

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
Viện Công nghệ thông tin và Truyền thông

BÁO CÁO BÀI TẬP LỚN
Bộ môn lập trình mạng

Đề tài

Xây dựng ứng dụng thi trắc nghiệm



Giảng viên hướng dẫn: **ThS. Bùi Trọng Tùng**
Nhóm

Bùi Ngọc Nam 20142998

Bùi Quang Trí 20142998

Hà Nội, ngày 15 tháng 1 năm 2019

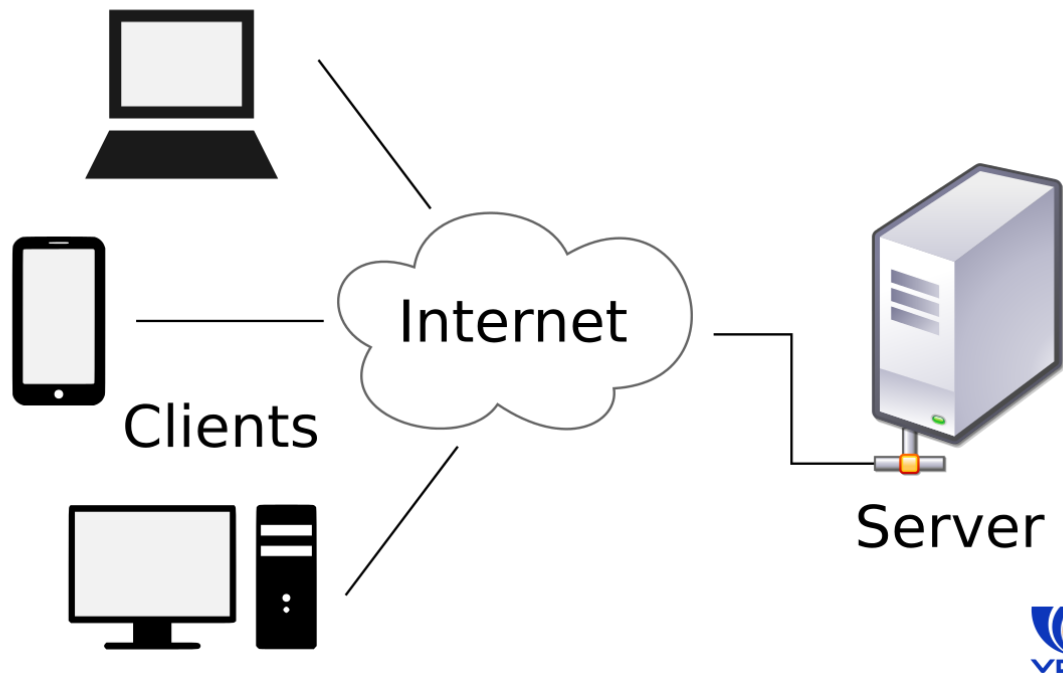
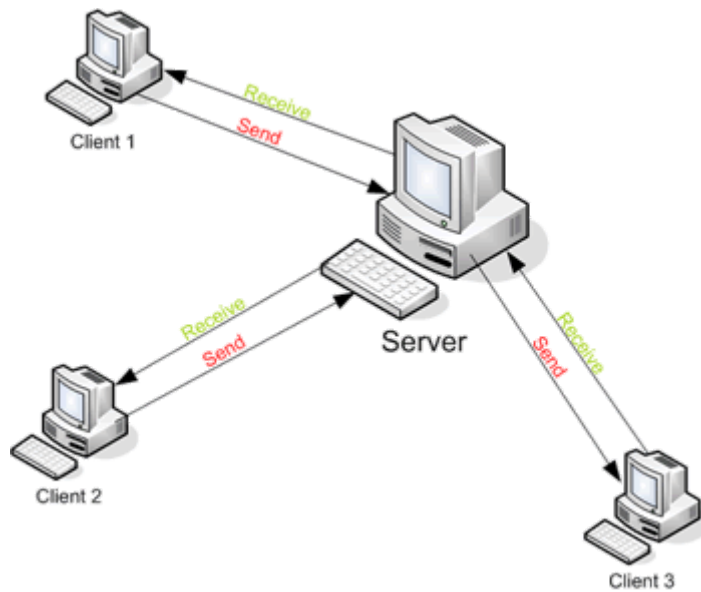
Mục lục

