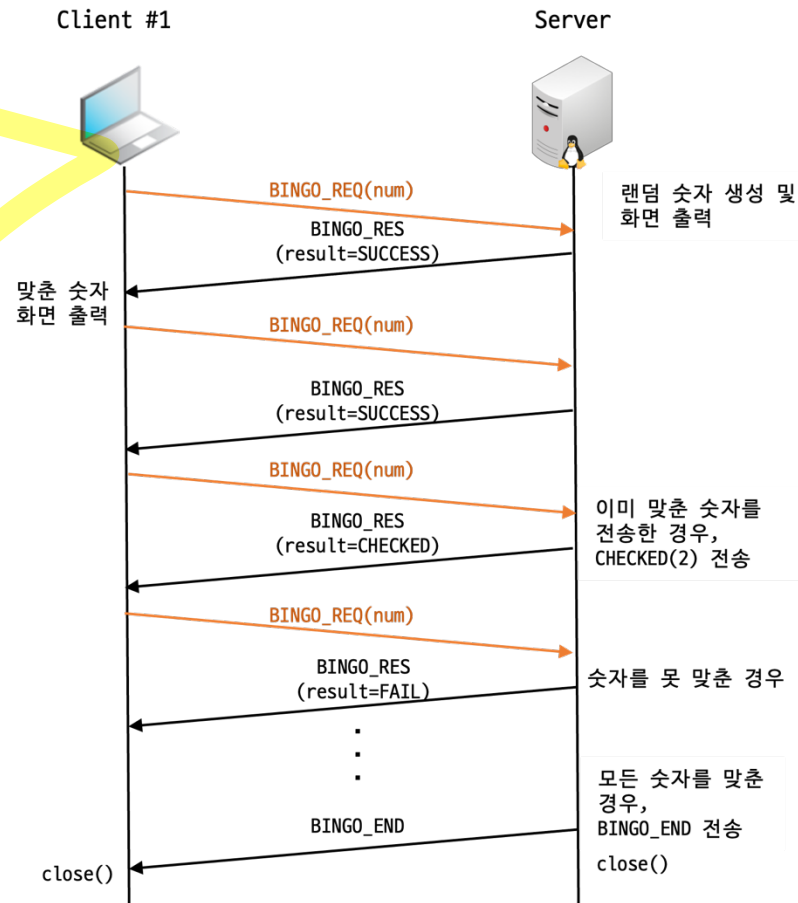


# 컴퓨터망 프로그래밍 과제 #04

## 1. UDP 프로토콜을 이용한 숫자 맞추기 (15 점)

- 제출파일: bingo\_server.c bingo\_client.c

### ■ 메시지 송수신 프로토콜



### ■ 구조체 및 메시지 내용

```
#define RAND_START 1 // 시작 랜덤 값
#define RAND_END 30 // 마지막 랜덤 값

// result field value
#define FAIL 0 // 숫자를 못 맞춘 경우
#define SUCCESS 1 // 숫자를 맞춘 경우
#define CHECKED 2 // 기존에 숫자를 맞춘 경우

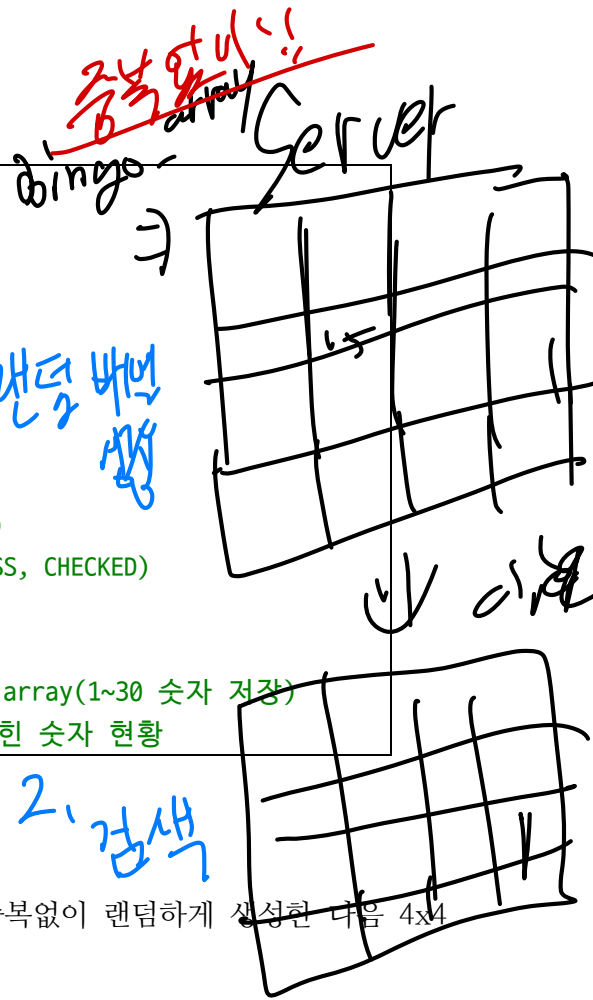
// cmd field value
#define BINGO_REQ 0
#define BINGO_RES 1
#define BINGO_END 2
// Client -> Server 전송 패킷
```

```

typedef struct {
    int cmd;           // BINGO_REQ
    int number;        // 랜덤 생성한 숫자(1~30)
}REQ_PACKET;
// Server -> Client 전송 패킷
typedef struct {
    int cmd;           // BINGO_RES, BINGO_END
    int number;        // 사용자가 선택한 숫자
    int board[ROW][COL]; // bingo 맞춘 현황(0 or bingo number)
    int result;        // result from the server(FAIL, SUCCESS, CHECKED)
}RES_PACKET;

int bingo_array[ROW][COL] = {0, }; // random bingo number array(1~30 숫자 저장)
int player_choice_array[ROW][COL] = {0, }; // Client 가 맞힌 숫자 현황

```



