

# **Bài tập thực hành**

## **Môn: TH Lập Trình Mạng**

### **Chương 7: Thiết kế Application Protocol**

#### **1. MỤC TIÊU**

Ở bài học này, sinh viên sẽ học cách thiết kế Trò Chơi Cờ ca-rô Sử Dụng TCP Sockets. Cụ thể, sinh viên sẽ được thực hành các kỹ năng sau:

- Thực hành thiết kế giao thức truyền thông giữa server và client cho một trò chơi Cờ ca-rô nhiều người chơi đơn giản.
- Cài đặt trò chơi bằng cách trao đổi thông điệp giữa client và server sử dụng giao thức truyền thông đã thiết kế.

#### **2. YÊU CẦU**

- Hiểu biết cơ bản về lập trình C.
- Kiến thức về lập trình socket TCP.

#### **3. BÀI THỰC HÀNH**

##### **Phần 1: Các Bước Thiết Kế Ứng Dụng**

Quá trình thiết kế bắt đầu từ việc hiểu cách server và client tương tác với nhau và các thông điệp sẽ được trao đổi như thế nào. Các phần sau sẽ hướng dẫn từng bước thiết kế.

##### **1.1 Xác Định Vấn Đề**

Chúng ta muốn thiết kế một trò chơi Cờ ca-rô đơn giản theo lượt, trong đó hai client kết nối đến server và lần lượt đưa ra các nước đi. Server sẽ điều khiển luồng trò chơi bằng cách luân phiên lượt giữa hai client, nhận các nước đi và kiểm tra điều kiện thắng hoặc hòa. Khi trò chơi kết thúc, server sẽ thông báo kết quả cho cả hai client.

##### **1.2 Thiết Kế Giao Thức Truyền Thông**

Server và client sẽ giao tiếp với nhau bằng các loại thông điệp được định trước. Mỗi thông điệp sẽ có mục đích riêng (ví dụ: gửi nước đi, cập nhật trạng thái trò chơi, hoặc thông báo kết quả). Server sẽ đóng vai trò là người điều khiển trò chơi, quyết

định lượt của ai và cập nhật bảng trò chơi sau mỗi nước đi.

Loại Thông điệp:

- MOVE (0x02): Được gửi từ client để gửi nước đi (hàng và cột).
- STATE\_UPDATE (0x03): Được gửi từ server cho cả hai client để cập nhật bảng trò chơi sau mỗi nước đi.
- RESULT (0x04): Được gửi từ server để thông báo kết quả trò chơi (thắng/thua/hòa).
- TURN\_NOTIFICATION (0x05): Được gửi từ server để thông báo cho client rằng đến lượt họ đi.

Định Dạng Thông điệp:

- Loại Thông điệp (1 byte): Xác định loại thông điệp.
- Payload: Dữ liệu bổ sung tùy thuộc vào loại thông điệp.
  - + Đối với MOVE: Vị trí hàng và cột của nước đi.
  - + Đối với STATE\_UPDATE: Trạng thái hiện tại của bảng trò chơi.
  - + Đối với RESULT: Kết quả trò chơi (thắng/thua/hòa)

Ví dụ Thông điệp:

- MOVE (Client gửi đến Server): [0x02] [hàng] [cột]
- TURN\_NOTIFICATION (Server gửi đến Client): [0x05]
- STATE\_UPDATE (Server gửi đến các Client): [0x03] [trạng thái bảng]
- RESULT (Server gửi đến các Client): [0x04] [kết quả] (trong đó kết quả là thắng, thua, hoặc hòa).

### 1.3 Xác Định Luồng Trò Chơi và Vai Trò

Trò chơi Cờ ca-rô là trò chơi theo lượt, vì vậy luồng trò chơi cần phải luân phiên giữa hai client. Server sẽ quản lý luồng trò chơi, đảm bảo rằng:

1. Các client lần lượt thực hiện nước đi.
2. Mỗi nước đi đều được kiểm tra và áp dụng vào bảng trò chơi.
3. Server kiểm tra trạng thái trò chơi sau mỗi nước đi để xác định thắng hay hòa.
4. Server cập nhật bảng trò chơi và thông báo kết quả cho cả hai client khi trò chơi kết thúc.

