네트워크 프로그래밍 과제 #08

☞ 주의 사항

- 아래의 조건(TCP 소켓, 구조체, 멀티 쓰레드 사용)등을 따르지 않고 구현한 경우, 0 점 처리함
- 기능 구현과 상관없이 프로그램 동작 중 예외가 발생되는 경우, 각 항목별 -2 점 감점함
- 각 기능별 부분 점수는 없음 (기능에 대한 코드만 있고 동작이 안되면 점수 없음)
- 각 소스 파일에 학번, 이름(영문 가능) 주석이 없는 경우, 파일당 -1 점 감점

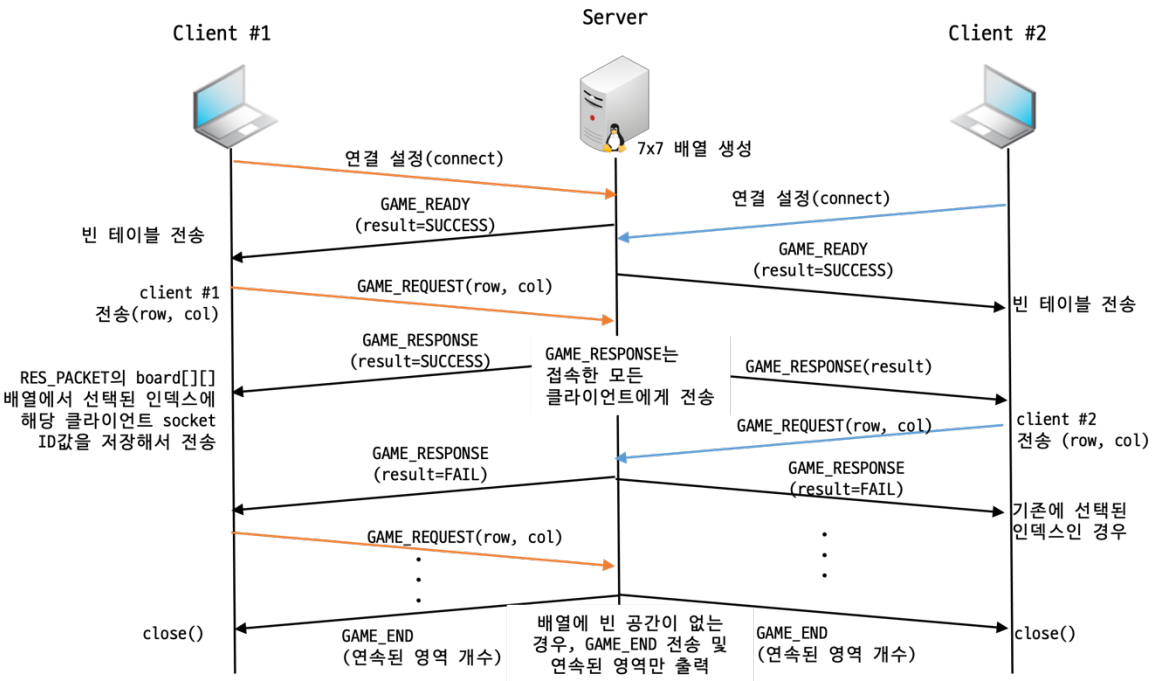
1. 멀티쓰레드를 이용한 연속된 영역 계산 프로그램 (20 점)

제출파일: land_server.c land_client.c

■ 동작 과정

- ✓ 서버는 7x7 배열을 생성하고 0으로 초기화 시킴
- ✓ 3개의 클라이언트는 각각 랜덤하게 배열의 위치를 선택
- ✓ 서버는 각 클라이언트가 선택한 내용을 모든 클라이언트에게 전송
- ✓ 7x7 배열에 더 이상 선택할 공간이 없는 경우에 게임을 종료함
- ✓ 게임을 종료한 다음, 각 클라이언트가 차지한 연속된 영역만 화면에 표시하고 영역의 개수를 출력함(따로 떨어져 있는 영역은 계산하지 않고 표시하지 않음)

