

#Project Topic#

Name:

임정민

Student ID:

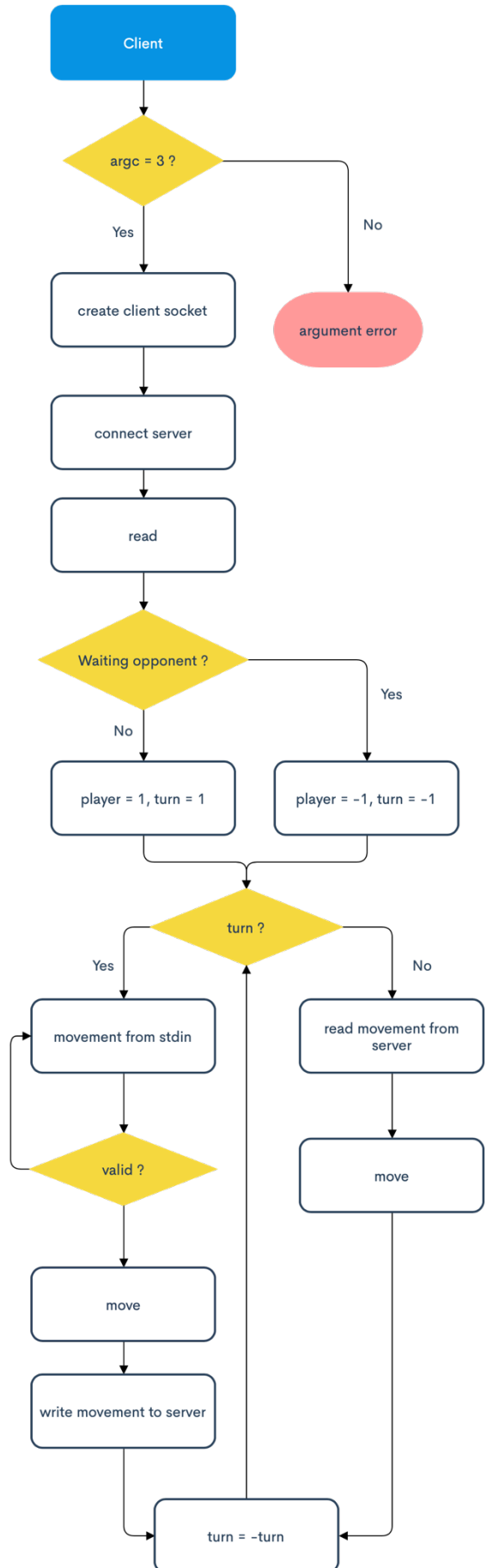
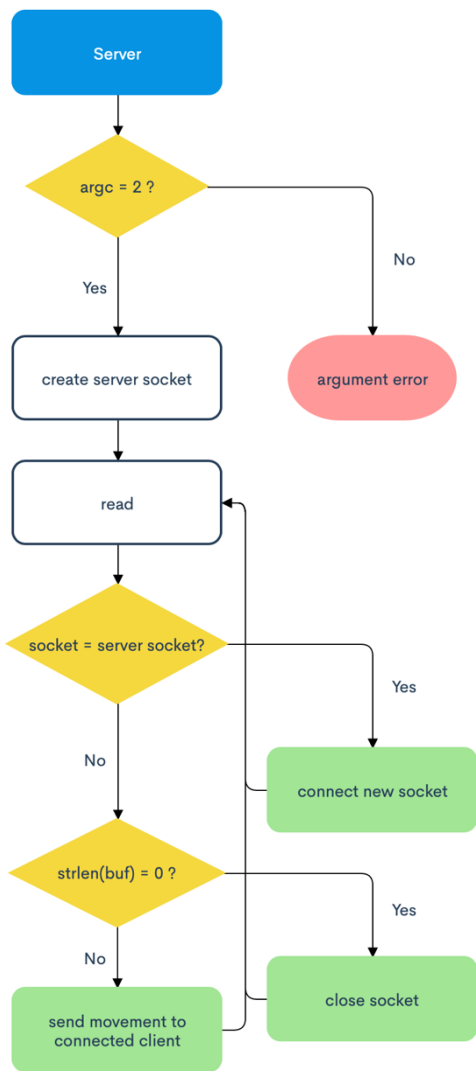
2017116186

