

**BÁO CÁO BÀI TẬP LỚN**

**MÔN HỌC: THỰC HÀNH LẬP TRÌNH MẠNG**

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

Viện Công Nghệ Thông Tin và Truyền Thông

**Giảng viên : ThS. Bùi Trọng Tùng**

**Lớp : Việt Nhật B - K59**

**Nhóm thực hiện : Bùi Quang Đại - 20140919**

**Nguyễn Tư Quyết – 2014xxxx**

**MỤC LỤC**

[PHẦN 1. MỞ ĐẦU 2](#_Toc470282002)

[1. Giới thiệu đề tài 2](#_Toc470282003)

[1.1 Tên đề tài 2](#_Toc470282004)

[1.2 Luật chơi 2](#_Toc470282005)

[2. Phân công công việc 2](#_Toc470282006)

[PHẦN 2. CƠ SỞ LÝ THUYẾT LIÊN QUAN 4](#_Toc470282007)

[1. Các kĩ thuật đề tài yêu cầu 4](#_Toc470282008)

[2. Cách thức giải quyết 4](#_Toc470282009)

[PHẦN 3. XÁC ĐỊNH CHƯƠNG TRÌNH 7](#_Toc470282010)

[1. Thiết kế giao thức 7](#_Toc470282011)

[2. Cấu trúc chương trình 7](#_Toc470282012)

[2.1 Phía client 7](#_Toc470282013)

[2.2 Phía server 7](#_Toc470282014)

[3. Các hàm và chức năng 8](#_Toc470282015)

[3.1 Phía client 8](#_Toc470282016)

[3.2 Phía server 10](#_Toc470282017)

[PHẦN 4. KIỂM THỬ CHỨC NĂNG 11](#_Toc470282018)

[1. Kết nối server-client 11](#_Toc470282019)

[2. Chức năng sign up 11](#_Toc470282020)

[3. Chức năng Login 12](#_Toc470282021)

[4. Chức năng Play 13](#_Toc470282022)

[5. Chức năng Quit 16](#_Toc470282023)

[Bắt sự kiện Ctrl+C 17](#_Toc470282024)

[PHẦN 5. KẾT LUẬN 18](#_Toc470282025)

[1. Kiến thức thu được sau môn học 18](#_Toc470282026)

[2. Lời cảm ơn 18](#_Toc470282027)

PHẦN 1. MỞ ĐẦU

1. Giới thiệu đề tài
   1. Tên đề tài

Xây dựng chương trình mô phỏng game show: “Ai là triệu phú”.

* 1. Luật chơi

Trong mỗi lượt chơi, các ứng viên trong phòng sẽ tham gia một phần trả lời nhanh trước khi chọn ra người chơi chính cho chương trình. Họ phải trả lời một câu hỏi bằng cách sắp xếp các phương án A, B, C, D theo thứ tự đáp án đúng của chương trình. Người trả lời đúng và nhanh nhất sẽ được chọn làm người chơi chính ngồi trên chiếc "ghế nóng" ở giữa sân khấu cùng với người dẫn chương trình.

Sau khi được chọn làm người chơi chính, người chơi phải trả lời 15 câu hỏi với cấp độ từ dễ đến khó, không giới hạn thời gian suy nghĩ trả lời. Mỗi câu hỏi đều được gắn với một mức tiền thưởng quy định. Người chơi cần phải vượt qua 3 mốc quan trọng là câu số 5 (mốc quan trọng 1), câu số 10 (mốc quan trọng 2) và câu số 15 (mốc "TRIỆU PHÚ") mà khi vượt qua các mốc này, họ chắc chắn có được số tiền thưởng tương ứng của các câu hỏi đó.

Người chơi có quyền rời khỏi cuộc chơi hoặc đấu tiếp. Nếu quyết định rời khỏi cuộc chơi, người chơi sẽ ra về với số tiền tương ứng với số câu hỏi họ mới vượt qua trước đó. Nếu quyết định đấu tiếp mà trả lời sai, số tiền thưởng sẽ bị xuống mốc quan trọng trước đó và người chơi sẽ phải ra về với số tiền này.[[3]](https://vi.wikipedia.org/wiki/Ai_l%C3%A0_tri%E1%BB%87u_ph%C3%BA#cite_note-3) Nếu trả lời sai khi chưa vượt qua mốc quan trọng đầu tiên, người chơi sẽ ra về mà không nhận được tiền thưởng của chương trình. Nếu trả lời đúng hết tất cả 15 câu hỏi của chương trình, người chơi sẽ trở thành "Triệu Phú", nhận được tấm séc với trị giá giải thưởng tương ứng với câu thứ 15 của chương trình.

