

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

————— * —————



BÁO CÁO BÀI TẬP LỚN MÔN LẬP TRÌNH MẠNG

Đề tài 11: Xây dựng trò chơi ô ăn quan theo mô hình một
server chơi với nhiều client

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

Sinh viên thực hiện : **Vũ Duy Hùng** _ **AS1**

Đào Văn Mạnh _ **AS2**

Hà Nội, tháng 11 năm 2010

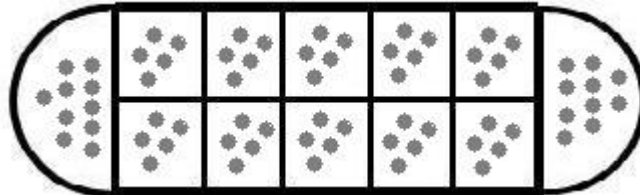
MỤC LỤC

MỤC LỤC	2
LỜI NÓI ĐẦU	3
I. Cơ sở lý thuyết	4
II. Xây dựng chương trình	6
III. Kiểm thử	10
IV. Kết luận	11
TÀI LIỆU THAM KHẢO	12
PHỤ LỤC	Error! Bookmark not defined.

LỜI NÓI ĐẦU

1. Giới thiệu trò chơi

“Ô ăn quan” là một trò chơi dân gian phổ biến, thường có 2 người chơi với nhau trên bàn chơi 10 ô như hình vẽ :



Người chơi sẽ chọn một ô con có quân bên phía mình và rải quân sang các ô bên cạnh theo một hướng nhất định, nếu rải hết quân mà ô tiếp theo có quân thì tiếp tục lấy và rải đi. Lượt đi dừng lại khi ô tiếp theo là ô trống, người chơi sẽ ăn được số quân nằm trong ô tiếp theo. Trò chơi dừng lại khi ai đó ăn hết số quân trong 2 ô quan to nhất.

2. Giới thiệu đề tài

Chương trình sẽ mô phỏng trò chơi theo mô hình một người chơi là máy khách – người chơi còn lại là máy chủ server. Máy chủ server phục vụ nhiều client một lúc, sau khi trò chơi kết thúc server sẽ gửi file log về cho client.

3. Phân công công việc :

- ✓ Hùng – làm API cho trò chơi, quản lí account.
- ✓ Mạnh – giao tiếp client – server.
- ✓ Cả 2 cùng ghép tích hợp và kiểm thử chương trình.

4. Tóm tắt nội dung báo cáo

Phần cơ sở lý thuyết : nêu lên loại hình server, loại giao thức được áp dụng để xây dựng giao tiếp client – server.

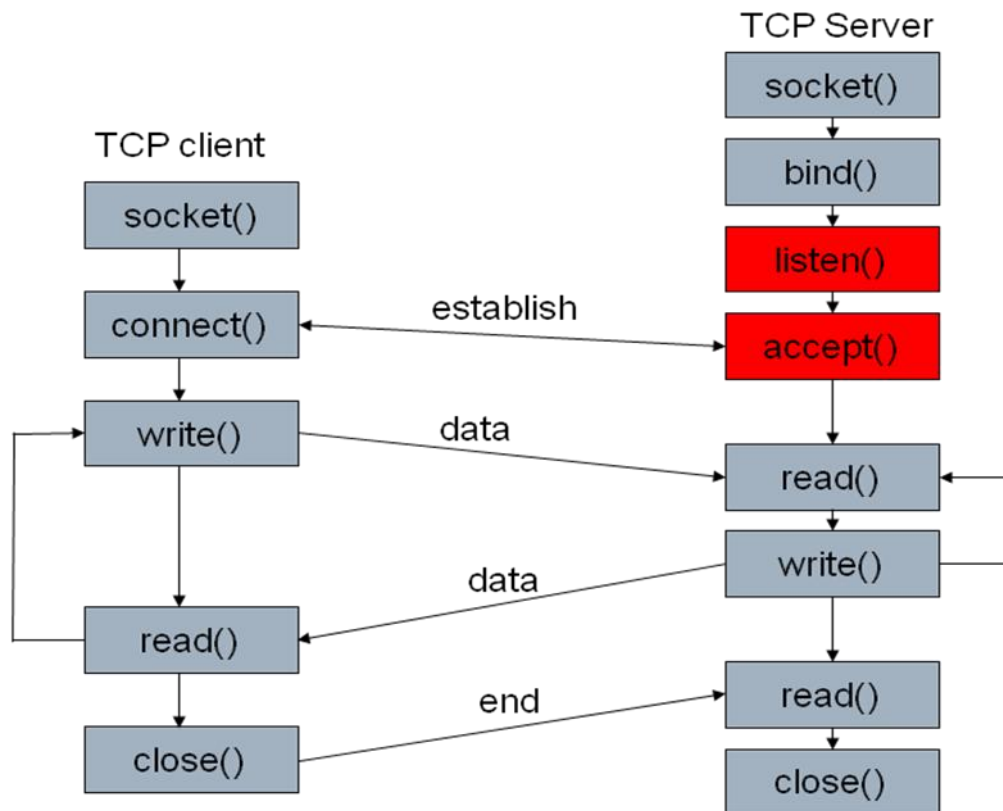
Phần xây dựng chương trình : nêu lên các quy ước giao tiếp giữa client – server trong chương trình.