| | |
|--|-----------|
| Tổng quan về lập trình socket | 2 |
| Mô tả bài toán | 4 |
| Giới thiệu đề tài | 4 |
| Yêu cầu chức năng | 4 |
| Giải pháp thực hiện | 5 |
| Luồng thực hiện | 5 |
| Sơ đồ trạng thái | 6 |
| Cấu trúc dữ liệu sử dụng | 9 |
| Kết quả và demo | 12 |
| Cấu trúc mã nguồn: | 12 |
| Test | 13 |
| Một số hình ảnh khi thực hiện chương trình | 13 |
| Hướng dẫn sử dụng | 17 |
| Phân chia công việc trong nhóm | 17 |

I. Tổng quan về lập trình socket

- Mô hình client/server:

- Mô hình được phổ biến nhất và được chấp nhận rộng rãi trong các hệ thống phân tán là mô hình client/server. Trong mô hình này sẽ có một tập các tiến trình mà mỗi tiến trình đóng vai trò như là một trình quản lý tài nguyên cho một tập hợp các tài nguyên cho trước và một tập hợp các tiến trình client trong đó mỗi tiến trình thực hiện một tác vụ nào đó cần truy xuất tới tài nguyên phần cứng hoặc phần mềm dùng chung. Bản thân các trình quản lý tài nguyên cần phải truy xuất tới các tài nguyên dùng chung được quản lý bởi một tiến trình khác, vì vậy một số tiến trình vừa là tiến trình client vừa là tiến trình server. Các tiến trình phát ra các yêu cầu tới các server bất kỳ khi nào chúng cần truy xuất tới một trong các tài nguyên của các server. Nếu yêu cầu là đúng đắn thì server sẽ thực hiện hành động được yêu cầu và gửi một đáp ứng trả lời tới tiến trình client.
- Mô hình client/server cung cấp một cách tiếp cận tổng quát để chia sẻ tài nguyên trong các hệ thống phân tán. Mô hình này có thể được cài đặt bằng rất nhiều môi trường phần cứng và phần mềm khác nhau. Các máy tính được sử dụng để chạy các tiến trình client/server có nhiều kiểu khác nhau và không cần thiết phải phân biệt giữa chúng; cả tiến trình client và tiến trình server đều có thể chạy trên cùng một máy tính. Một tiến trình server có thể sử dụng dịch vụ của một server khác.
- Mô hình truyền tin client/server hướng tới việc cung cấp dịch vụ. Quá trình trao đổi dữ liệu bao gồm:
 - Truyền một yêu cầu từ tiến trình client tới tiến trình server
 - Yêu cầu được server xử lý
 - Truyền đáp ứng cho client
- Mô hình truyền tin này liên quan đến việc truyền hai thông điệp và một dạng đồng bộ hóa cụ thể giữa client và server. Tiến trình server phải nhận thức được thông điệp được yêu cầu ở bước một ngay khi nó đến và hành động phát ra yêu cầu trong client phải được tạm dừng (bị phong tỏa) và buộc tiến trình client ở trạng thái chờ cho tới khi nó nhận được đáp ứng do server gửi về ở bước ba.
- Mô hình client/server thường được cài đặt dựa trên các thao tác cơ bản là gửi (send) và nhận (receive)
- Một số hình ảnh về mô hình client-server



- Kết nối Server Client với TCP/IP
 Khi được chạy, server cần được xác định rõ địa chỉ IP và sẽ “lắng nghe” trên một port cụ thể. Server sẽ nằm trong trạng thái này cho đến khi client gửi đến một yêu cầu kết nối. Sau khi được server chấp nhận, một connection sẽ hình thành cho phép server và client giao tiếp với nhau
- Bài toán thực hiện xây dựng một ứng dụng chia sẻ file. Như vậy khi người dùng gửi yêu cầu lên, server cần xác định rõ đó là người dùng nào, và sẽ ở trạng thái chờ đợi cho đến khi client gửi yêu cầu lên. Do đó việc giải quyết vấn đề bằng cách sử dụng TCP/IP là phù hợp