Người chơi có 2 sự trợ giúp sau có thể được sử dụng bất cứ lúc nào nếu họ không biết câu trả lời hoặc chưa chắc chắn với suy nghĩ của mình. Trong một câu hỏi, người chơi có quyền dùng tất cả sự trợ giúp, nhưng tất cả quyền trợ giúp chỉ được sử dụng một lần:

\*50:50: Máy tính bỏ đi 2 phương án sai để người chơi chọn một trong 2 phương án còn lại.

\*Hỏi ý kiến khán giả trong phòng: Mỗi khán giả trong phòng sẽ nhận được yêu cầu trợ giúp để đưa ra phương án mình chọn. Khi chọn sự trợ giúp này, người chơi sẽ nhận được kết quả trợ giúp dưới dạng phần trăm số khán giả lựa chọn từng phương án.

1. Phân công công việc

* Thiết kế giao thức: Bùi Quang Đại, Nguyễn Tư Quyết
* Viết báo cáo: Bùi Quang Đại, Nguyễn Tư Quyết
* Phân chia công việc cụ thể :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Task | Message | Subtask | Assignee |
| 1 | Đăng nhập, đăng kí, đăng xuất | USER, PASS | Đăng nhập | Quyết |
| SIGU, SIGP, SIGC | Đăng kí |
| LOUT | Đăng xuất |
|  |  |
| 3 | Lấy câu hỏi, truyền câu hỏi cho client |  | Lấy câu hỏi từ file | Quyết |
|  |  |  |
|  |  |  |
|  |  |  |
| 2 | Tạo phòng, vào phòng | CRRM number | Tạo phòng | Đại |
| JOIN number | Vào phòng |
| STAR | gửi câu hỏi chung cho các người chơi trong phòng |
|  |  |
| 4 | Chơi | ANSW mã câu trl | Trả lời câu hỏi | Đại |
|  |  |
|  |  |
|  |  |
| 6 | Trợ giúp (HELP) | HELP 1 | 50 50 | Quyết |
| HELP 2 | Tư vấn |
|  |  |
|  |  |
| 7 | Dừng cuộc chơi | STOP | Dừng cuộc chơi | Quyết |
|  |  |
|  |  |
|  |  |
| 8 | Thoats khỏi phòng | LEAV | Thoát khỏi phòng chơi hiện tại | Quyết |
|  |  |
|  |  |
|  |  |

**Sơ đồ thiết kế giao thức cụ thể:**

PHẦN 2. CƠ SỞ LÝ THUYẾT LIÊN QUAN

1. Các kĩ thuật đề tài yêu cầu

Lập trình đa luồng bằng ngôn ngữ C.

1. Cách thức giải quyết
   1. Lập trình đa luồng bằng ngôn ngữ C(chỗ này chưa sửa)

Lập trình đa luồng sẽ được tiến hành bên phía server. Nhờ đó các client sẽ làm việc với server mà không phải chờ nhau. Cụ thể như sau:

Khi có một client kết nối tới, server sẽ tạo một tiến trình con để giao tiếp với client trên một kết nối socket giữa server và client đó.

Sử dụng lệnh gọi hệ thống fork() để sinh tiến trình con. Tiến trình con được sinh ra này sẽ sao chép toàn bộ lệnh cũng như dữ liệu của tiến trình cha, nên cần có một số lưu ý sau:

* Trong tiến trình con, đóng socket mà nhận kết nối từ client của tiến trình cha, đồng thời tiến trình cha đóng socket giữa server và client để cho tiến trình con sử dụng socket đó.
* Khi tiến trình con kết thúc, lệnh cũng như dữ liệu nó đã sao chép vẫn lưu trên bộ nhớ, vì thế cần có một lệnh gọi hệ thống bắt sự kiện tiến trình con kết thúc để dọn rác do nó để lại, tránh tình trạng lãng phí bộ nhớ.
* Hàm fork() sẽ trả về một id để nhà phát triển biết được đâu là tiến trình con, đâu là tiến trình cha, từ đó có các xử lý phù hợp. Cụ thể, fork() trả về tiến trình con giá trị 0, trả về tiến trình cha giá trị dương.

