};

Và dùng cấu trúc này để hiển thị câu hỏi

* + Dùng hàm answerinput() để nhập vào câu trả lời, trong hàm này dùng hàm **alamr(15)** để giới hạn thời gian cho mỗi câu hỏi chỉ được trả lời trong 15s
  + Gửi câu trả lời tới Server
  + Sau khi Test xong, có thể tải về đề và đáp án bằng cách gửi Request tới Server

CHƯƠNG 3. CÀI ĐẶT VÀ KIỂM THỬ

## Cài đặt chương trình

Chương trình được xây dựng trên nền Linux. Và sử dụng thư viện Socket của Linux để lập trình.

## Dịch và chạy chương trình

Tại Server dịch theo các bước sau :

* $gcc –c server.c // biên dịch ra file server.o
* $gcc –c fields.c // biên dịch ra file fields.o
* $ gcc –o server.out server.o fields.o // Biên dịch ra file chạy server.out (liên kết thư viện)

Tại Client dịch theo các bước sau :

* $gcc –c client client.c //Biên dịch ra file chạy client

Chạy chương trình tại Server :

* $./server.out

Chạy chương trình tại Client :

* $./client

1. **Kiểm thử:**

Các kịch bản kiểm thử :

* Test kết nối Client – Server
* Test chức năng đăng nhập : Thành công và không thành công
* Test chức năng đăng ký : Thành công và không thành công
* Test gửi câu hỏi,gửi câu trả lời
* Test chấm bài thi
* Test nhận file

# CHƯƠNG 4. KẾT LUẬN

* 1. **Về chức năng so với yêu cầu bài toán :**

Chương trình đã thực hiện đầy đủ các yêu cầu mà bài toán đặt ra. Cụ thể:

* Đăng nhập .
* Đăng kí người dùng.
* Gửi câu hỏi , nhận câu trả lời, tính toán thời gian cho mỗi câu hỏi.
* Tính toán đưa ra kết quả.
* Khi kết thúc, có thể nhận đề và đáp án để check.

II. **Về hiệu năng chương trình :**

Ưu điểm :

* Tính toán, gửi câu hỏi chính xác
* Server có thể phục vụ được nhiều Client truy cập tới cùng lúc(đã test thành công với 5 client)

Thiếu sót :

* Chương trình còn thiếu giao diện đồ họa để có thể tối đa sức mạnh tương tác người dùng.

# TÀI LIỆU THAM KHẢO

[1] Sile bài giảng Thực hành lập trình mạng do thầy giáo Bùi Trọng Tùng giảng dạy.

[2] TCP/IP Sockets in C : Practical Guide for Programmer – Michael J. Donahoo, Kenneth L. Cavert

[3] Unix Network Programming Volume 1, Third Edition: The Sockets Networking API – W. Richard Stevens, Bill Fenner, Andrew M. Rudoff

[4] Các trang web về lập trình C, lập trình socket

[+] http://www.cs.cf.ac.uk/Dave/C/

[+] http://librenix.com/?inode=3044