Phần kiểm thử : đưa ra các trường hợp kiểm thử và kết quả kiểm thử.

Phần kết luận : nhận xét ưu nhược điểm của chương trình.

I. Cơ sở lý thuyết

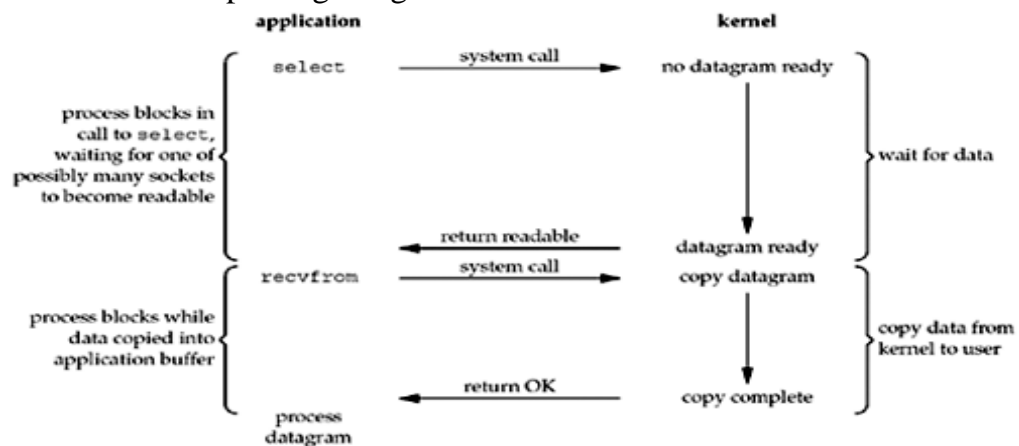
1. Mô hình TCP client-server



2. Concurrent server

Có 2 loại concurrent server là forking server và select server, do đặc tính chương trình là đáp ứng nhiều người chơi cùng 1 lúc, trao đổi giao tiếp giữa các người chơi với nhau nên dùng select server sẽ có nhiều ưu điểm hơn so với forking server.

Sơ đồ I/O Multiplexing dùng select



Cơ chế của select server là :

Dùng hàm *int select(int maxDescPlusl, fd_set *readDescs, fd_set *writeDescs, fd_set *exceptionDescs, stnlct timeval *timeout)*

để biết khi nào các socket client muốn giao tiếp với server.

Các tham số của select:

- maxDescPlusl : giá trị socket lớn nhất và được cộng thêm 1, để có thể thực hiện đọc được tất cả các socket
- readDescs: là 1 tập các socket sẵn sàng cho việc đọc dữ liệu
- writeDescs: tập các socket sẵn sàng cho việc ghi dữ liệu
- timeout: thời gian timeout nếu không có socket nào nhận, gửi dữ liệu

Dùng hàm *FD_ZERO()* để xoá tất cả các socket trong 1 tập các socket

Dùng hàm *FD_SET()* để gán 1 socket nào đó vào 1 tập các socket

Dùng hàm *FD_CLR()* để xoá 1 socket nào đó khỏi 1 tập các socket

Dùng hàm *FD_ISSET()* để kiểm tra 1 socket có tồn tại trong 1 tập các socket hay không