■ 메시지 송수신 프로토콜



■ 데이터 전송 구조체

```
#define ROW 7
#define COL 7
// cmd field value
#define GAME_READY      0   // Server -> Client
#define GAME_REQUEST    1   // Client -> Server
#define GAME_RESPONSE   2   // Server -> Client
#define GAME_END        3   // Server -> Client

// result 값
#define FAIL        0   // 해당 인덱스에 다른 클라이언트가 선택한 경우
#define SUCCESS     1   // 해당 인덱스를 선택하는 데 성공

// Client -> Server (GAME_REQUEST 메시지)
typedef struct {
    int cmd; // cmd field value
    int row; // 랜덤하게 선택한 row 값
    int col; // 랜덤하게 선택한 col 값
}REQ_PACKET;

// Server -> Client (GAME_READY, GAME_RESPONSE, GAME_END 메시지)
typedef struct {
    int cmd;            // cmd field value
    int board[ROW][COL]; // 클라이언트가 선택한 값(Thread id: 4, 5, 6)
    int result;         // result 값: FAIL, SUCCESS
}RES_PACKET
```

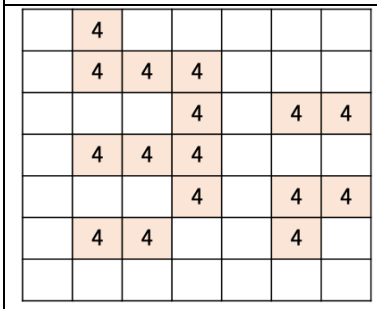
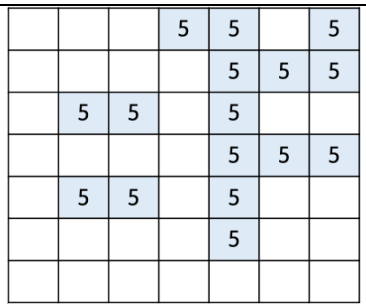
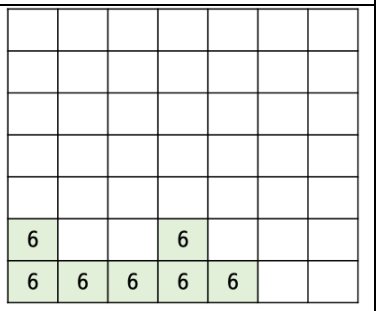
■ 연속된 영역 계산 방법

X 5	4	X 6	5	5	X 4	5
X 6	4	4	4	5	5	5
X 4	5	5	4	5	4	4
X 5	4	4	4	5	5	5
X 4	5	5	4	5	4	4
6	4	4	6	5	4	X 6
6	6	6	6	6	X 5	X 4

연속된 영역 계산
- 대각선 셀을 제외한 인접한 영역
- 가로, 세로로 인접한 셀에 같은 숫자가 최소 2개 이상 존재하는 경우임

X 로 표시된 숫자는 연속된 공간에서 제외됨 (주변에 동일한 숫자가 없이 단독으로 존재함)
4 의 영역의 크기: 16
5 의 영역의 크기: 16
6 의 영역의 크기: 7

■ 출력 양식: 각 클라이언트 id별로 분리해서 출력함

Client 4 (Space size: 16)	Client 5 (Space size: 16)	Client 6 (Space size: 7)
		

■ 서버 기능 (15점)

✓ 공유 자원에 대한 동기화 기능이 없는 경우, 정상 동작되는 것처럼 보여도 -10점 감점함

✓ **2차원 배열 생성**
- 서버 프로그램이 시작되면 7x7 크기의 2차원 배열을 생성
- `int board[7][7] = {0}; // 공유 자원`

✓ GAME_READY 메시지 전송 기능 (1 점)
- 서버는 클라이언트가 접속을 하면 GAME_READY 메시지를 전송하고 클라이언트에게 준비가 되었음을 알림(board[][]의 내용은 0으로 초기화해서 전송)

✓ GAME_REQUEST 메시지 수신 기능 (2 점)
- 클라이언트가 전송한 랜덤 번호(row, col)를 수신하면 서버가 생성한 배열에 빈 공간인 경우 해당 클라이언트의 id를 표시 (Client 1: 4, Client 2: 5, Client 3: 6 표시)
- 다른 클라이언트가 이미 선택한 경우에는 "[Client id] already chose.(행번호, 열번호)chose."를 화면에 출력하고 기존 선택은 유지함
- 3개의 클라이언트가 공유 자원에 접근하기 때문에 동기화 기능 구현 필수
- 클라이언트에 출력되는 내용과 서버에 출력되는 내용이 다를 경우 점수 없음