II. Mô tả bài toán

1. Giới thiệu đề tài

Bài toán thực hiện xây dựng ứng dụng thi trắc nghiệm trực tuyến. Với mỗi client kết nối đến server, người dùng có thể thực hành kiểm tra thử, tạo phòng thi với số câu hỏi và thời gian thi mong muốn hoặc tham gia vào phòng thi khác. Các client chỉ có thể tham gia những phòng mà chủ phòng chưa bắt đầu thi. Sau đó, chấm điểm và gửi kết quả về cho client

Ứng dụng sử dụng mô hình TCP/IP socket để xây dựng.

2. Yêu cầu chức năng

- Server có các chức năng sau:
 - Gửi câu hỏi đến cho người dùng
 - Chấm điểm bài thi và gửi kết quả cho người dùng.
- Client có các chức năng sau:
 - Yêu cầu làm bài thi mẫu, số câu hỏi do server quyết định
 - Yêu cầu tạo phòng thi mới, tự quyết định số câu hỏi và thời gian thi
 - Tham gia vào một phòng có sẵn.

III. Giải pháp thực hiện

1. Luồng thực hiện

- Ứng dụng sử dụng TCP trên giao diện console để giao tiếp với người dùng
- Luồng thực hiện:
 - Tại màn hình console, một menu sẽ hiện ra cho người dùng thực hiện lựa chọn. Có 3 chế độ: luyện tập, tạo phòng mới, và tham gia phòng hiện có. Sau khi người dùng lựa chọn, server sẽ gửi câu hỏi đến cho người dùng thực hiện bài kiểm tra.
 - Client sẽ gồm một socket giao tiếp với người dùng để thực hiện những chức năng.
 - Server sẽ thực hiện theo đa tiến trình để đảm bảo phục vụ được nhiều client tại 1 thời điểm. Trên server ta sẽ lưu lại danh sách câu hỏi, danh sách các phòng lưu trên một mảng cấu trúc.

- Luồng thực hiện có thể ví dụ như sau: Ban đầu client A yêu cầu chức năng luyện tập, server sẽ nhận yêu cầu, chọn ra số câu hỏi và gửi lại cho client. Client A sẽ thực hiện bài kiểm tra, trả lời câu hỏi và gửi đáp án cho server. Server sẽ đối chiếu đáp án của client với đáp án đúng, và gửi số câu hỏi đúng về cho client.

2. Cấu trúc dữ liệu sử dụng

- Danh sách câu hỏi được lưu trong file test.db, các trường trong database gồm:
 - QuestionID <int> : chứa giá trị ID của câu hỏi
 - Question <text> : Nội dung câu hỏi
 - Answer_A<text> : Câu trả lời 1
 - Answer_B <text> : Câu trả lời 2
 - Answer_C <text> : Câu trả lời 3
 - Answer_D <text> : Câu trả lời 4
 - Valid_answer <text> : Đáp án đúng cho câu hỏi
- Khi ứng dụng được khởi động, danh sách câu hỏi được đọc ra và lưu dưới dạng mảng mà mỗi mắt xích trong danh sách là một cấu trúc **exam** có dạng:

```

BUFF_SIZE = 1024;
typedef struct
{
    int quesID;
    char question[BUFF_SIZE];
    char answerA[BUFF_SIZE];
    char answerB[BUFF_SIZE];
    char answerC[BUFF_SIZE];
    char answerD[BUFF_SIZE];
} exam;

```

- Client và server giao tiếp với nhau thông qua các hàm gửi và nhận tin nhắn, trong đó mỗi tin nhắn được xây dựng theo định dạng:

ACTION + <message>

Chi tiết các danh sách lệnh:

Message từ client gửi đến server:

| | | | |
|-------------|---------------|--------|---------------------------------------|
| PRACTICE | | | Bắt đầu bài thi luyện tập |
| NEW | <num_of_ques> | <time> | Bắt đầu bài thi thật, tạo phòng mới. |
| CHOOSE | <room_id> | | Bắt đầu bài thi thật, chọn phòng thi |
| FINISH | <answer> | | Hoàn thành, gửi đáp án. |
| REQUESTLIST | | | Yêu cầu lấy danh sách phòng thi trống |
| QUIT | | | Thoát |

Message từ server gửi đến client:

| | | |
|--------------|-----------|--------------------------------|
| SENDROOMLIST | | Gửi danh sách phòng đến client |
| SEND | <package> | Gửi gói câu hỏi cho client |
| SENDRESULT | <result> | Gửi kết quả cho người thi |
| NOROOM | | Không có phòng phù hợp |
| END | | Thời gian làm bài kết thúc. |
| CREATEROOM | | Tạo phòng thành công |

IV. Kết quả và demo

1. Cấu trúc mã nguồn:

Project:

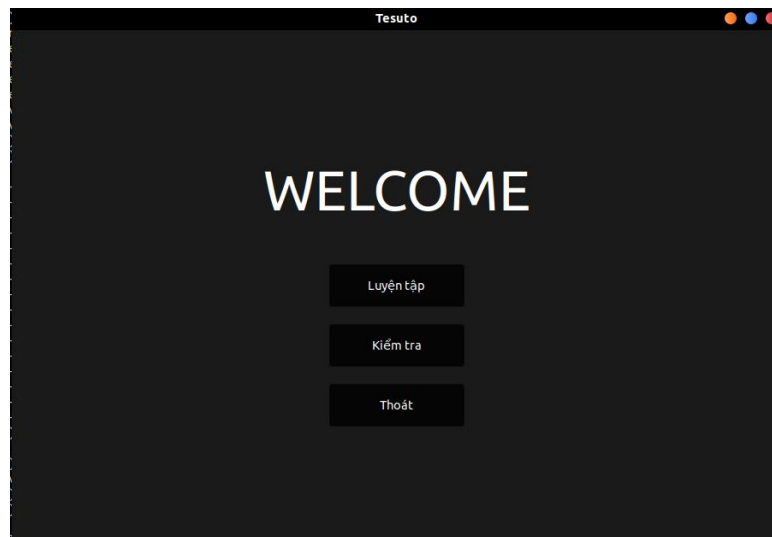
```
|_ server.c  
|_ client.c  
|_ test.db  
|_ makefile  
|_ readme.md
```

- Mã nguồn chương trình được chia theo 2 file chính **server.c** và **client.c** ứng với từng nhóm chức năng riêng biệt.
- 2 file chính:
 - **client.c** dùng để chạy ứng dụng phía client
 - **server.c** dùng để chạy ứng dụng phía server
- Các file được sử dụng trong file server và file client được chỉ ra trong file **makefile**
- Biên dịch chương trình: từ cửa sổ terminal chạy lệnh
`make`