Luồng Trò Chơi:

1. Server bắt đầu: Server chờ hai client kết nối.
2. Lượt chơi: Server thông báo cho người chơi hiện tại rằng đến lượt họ chơi.
3. Gửi nước đi: Người chơi gửi nước đi (hàng và cột), và server cập nhật bảng trò chơi.
4. Cập nhật trạng thái trò chơi: Server gửi bảng cập nhật cho cả hai người chơi.
5. Kiểm tra thắng/hòa: Sau mỗi nước đi, server kiểm tra xem trò chơi đã kết thúc chưa.
6. Thông báo kết quả: Nếu trò chơi kết thúc, server gửi kết quả cho cả hai người chơi và đóng kết nối.

Vai Trò:

- Server:
  - + Điều khiển luồng trò chơi.
  - + Luân phiên lượt giữa hai người chơi.
  - + Nhận và xác thực nước đi.
  - + Gửi cập nhật và kết quả cho cả hai client.
- Client:
  - + Chờ server thông báo khi đến lượt.
  - + Gửi nước đi hợp lệ khi đến lượt.
  - + Nhận cập nhật trạng thái trò chơi và kết quả từ server.

## 1.4 Cấu Trúc Dữ Liệu và Xử Lý Thông Điệp

Bảng Trò Chơi:

Bảng trò chơi là một lưới 3x3 được biểu diễn dưới dạng mảng hai chiều của các số nguyên. Mỗi ô trên bảng có thể có một trong các giá trị sau:

- 0: Ô trống.
- 1: Nước đi của Người chơi 1 (X).
- 2: Nước đi của Người chơi 2 (O).

```
int game_board[3][3];
```

Xác Thực Nước Đi:

Server phải xác nhận rằng nước đi của người chơi nằm trong giới hạn và ô được chọn chưa bị chiếm.

Logic Trò Chơi:

Server sẽ kiểm tra điều kiện thắng sau mỗi nước đi. Người chơi thắng nếu họ đặt được ba dấu liên tiếp theo hàng, cột, hoặc chéo. Server cũng sẽ kiểm tra nếu trò chơi kết thúc với kết quả hòa (khi tất cả các ô đã được đi mà không có ai thắng).

---

## Phần 2: Thiết Kế Server Từng Bước

### 2.1 Thiết Lập Kết Nối

- Server phải thiết lập kết nối với hai client trước.
- Server lắng nghe trên một cổng cụ thể (ví dụ: 8080) cho các kết nối TCP đến.
- Khi cả hai client kết nối, server bắt đầu trò chơi bằng cách luân phiên lượt giữa hai client.

Ví Dụ Mã Lệnh:

```
listen(tcp_sock, 2); // Lắng nghe kết nối
// Người chơi 1 kết nối
client1_sock = accept(tcp_sock, (struct
sockaddr*)&client_addr, &addr_len);

// Người chơi 2 kết nối
client2_sock = accept(tcp_sock, (struct
sockaddr*)&client_addr, &addr_len);
```

### 2.2 Luân Phiên Lượt Chơi

- Server luân phiên giữa Người chơi 1 và Người chơi 2.
- Server gửi TURN\_NOTIFICATION cho người chơi hiện tại, chờ nước đi của họ, và cập nhật trạng thái trò chơi.

Ví Dụ Mã Lệnh:

```
// Thông báo cho Người chơi 1 rằng đến lượt họ
notify_turn(client1_sock);

// Nhận nước đi từ Người chơi 1
recv(client1_sock, buffer, BUFFER_SIZE, 0);
```

## 2.3 Xác Thực Nước Đi và Cập Nhật Trạng Thái Trò Chơi

- Sau khi nhận được nước đi từ người chơi hiện tại, server kiểm tra nếu nước đi hợp lệ (tức là ô còn trống).
- Nếu hợp lệ, server cập nhật bảng trò chơi và gửi trạng thái mới cho cả hai người chơi.