3. Chương trình sử dụng bộ thư viện BTree để quản lí tài khoản người chơi.

II. Xây dựng chương trình

1. Cấu trúc dữ liệu

Cấu trúc dữ liệu để lưu trữ thông tin tài khoản

```
typedef struct account
{
    char user[50];
    char pass[50];
}account;
```

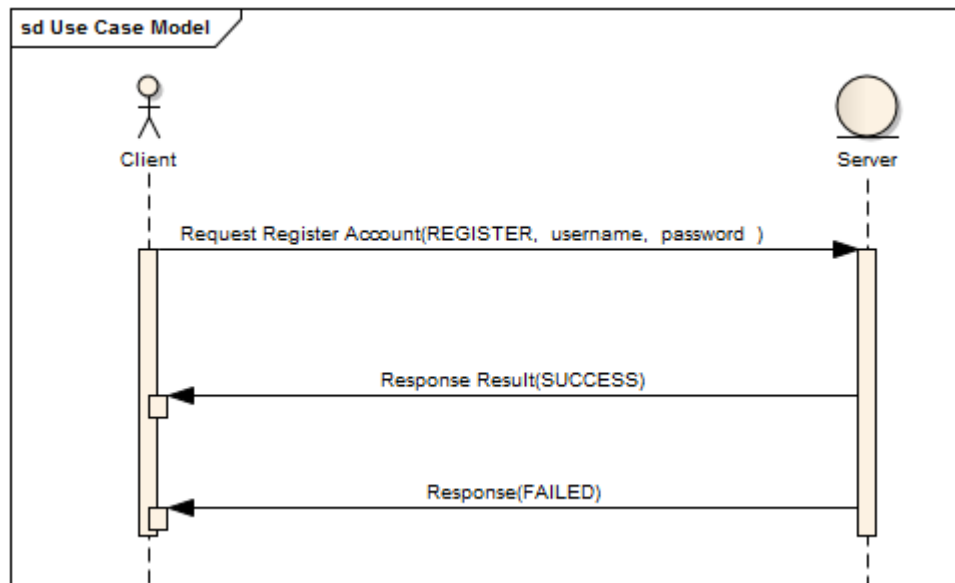
Cấu trúc dữ liệu để lưu trữ thông tin nước chơi

```
typedef struct step
{
    int position;    // position : 1->11
    char way;        // l = left or r = right
}step;
```

Cấu trúc dữ liệu truyền trên socket

```
typedef struct data
{
    int flag;        // cờ để nhận biết : REGISTER, SUCCESS,...
    step play;
    account object;
    char message[100];
}data;
```

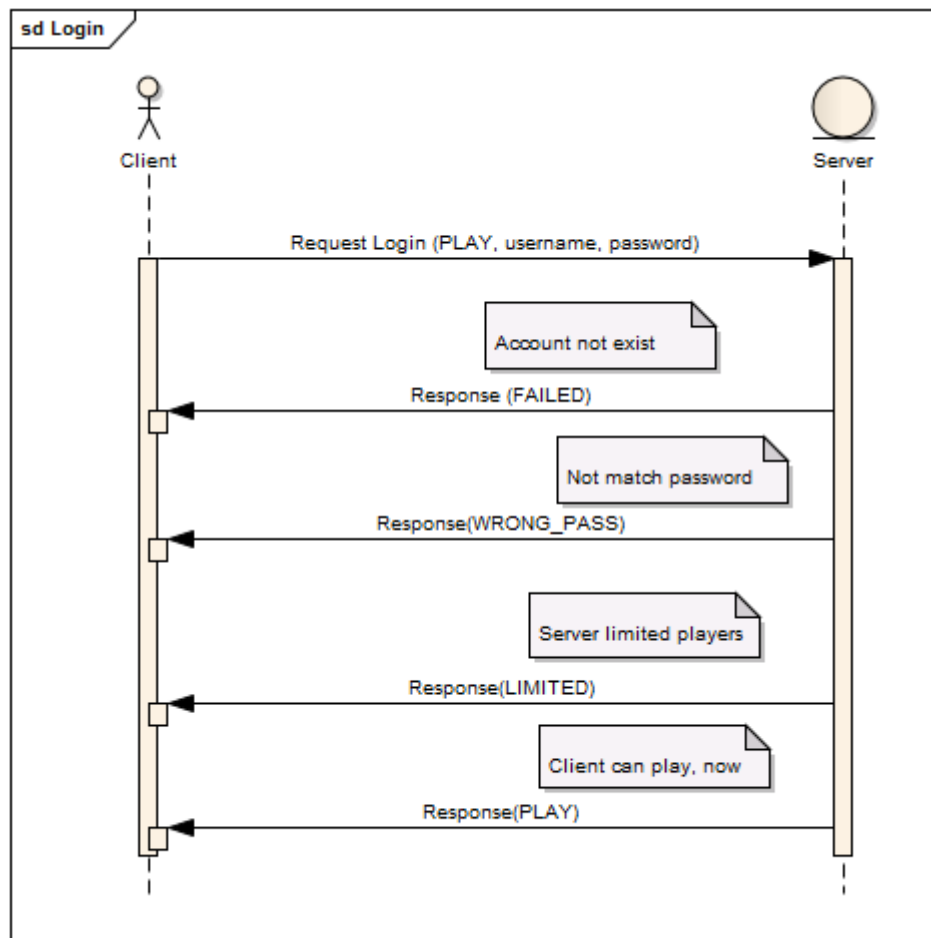
2. Quy ước giao tiếp khi đăng kí tài khoản