✓ GAME_RESPONSE 전송 기능 (2점)
- 서버는 GAME_REQUEST 메시지의 응답으로, 각 클라이언트가 선택한 위치를 board[][]에 저장하여 모든 클라이언트에게 전송
- GAME_RESPONSE 메시지의 result 필드는 상황에 따라 FAIL, SUCCESS을 전송하며 화면에 내용 출력

✓ 연속된 공간 계산 및 화면 출력 기능 (9점)
- 클라이언트가 배열을 모두 선택한 경우, 연속된 공간을 확인해서 클라이언트 id별로 연속된 공간을 출력 양식의 그림과 같이 화면에 출력 (6점, 각 2점)
- 각 클라이언트별로 연속된 공간의 개수를 화면에 출력 (3점)

✓ GAME_END 메시지 전송 (1 점)
- 클라이언트가 7x7 배열을 모두 선택한 경우, GAME_END 메시지를 접속한 모든 클라이언트에게 전송

■ 클라이언트 기능 (5점)

✓ 서버로 부터 메시지를 수신(①)한 다음, 전송(②)하기 위해 semaphore 사용 (2점)
  - 두 개의 쓰레드 생성: recv_msg, send_msg

✓ GAME_REQUEST 전송 기능 (1 점)
  - 서버에 접속 후, 초기 GAME_READY 메시지를 수신하면 클라이언트는 행(row)과 열(col) 값을 랜덤하게 생성하고 GAME_REQUEST 메시지를 서버로 전송 (REQ_PACKET 사용)
  - 각 클라이언트는 sleep(1)함수 사용

✓ GAME_RESPONSE 수신 기능 (1 점)
  - 각 클라이언트가 선택한 위치를 화면에 출력

✓ GAME_END 수신 기능 (1 점)
  - GAME_RESPONSE 메시지 수신과 같이 클라이언트가 맞힌 숫자 현황 출력

■ **실행 결과: 제공된 동영상 파일 참조**

<table>
<tr><th>서버 실행 초기 화면</th><th>클라이언트 #1</th></tr>
<tr><td rowspan="2">$ ./server 9190
+-----------------------------------+
¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦
+-----------------------------------+
¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦
+-----------------------------------+
¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦
+-----------------------------------+
¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦
+-----------------------------------+
¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦
+-----------------------------------+
¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦
+-----------------------------------+
¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦
+-----------------------------------+
Occupied: 0

// 동작 과정 생략
. . .
[Tx] cmd: 2, clinet_id: 4, result: 1
[Tx] cmd: 2, clinet_id: 5, result: 1
[Tx] cmd: 2, clinet_id: 6, result: 1</td><td>$ ./client 127.0.0.1 9190

. . .

[Tx] cmd: 1, index(4, 3)