Ví Dụ Mã Lệnh:

```
if (game_board[row][col] == 0) {  
    game_board[row][col] = current_player;  
    move_count++;  
}
```

## 2.4 Kiểm Tra Thắng hoặc Hòa

- Sau mỗi nước đi, server kiểm tra nếu có người thắng hoặc nếu trò chơi kết thúc với hòa.
- Nếu trò chơi kết thúc, server gửi thông báo RESULT cho cả hai client và đóng kết nối.

Ví Dụ Mã Lệnh:

```
int winner = check_winner();  
if (winner > 0 || move_count == 9) {  
    buffer[0] = 0x04; // Loại thông điệp RESULT  
    buffer[1] = (winner > 0) ? winner : 0x03; // 0x03  
    cho hòa  
    send(client1_sock, buffer, BUFFER_SIZE, 0);  
    send(client2_sock, buffer, BUFFER_SIZE, 0);  
}
```

---

## Phần 3: Thiết Kế Client Từng Bước

### 3.1 Thiết Lập Kết Nối

- Client thiết lập kết nối với server bằng TCP và chờ lượt chơi.

Ví Dụ Mã Lệnh:

```
// Kết nối đến server
connect(tcp_sock, (struct sockaddr*)&server_addr,
sizeof(server_addr));
```

### 3.2 Nhận Thông Báo Lượt và Gửi Nước Đi

- Client chờ nhận thông báo TURN\_NOTIFICATION từ server.
- Khi đến lượt, client nhập nước đi (hàng và cột) và gửi đến server thông qua thông điệp MOVE.

Ví Dụ Mã Lệnh:

```
recv(tcp_sock, buffer, BUFFER_SIZE, 0); // Chờ thông
báo lượt
printf("Nhập nước đi của bạn (hàng và cột): ");
scanf("%d %d", &row, &col); // Nhập nước đi
buffer[0] = 0x02; // Thông điệp MOVE
buffer[1] = row;
buffer[2] = col;
send(tcp_sock, buffer, BUFFER_SIZE, 0); // Gửi nước đi
đến server
```

### 3.3 Nhận Cập Nhật Trạng Thái Trò Chơi và Kết Quả

- Client nhận các thông báo STATE\_UPDATE để cập nhật bảng trò chơi và hiển thị nó.
- Khi trò chơi kết thúc, client nhận thông báo RESULT từ server.

Ví Dụ Mã Lệnh:

```
recv(tcp_sock, buffer, BUFFER_SIZE, 0); // Nhận cập
nhật trạng thái hoặc kết quả
if (buffer[0] == 0x03) {
    // Cập nhật bảng trò chơi
} else if (buffer[0] == 0x04) {
    // Xử lý kết quả trò chơi (thắng/thua/hòa)
}
```

---

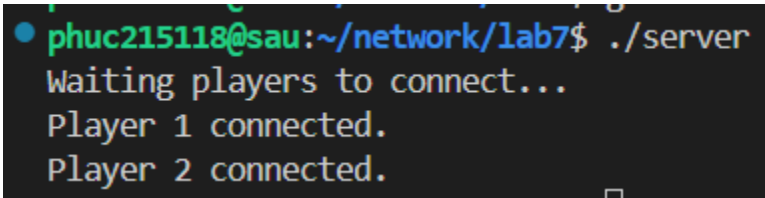
## Phần 4: Kiểm Tra Ứng Dụng

1. Kiểm Tra Luồng Trò Chơi Cơ Bản:
  - Chạy server và kết nối hai client.
  - Đảm bảo rằng các client có thể luân phiên gửi nước đi, và bảng trò chơi cập nhật chính xác cho cả hai người chơi.
2. Điều Kiện Thắng và Hòa:
  - Kiểm tra các kịch bản trong đó Người chơi 1 hoặc Người chơi 2 thắng.
  - Kiểm tra các kịch bản trong đó trò chơi kết thúc với hòa (không còn nước đi nào hợp lệ).
3. Nước Đi Không Hợp Lệ:
  - Kiểm tra tình huống khi một người chơi cố gắng đi vào ô đã bị chiếm.
  - Server sẽ bỏ qua nước đi không hợp lệ và yêu cầu người chơi thử lại.