## ■ 서버 기능 (10점)

### ✓ 랜덤 숫자 생성 및 화면 출력 (2점)

- 1~30 (RAND\_START ~ RAND\_END) 사이의 숫자를 중복없이 랜덤하게 생성한 다음 4x4 배열에 저장 (int bingo\_array[4][4])
- 랜덤 번호 생성 샘플 소스

```

#include <stdlib.h>
#include <time.h>

```

```

srand(time(NULL)); // 현재 시간 값으로 랜덤 seed 를 초기화
// rand(): 0 ~ RAND_MAX(32767)
rand_num = (rand() % RAND_END) + RAND_START;

```

### ✓ BINGO\_REQ 메시지 수신 기능 (2점)

- 서버는 클라이언트가 전송한 BINGO\_REQ 메시지 내부의 랜덤 번호와 서버가 이미 생성한 랜덤 번호 배열과 일치하는지 비교함

### ✓ BINGO\_RES 메시지 전송 기능 (3점)

- bingo\_array[][]에 동일한 번호가 존재하는 경우, RES\_PACKET의 board[][]에 일치하는 번호의 위치에 해당 숫자를 표시하고 클라이언트에게 전송함 (2점)
- 화면에 일치한 숫자만 화면 출력 (확인 용도) (2점)

### ✓ BINGO\_END 메시지 전송 (3점)

- 클라이언트가 4x4 배열의 숫자를 모두 맞춘 경우, BINGO\_END 메시지를 클라이언트에게 전송
- 클라이언트가 맞추지 못한 숫자가 있는지 확인하는 기능 구현

## ■ 클라이언트 기능 (5점)

작년 기말고사  
원래는  
2명 이상의 클라이언트가  
서버에 접속할 수 있는  
기능

- ✓ 랜덤 값 생성 (1~30 사이의 값) (1점)
- ✓ BINGO\_REQ 전송 기능 (1점)
  - 클라이언트는 랜덤 번호(1~30)를 생성하고 BINGO\_REQ 메시지를 서버로 전송 (REQ\_PACKET 사용)
- ✓ BINGO\_RES 수신 기능 (2점)
  - 클라이언트가 맞힌 숫자 현황을 화면에 출력
- ✓ BINGO\_END 수신 기능 (1점)
  - BINGO\_END 메시지를 수신하면 소켓 `close()` 및 프로그램 종료

■ 서버 실행 화면 (첨부된 동영상 파일 참조)