[Rx] cmd: 3, result: 1
GAME_END. Game is over!
+-----------------------------------+
¦ 5 ¦ 4 ¦ 6 ¦ 5 ¦ 5 ¦ 4 ¦ 5 ¦
+-----------------------------------+
¦ 6 ¦ 4 ¦ 4 ¦ 4 ¦ 5 ¦ 5 ¦ 5 ¦
+-----------------------------------+
¦ 4 ¦ 5 ¦ 5 ¦ 4 ¦ 5 ¦ 4 ¦ 4 ¦
+-----------------------------------+
¦ 5 ¦ 4 ¦ 4 ¦ 4 ¦ 5 ¦ 5 ¦ 5 ¦
+-----------------------------------+
¦ 4 ¦ 5 ¦ 5 ¦ 4 ¦ 5 ¦ 4 ¦ 4 ¦
+-----------------------------------+
¦ 6 ¦ 4 ¦ 4 ¦ 6 ¦ 5 ¦ 4 ¦ 6 ¦
+-----------------------------------+
¦ 6 ¦ 6 ¦ 6 ¦ 6 ¦ 6 ¦ 5 ¦ 4 ¦
+-----------------------------------+</td></tr>
<tr><th>클라이언트 #2</th></tr>
</table>

```
[Rx] client: 5, cmd: 1, index:(4, 3)
+---------------------------------+
|  5 |  4 |  6 |  5 |  5 |  4 |  5 |
+---------------------------------+
|  6 |  4 |  4 |  4 |  5 |  5 |  5 |
+---------------------------------+
|  4 |  5 |  5 |  4 |  5 |  4 |  4 |
+---------------------------------+
|  5 |  4 |  4 |  4 |  5 |  5 |  5 |
+---------------------------------+
|  4 |  5 |  5 |  4 |  5 |  4 |  4 |
+---------------------------------+
|  6 |  4 |  4 |  6 |  5 |  4 |  6 |
+---------------------------------+
|  6 |  6 |  6 |  6 |  6 |  5 |  4 |
+---------------------------------+
Occupied: 49
Game board is full. Game is over.

[Client 4] Continuous Space
+---------------------------------+
|    |  4 |    |    |    |    |    |
+---------------------------------+
|    |  4 |  4 |  4 |    |    |    |
+---------------------------------+
|    |    |    |  4 |    |  4 |  4 |
+---------------------------------+
|    |  4 |  4 |  4 |    |    |    |
+---------------------------------+
|    |    |    |  4 |    |  4 |  4 |
+---------------------------------+
|    |  4 |  4 |    |    |  4 |    |
+---------------------------------+
|    |    |    |    |    |    |    |
+---------------------------------+
Space size: 16

[Client 5] Continuous Space
+---------------------------------+
|    |    |    |  5 |  5 |    |  5 |
+---------------------------------+
|    |    |    |    |  5 |  5 |  5 |
+---------------------------------+
|    |  5 |  5 |    |  5 |    |    |
+---------------------------------+
|    |    |    |    |  5 |  5 |  5 |
+---------------------------------+
|    |  5 |  5 |    |  5 |    |    |
+---------------------------------+
|    |    |    |    |  5 |    |    |
+---------------------------------+
|    |    |    |    |    |    |    |
+---------------------------------+
```

```
$ ./client 127.0.0.1 9190

. . .
[Rx] cmd: 3, result: 1
GAME_END. Game is over!
+---------------------------------+
|  5 |  4 |  6 |  5 |  5 |  4 |  5 |
+---------------------------------+
|  6 |  4 |  4 |  4 |  5 |  5 |  5 |
+---------------------------------+
|  4 |  5 |  5 |  4 |  5 |  4 |  4 |
+---------------------------------+
|  5 |  4 |  4 |  4 |  5 |  5 |  5 |
+---------------------------------+
|  4 |  5 |  5 |  4 |  5 |  4 |  4 |
+---------------------------------+
|  6 |  4 |  4 |  6 |  5 |  4 |  6 |
+---------------------------------+
|  6 |  6 |  6 |  6 |  6 |  5 |  4 |
+---------------------------------+
```

클라이언트 #3

```
$ ./client 127.0.0.1 9190

. . .
[Rx] cmd: 3, result: 1
GAME_END. Game is over!
+---------------------------------+
|  5 |  4 |  6 |  5 |  5 |  4 |  5 |
+---------------------------------+
|  6 |  4 |  4 |  4 |  5 |  5 |  5 |
+---------------------------------+
|  4 |  5 |  5 |  4 |  5 |  4 |  4 |
+---------------------------------+
|  5 |  4 |  4 |  4 |  5 |  5 |  5 |
+---------------------------------+
|  4 |  5 |  5 |  4 |  5 |  4 |  4 |
+---------------------------------+
|  6 |  4 |  4 |  6 |  5 |  4 |  6 |
+---------------------------------+
|  6 |  6 |  6 |  6 |  6 |  5 |  4 |
+---------------------------------+
```

```
Space size: 16

[Client 6] Continuous Space
+--------------------------------+
|    |    |    |    |    |    |    |    |
+--------------------------------+
|    |    |    |    |    |    |    |    |
+--------------------------------+
|    |    |    |    |    |    |    |    |
+--------------------------------+
|    |    |    |    |    |    |    |    |
+--------------------------------+
|    |    |    |    |    |    |    |    |
+--------------------------------+
|  6 |    |    |  6 |    |    |    |    |
+--------------------------------+
|  6 |  6 |  6 |  6 |  6 |    |    |    |
+--------------------------------+
Space size: 7

[Tx] cmd: 3, clinet_id: 4, result: 1
[Tx] cmd: 3, clinet_id: 5, result: 1
[Tx] cmd: 3, clinet_id: 6, result: 1
Close client_sock: 5, client_cnt: 2
Close client_sock: 4, client_cnt: 1
Close client_sock: 6, client_cnt: 0
```