## Yêu Cầu Nộp Bài

Mỗi sinh viên phải nộp:

- Nộp mã nguồn cho cả server và client.
- Cung cấp một báo cáo ngắn mô tả:
  - + Định dạng thông điệp và giao thức sử dụng.
    - *Định dạng thông điệp:*
      - 0x02: Client gửi nước đi*
      - 0x03: Server gửi board hiện tại*
      - 0x04: Server gửi kết quả*
      - 0x05: Server gửi thông báo đến lượt chơi của client*
      - 0x06: Server thông báo nước đi không hợp lệ (nằm ngoài giới hạn board hoặc ô đã được chọn)*
    - *Giao thức sử dụng: Giao thức TCP*

- + Luồng trò chơi và cách xử lý lượt chơi.
  1. *Server bắt đầu: Server chờ hai client kết nối.*
  2. *Lượt chơi: Server thông báo cho người chơi hiện tại rằng đến lượt họ chơi.*
  3. *Gửi nước đi: Người chơi gửi nước đi (tọa độ hàng và cột) đến server. Server kiểm tra tính hợp lệ của nước đi (nếu không hợp lệ thì yêu cầu người chơi gửi lại) và cập nhật bảng trò chơi.*
  4. *Cập nhật trạng thái trò chơi: Server gửi bảng cập nhật cho cả hai người chơi.*
  5. *Kiểm tra thắng/hòa: Sau mỗi nước đi, server kiểm tra xem trò chơi đã kết thúc chưa.*
  6. *Thông báo kết quả: Nếu trò chơi kết thúc, server gửi kết quả cho cả hai người chơi và đóng kết nối.*
- + Những thách thức gặp phải trong quá trình thiết kế và triển khai.
  - *Kiểm tra nước đi hợp lệ: Cần kiểm tra xem người chơi có gửi tọa độ hợp lệ trong bảng hay gửi tọa độ đã được chọn trước đó hay không.*  
→ *Kiểm tra giá trị nhập có trong khoảng  $[0, 2]$  và ô được chọn có giá trị không (ô trống có giá trị là 0)*
  - *Kiểm tra thắng/hòa: Cần kiểm tra có người thắng không (hàng, cột, đường chéo) hoặc hòa nếu full bảng.*  
→ *Kiểm tra các hàng, cột, đường chéo có người chơi nào đã điền đủ 3 ô liên tiếp không.*
- + Ảnh chụp màn hình kết quả thử nghiệm cho thấy trò chơi đã chạy thành công
  - *Server:*


```

phuc215118@sau:~/network/lab7$ ./server
Waiting players to connect...
Player 1 connected.
Player 2 connected.

```
  - *Client 1:*



```

phuc215118@sau:~/network/lab7$ ./client
Connected to the server.
It's your turn. Enter your move (row and column, 0-2): 0 0
Current game board:
X . .
. . .
. . .
Current game board:
X O .
. . .
. . .
It's your turn. Enter your move (row and column, 0-2): 1 1
Current game board:
X O .
. X .
. . .
Current game board:
X O O
. X .
. . .
It's your turn. Enter your move (row and column, 0-2): 2 2
Player 1 wins!

```

- *Client 2:*

```

● phuc215118@sau:~/network/lab7$ ./client
Connected to the server.
Current game board:
x . .
. . .
. . .
It's your turn. Enter your move (row and column, 0-2): 0 3

Invalid move! Please try again.
Current game board:
x . .
. . .
. . .
It's your turn. Enter your move (row and column, 0-2): 0 1
Current game board:
x o .
. . .
. . .
Current game board:
x o .
. x .
. . .
It's your turn. Enter your move (row and column, 0-2): 0 2
Current game board:
x o o
. x .
. . .
Player 1 wins!

```