Căn cứ vào cờ REGISTER mà server nhận biết được client đang xin đăng kí tài khoản.

Client nhận dữ liệu trả về, căn cứ vào cờ SUCCESS, FAILED để biết đăng kí có thành công hay không.

3. Quy ước giao tiếp khi đăng nhập vào chơi



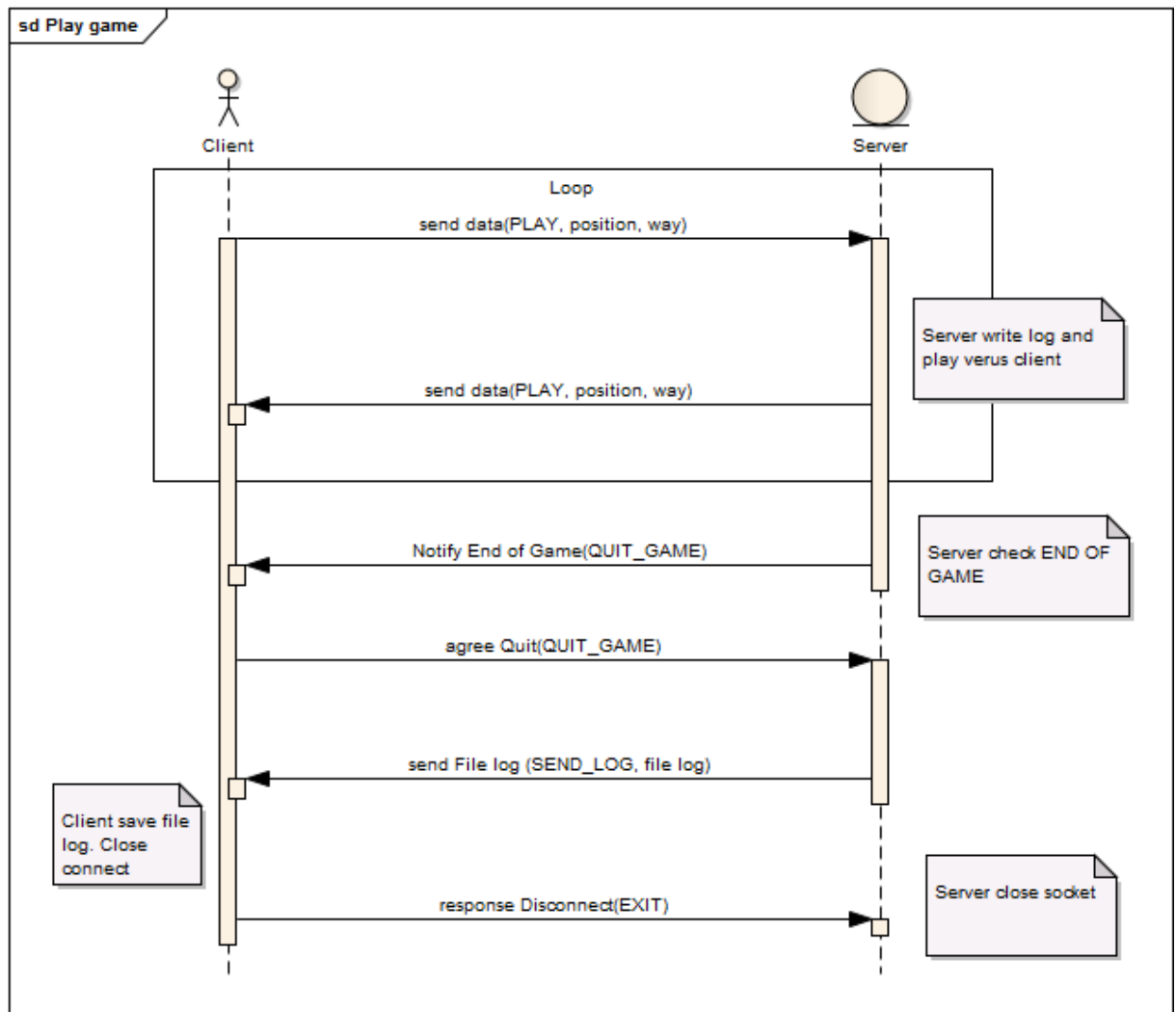
Khi server nhận được cờ PLAY, server sẽ kiểm tra xem client vẫn đang chơi hay là xin chơi lần đầu.