PHẦN 3. XÁC ĐỊNH CHƯƠNG TRÌNH

1. Thiết kế giao thức
2. Cấu trúc chương trình
   1. Phía client

Mã nguồn xử lý được viết trong file client.c

Các model và các hàm hỗ trợ được lưu trong thư mục lib

* + Login\_signup.h: Các hàm hỗ trợ đăng kí, đăng nhập
  + play.h: Các hàm hỗ trợ người dùng chơi game
  + protocol.h: Cấu trúc protocol dùng để giao tiếp giữa client và server
  + user.h : Cấu trúc user, gồm user\_id, password
  + question.h: Cấu trúc Question, gồm nội dung câu hỏi, 4 câu trả lời...
  + answer.h: Cấu trúc Answer, gồm 4 đặt cược cho 4 câu trả lời
  + title.c: Cấu trúc Title, gồm 2 chủ đề cùng với 1 biến lựa chọn
  + fire\_work.h: Các hàm hiển thị

* 1. Phía server

Mã nguồn chương trình server được viết trong file fork\_server.c

Các model được lưu trong thư mục lib

* protocol.h, protocol.c : Cấu trúc protocol dùng để giao tiếp giữa client và server, cùng với các hàm hỗ trợ protocol.
* User.h, user.c: Cấu trúc user, gồm user\_id, password và các hàm hỗ trợ cho cấu trúc user.
* Question.h, question.c: Cấu trúc Question, gồm nội dung câu hỏi, 4 câu trả lời, … và các hàm hỗ trợ cấu trúc question
* Answer.h, answer.c: Cấu trúc Answer, gồm 4 đặt cược cho 4 câu trả lời và các hàm hỗ trợ cấu trúc answer
* Title.h, title.c: Cấu trúc Title, gồm 2 chủ đề cùng với 1 biến lựa chọn và các hàm hỗ trợ cấu trúc Title

Các hàm hỗ trợ được lưu trong thư mục helper

1. 3. Các hàm và chức năng
   1. Phía client

Login\_signup.h:

* + Hàm int get\_pass(char pass[]): ẩn password trong terminal
  + Hàm int requestSignup(protocol \*p): gửi yêu cầu đăng kí lên server, trả về 0 khi yêu cầu thất bại, ngược lại trả về 1.
  + Hàm int signUp(protocol \*p): tạo giao diện đăng kí, khởi tạo các gía trị cho protocol để gửi lên server, hàm trả về -1 nếu yêu cầu đăng nhập thất bại, trả về 1 nếu nhập đúng xác nhận mật khẩu, nếu không trả về 0.
  + Hàm int requestSignin(protocol \*p): gửi yêu cầu đăng nhập lên server, trả về 0 khi yêu cầu thất bại, ngược lại trả về 1.
  + Hàm int signIn(protocol \*p): tạo giao diện đăng nhập, khởi tạo các gía trị cho protocol để gửi lên server, hàm trả về -1 nếu yêu cầu đăng nhập thất bại, trả về 1 nếu thành công.

play.h:

* + Hàm void printMenuPlay(): Hiển thị menu play
  + Hàm void delay(): delay
  + Hàm int chooseTitle(protocol \*p): hiện thị chủ đề cho người chơi lựa chọn, hàm trả về chủ đề mà người chơi lựa chọn .
  + Hàm void printQuestion(protocol \*p,int x): hiện thị nội dung câu hỏi, đọc câu trả lời từ bàn phím, kiểm tra lỗi, gán các giá trị trả lời hợp lệ cho biến protocol, và thông điệp sent anser cho biến để gửi lên server.
  + Hàm int play(protocol \*p,int sockfd): hiển thị giao diện chơi game, gửi chủ đề lên server,nhận câu hỏi, gửi câu trả lời, hàm trả về 0 khi người dùng thoát game, ngược lại trả về 1.

client.c:

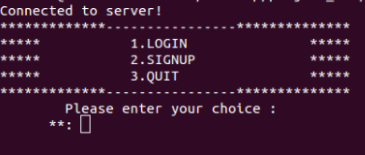
* + Hàm void INThandler(int sig): hàm bắt sự kiện Ctrl+C.
  + Hàm int createConnection(int PORT,char IP[10]): kết nối với server, trả về 1 nếu kết nối thành công, ngược lại trả về 0.
  + Hàm void printMenu(): hàm in menu làm việc chính

Fire\_work.h: Các hàm hiển thị.