1. Introduction

➔ **Abstract:** TCP server-client 모델을 이용한 온라인 체스 게임

➔ **Overview of your solution:**

- multiplexing select 혹은 epoll을 사용한 클라이언트와 클라이언트 간 플레이 서버 구현
- 클라이언트에서 서버와의 연결 및 체스 플레이 작동 구현
- 서버, 클라이언트 flowchart 모델

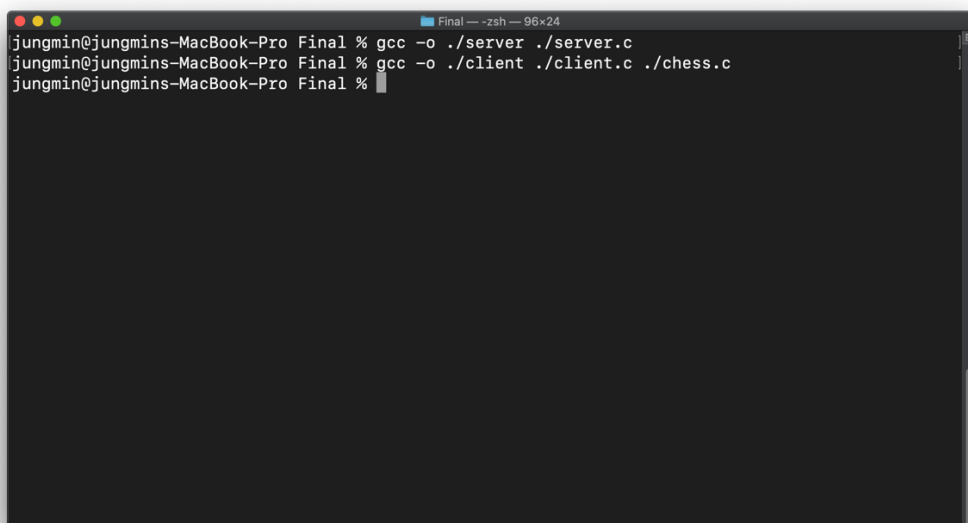


➔ **Program components:**

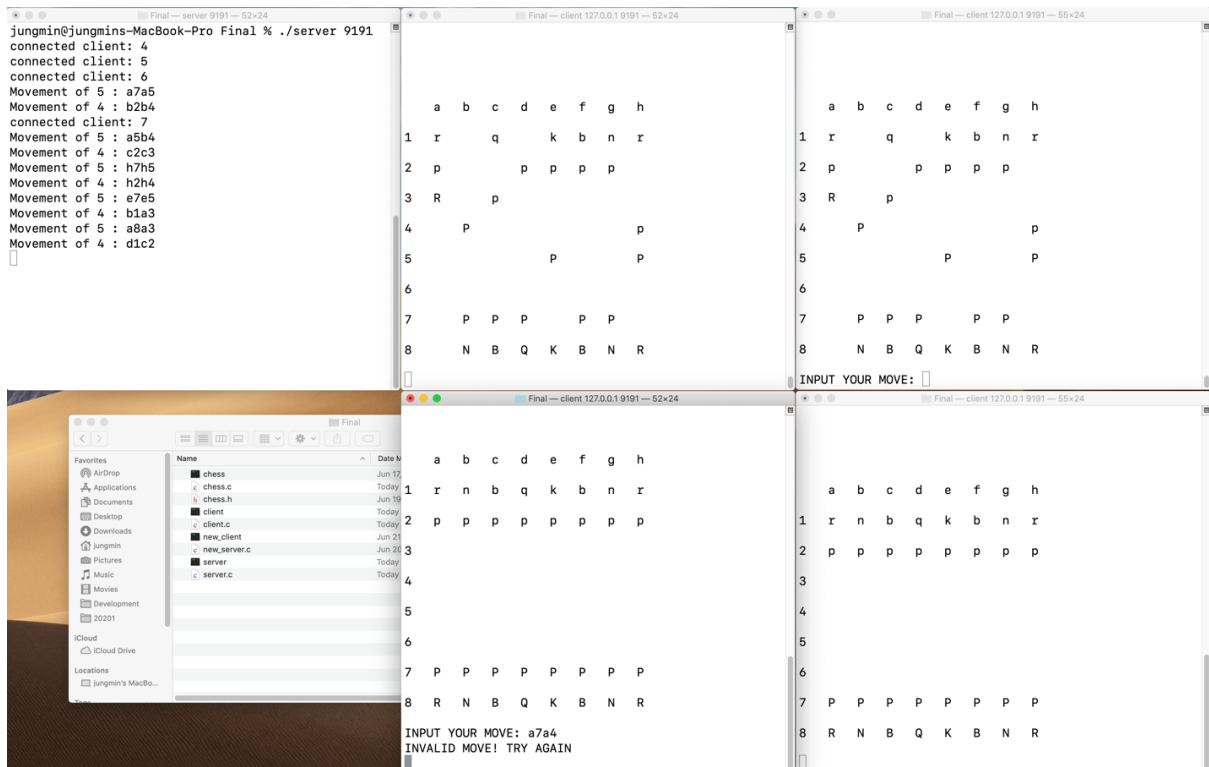
- 클라이언트 체스 플레이 관련 함수
 - void print_board() : clear_board() 함수를 이용하여 터미널을 clear한 후 체스 보드를 출력. black은 대문자로 white는 소문자로 출력
 - void clear_board() : printf("\32[2J"); 를 이용하여 터미널을 clear.
 - int input(char move[], int player) : "e7e5"와 같이 움직일 말의 좌표의 x값, y값, 목표 위치의 x값, y값 순으로 player나 opponent의 movement를 입력받고 valid한지 검사하여 invalid하다면 -1을 return. valid하다면 입력받은 movement 실행 후 상대편의 KING을 잡아 게임이 끝났다면 return 100, 그렇지 않으면 return 1.
 - int king/queen/bishop/knight/rook/pawn(int sx, int sy, int ex, int ey) : input 함수에서 movement의 validity를 확인하기 위한 함수