Nếu là lần đầu chơi :

- + Server kiểm tra tính xác thực của tài khoản, nếu tài khoản không tồn tại thì trả về cờ FAILED, nếu sai mật khẩu thì trả về cờ WRONG_PASS

- + Nếu thông tin tài khoản chính xác : server tạo 1 mảng lưu các giá trị trên bàn chơi cho client và trả về cờ PLAY cho client.

4. Quy ước giao tiếp khi chơi trò chơi



Giữa client và server chỉ cần gửi dữ liệu là nước chơi cho nhau. Phía server sẽ ghi nhật kí trò chơi và luôn kiểm tra xem trò chơi đã kết thúc hay chưa. Nếu trò chơi kết thúc thì gửi cờ QUIT_GAME về cho client. Lúc này client sẽ gửi lại cờ QUIT_GAME báo đồng ý thoát và sẵn sàng nhận file nhật kí về.

Khi client muốn thoát khỏi chương trình thì gửi cờ EXIT lên server, server sẽ đóng socket mà client này đang kết nối vào.

III. Kiểm thử

1. Chức năng : đăng kí tài khoản

- Tạo một tài khoản mới
- Tạo một tài khoản trùng với một tài khoản đã có

2. Chức năng : đăng nhập

- Đăng nhập với một tài khoản đã có : đúng username, password
- Đăng nhập với một tài khoản chưa đăng kí.
- Đăng nhập đúng username nhưng sai password.

3. Chức năng : chơi game

- Client nhập vào nước chơi đúng với quyền hạn của mình.
- Client nhập vào nước chơi không phải của mình.
- Client nhập vào nước chơi mà không có quân nào trong ô.
- Client chơi đến khi hết nước đi.
- Client xin thoát trò chơi giữa chừng.

4. Chức năng : ghi file nhật kí

- Client luôn có thể lưu lại file log dưới bất kì định danh nào.

5. Chức năng : đóng, mở các socket

- Mở nhiều client kết nối vào server để vượt quá số BACKLOG cho phép
- Client thoát giữa chừng mà không xin thoát.

Chương trình đã đảm bảo vượt qua các trường hợp kiểm thử ở trên.

IV. Kết luận

Mô hình kết nối client – server thông qua socket có rất nhiều ưu điểm : khả năng truyền dữ liệu không giới hạn, có thể truyền dữ liệu dưới mọi định dạng khác nhau ví dụ : kí tự, nhị phân.

Khả năng giao tiếp dựa trên địa chỉ IP và cổng kết nối cho phép chúng ta có thể cung cấp nhiều dịch vụ khác nhau trên một server trên các cổng khác nhau. Và trên mỗi cổng đó có thể phục vụ nhiều client khác nhau cùng một lúc.

TÀI LIỆU THAM KHẢO

- [1] Slide và bài giảng của thầy **Bùi Trọng Tùng**.
- [2] Socket Programming in C : Michael J. Donahoo
Kenneth L. Canvert