* 1. Phía server

PHẦN 4. KIỂM THỬ CHỨC NĂNG

1. 1. Kết nối server-client

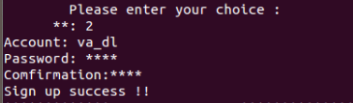
Client:

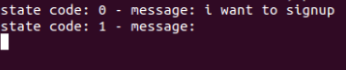


Server:

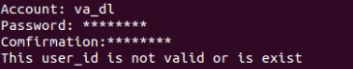
1. 2. Chức năng sign up

***Đăng kí 1 tài khoản mới***

Client: 

Server:

***Đăng kí 1 tài khoản đã tồn tại***

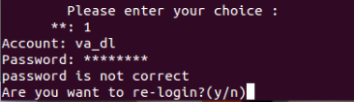
Client:

******

Server:

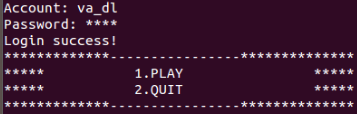
1. 3. Chức năng Login

***Login sai :***

Client: 

Server: 

***Login thành công (hiện menu play):***

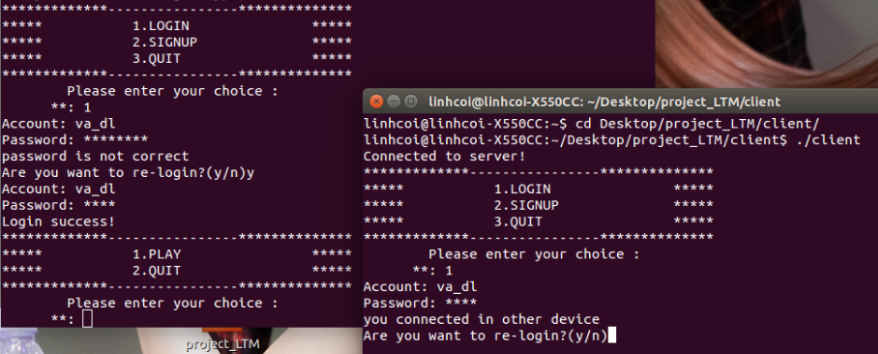
Client: 



Server:

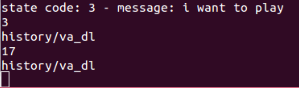
***Login cùng 1 tài khoản trên 2 client (thông báo lỗi, không cho login):***

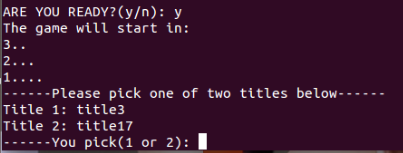
Client:



1. 4. Chức năng Play

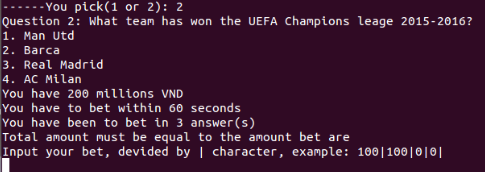
***Khi chọn play trong menu, server sẽ gửi về cho client 2 chủ đề để lựa chọn:***

Server: 

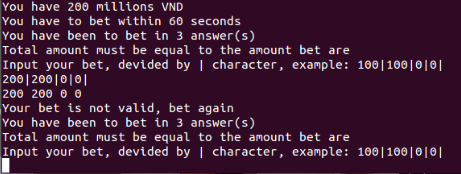
Client: 

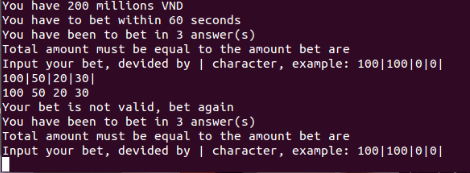
***Sau khi client chọn chủ đề, gửi cho server, server sẽ gửi lại câu hỏi và bắt đầu đếm thời gian***

Server: 