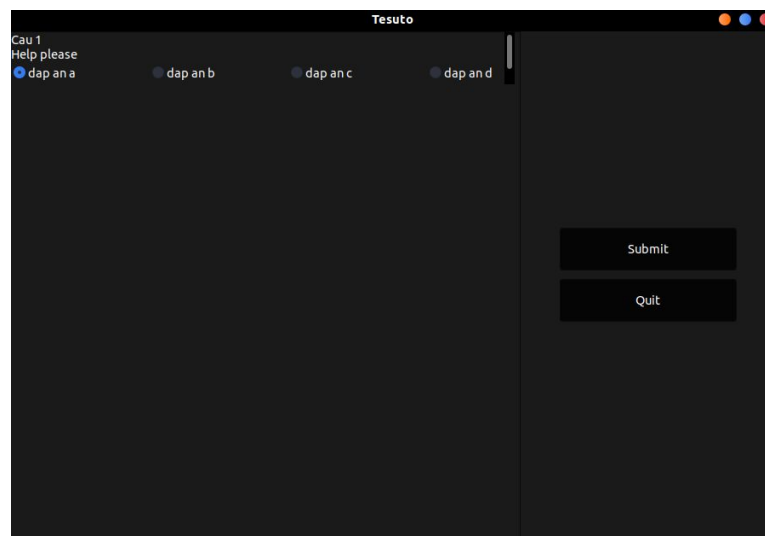
2. Kiểm thử

Để dễ dàng thực hiện kiểm tra, đánh giá chức năng cũng như độ chính xác, tin cậy của sản phẩm, nhóm đã chỉnh sửa mã nguồn: chỉ hiển thị những phòng trong trạng thái chờ đợi. Những phòng trong trạng thái đang thi hoặc đã kết thúc thi thì sẽ không hiển thị với người dùng.

3. Một số hình ảnh khi thực hiện chương trình:



Hình 1: Giao diện chính khi khởi động chương trình bản GUI



Hình 2: Giao diện khi user làm bài thi bản GUI

```
1. Practice
2. Create a new room
3. Choose a room
4. Quit
Choose (1-4):
```

Hình 3: Giao diện menu chương trình bản terminal

```
1. Practice
2. Create a new room
3. Choose a room
4. Quit
Choose (1-4): 1
Question 1: 1+1.
    Answer A: 1
    Answer B: 3
    Answer C: 2
    Answer D: 4
Your answer: C
Question 2: 2+2.
    Answer A: 1
    Answer B: 2
    Answer C: 4
    Answer D: 3
Your answer: C
Question 3: 2+3.
    Answer A: 1
    Answer B: 3
    Answer C: 5
    Answer D: 6
Your answer: C
YOUR Score: 3
```

Hình 4: Người dùng nhận kết quả sau khi hoàn thành chế độ luyện tập

```
1. Practice
2. Create a new room
3. Choose a room
4. Quit
Choose (1-4): 2
Please insert number of question (1-30): 10
Please insert time (minute): 100
Minimum is 1, maximum is 60
Please insert time (minute): 30
--- Room created successfully!
Shall we begin (type "yes" to start):
```

Hình 5: Người dùng tạo phòng thi mới và thiết lập số câu hỏi cùng thời gian

```

1. Practice
2. Create a new room
3. Choose a room
4. Quit
Choose (1-4): 3
Room id: 2.
        Number of question: 1
        Time: 1 minute
Choose a room:

```

Hình 6: Chương trình hiện danh sách các phòng chứa thi cho người dùng khác

V. Hướng dẫn sử dụng

- Tải mã nguồn
 - link github: https://github.com/mandela02/final_tesuto
- Di chuyển vào thư mục Project/Source trong thư mục vừa tải về được qua cửa sổ terminal
- Khởi động server theo cú pháp: **./server port**
trong đó port là số hiệu cổng kết nối
- Khởi động client tại cửa sổ terminal khác với cú pháp: **./client IP port**
trong đó IP là địa chỉ ip của server muốn kết nối đến (ở đây sẽ là localhost có địa chỉ 127.0.0.1), port là số hiệu cổng kết nối (giống với số hiệu đã chạy ở server)
- Để xoá các file biên dịch đi, người dùng thực hiện dòng lệnh **make clean** trong cửa sổ terminal

VI. Phân chia công việc trong nhóm

| Công việc | Người thực hiện |
|---|-----------------|
| Thiết kế cơ sở dữ liệu | Nam |
| Thiết kế COMMAND LIST giữa server và client | Nam, Trí |
| Thống nhất cấu trúc tệp, struct sử dụng | Nam, Trí |
| Chức năng luyện tập | Nam |
| Chức năng tạo phòng mới | Trí |

| | |
|--|-----|
| Chức năng tham gia phòng sẵn có | Trí |
| Chức năng chấm điểm | Trí |
| Thiết kế giao diện cửa sổ cho người dùng | Nam |