- 서버의 클라이언트-클라이언트 연결을 위한 자료구조
 - 서버에서 connection 배열을 통해 file descriptor에 연결된 file descriptor를 저장. 만들어지지 않은 경우 -1. 상대의 연결을 기다리고 있는 경우 0을 저장.
 - 서버에서 current 변수를 이용하여 opponent 연결을 기다리는 player를 저장. 이후 새로운 player가 연결될 때 대기중인 current의 player와 연결

2. Results & Explanations

➔ Please paste the screenshots of your execution results here



```
Final — zsh — 96x24
jungmin@jungmins-MacBook-Pro Final % gcc -o ./server ./server.c
jungmin@jungmins-MacBook-Pro Final % gcc -o ./client ./client.c ./chess.c
jungmin@jungmins-MacBook-Pro Final %
```



➔ Briefly describe & explain your results

프로그램의 기능은 멀티플레이 체스 프로그램입니다. 클라이언트 - 서버 간 연결은 TCP를 이용하여 구현하였고, 서버는 여러 클라이언트의 접속을 위해 `multiplexing`의 `select`함수를 사용하여 구현하였습니다. 클라이언트에서 서버에 접속하면 대기중인 플레이어가 없을 경우 새로운 클라이언트가 접속할 때까지 기다리고 대기중인 플레이어가 있을 경우 대기중인 플레이어와 새로운 클라이언트를 매칭하도록 하였습니다. 매칭이 되면 게임을 시작합니다. KING이 잡히면 게임이 끝나고 프로그램이 종료됩니다.