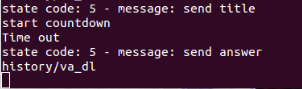
Client: 

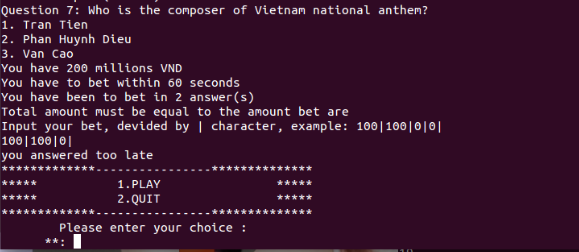
***Người chơi nhập số tiền đặt cược theo mẫu từ bàn phím, nếu nhập sai qui tắc,client sẽ bắt nhập lại***

Client:



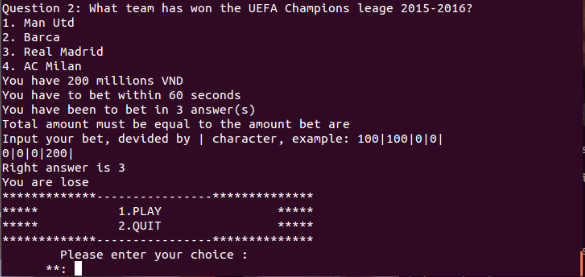
***Khi người chơi trả lời quá 40s, người chơi sẽ bị thua, đồng thời server sẽ setup người chơi lại trạng thái ban đầu (thoát khỏi trạng thái PLAYING)***

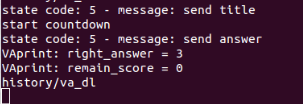
Server: 

Client: 

***Tương tự khi người chơi trả lời sai:***

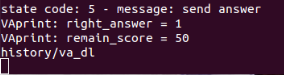
Client:



Server:

***Khi người chơi thắng, sẽ có thông báo thắng cuộc, số tiền nhận được, và quay lại menu play, đồng thời server sẽ setup người chơi lại trạng thái ban đầu (thoát khỏi trạng thái PLAYING)***

Server:

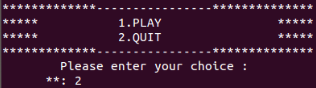


Client:



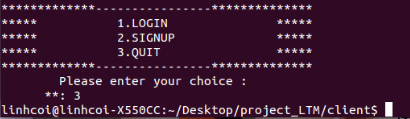
1. 5. Chức năng Quit

***Khi người chơi chọn quit trong menu Play, client sẽ gửi thông điệp yêu cầu sign out lên server***

Client: 

Server: 

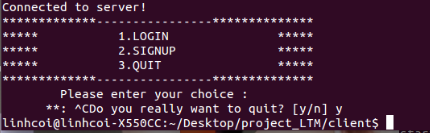
***Khi người chơi chọn quit trong menu chính, kết nối giữa client và server sẽ bị ngắt***

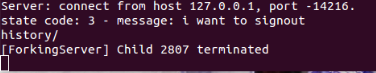
Client:

Server: 

1. Bắt sự kiện Ctrl+C

***Khi người chơi đột ngột nhấn Ctrl+C để thoát chương trình, client sẽ bắt sự kiện đó, và hỏi xác nhận, sau đó gửi cho server thông điệp I want to sign out***

Client: 

Server: 

PHẦN 5. KẾT LUẬN

1. Kiến thức thu được sau môn học

* Nắm được cách gửi dữ liệu qua socket giữa client và server.
* (Phần này m làm nhé)

1. Lời cảm ơn

Chúng em xin được chân thành cảm ơn sự giúp đỡ, giảng dạy nhiệt tình của thầy Bùi Trọng Tùng. Nhờ những kiến thức về môn lập trình mạng mà thầy đã nhiệt tình chỉ dạy trên những giờ học trên lớp, chúng em đã hoàn thành được chương trình này.

Mặc dù chúng em đã cố gắng thực hiện đầy đủ các yêu cầu đề ra, tuy nhiên những thiếu sót trong quá trình làm và tổng hợp là không thể tránh khỏi, rất mong nhận được những ý kiến đóng góp, chỉ bảo từ thầy để chúng em có thể nắm vững hơn các vấn đề đặt ra của môn học. Chúng em xin chân thành cảm ơn!

Hà Nội,ngày 4 tháng 1 năm 2018