설계

(1) 서버모델 설정

마피아 게임을 구현하기 위해서는 다수의 클라이언트를 동시에 처리할 수 있는 서버모델을 선택해야 되는데 윈도우 환경에서 사용할 수 있는 후보로 select, multi thread, IOCP 가 있었다. select함수는 간단하지만 함수를 호출할 때마다 인자로 매번 관찰대상에 대한 정보를 전달해야 되고 결과값을 받으면 모든 파일 디스크립터를 대상으로 반복하면서 확인해 주어야 한다. multi thread는 연결하는 클라이언트마다 thread를 생성해 주어야 되기 때문에 자원소모가 크다. 하지만 IOCP는 반복문을 통해 핸들을 찾을 필요도 없고 적절한 thread수를 통해 처리할 수 있기 때문에 좋은 성능을 낼 수 있는 구조라고 생각하였고 따라서 IOCP모델을 선택하여 구현하기로 했다.

(2) 게임흐름

1. 클라이언트가 서버에 접속하면 현재 방에 몇 명 들어가있는지 확인할 수 있고 몇 번 방에 들어갈 지 결정할 수 있다. 방들은 서로 독립적으로 게임을 진행한다.

2. 게임 시작 전 플레이어들간 채팅을 통해 소통할 수 있다.

3. 방에 일정 인원이 차면 게임은 자동으로 시작된다.

4. 플레이어들은 역할을 부여 받고 낮이 된다.

5. 낮 동안 플레이어들끼리 채팅을 통해 죽일 플레이어를 정하고 투표를 한다.

6. 투표를 통해 죽은 플레이어를 확인하고 밤이 된다.

7. 밤 동안 마피아들끼리 채팅을 할 수 있으며 죽일 시민을 결정한다.

8. 마피아에 의해 죽은 플레이어를 확인하고 낮이 된다.

9. 마피아 수와 시민수가 같아지거나 마피아가 다 죽을 시 게임은 종료되고 게임 결과를 확인한다.

(3) 데이터 구조 설정

서버와 클라이언트간 데이터를 주고 받을 때 사용되는 구조를 설정하였다.

|  |  |  |
| --- | --- | --- |
| 상태 | 데이터 길이 | 데이터 |

모든 구조는 다음과 같은 구조를 기본으로 한다.

1. 시작 메시지

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | 데이터 길이 | 플레이어 번호 | 역할 | 회의시간 | 투표시간 |

게임 시작 시 서버에서 클라이언트로 보내는 메시지이다.

2. 채팅 메시지

|  |  |  |  |
| --- | --- | --- | --- |
| B | 데이터 길이 | 플레이어 번호 | 채팅 메시지 |

클라이언트에서 서버로 보내면 해당 방에 있는 플레이어들에게 전달된다.

3. 낮 트리거 메시지

|  |  |  |
| --- | --- | --- |
| C | 데이터 길이 | 죽은 플레이어 번호 |

서버에서 클라이언트에게 낮임을 알려주고 밤에 마피아에 의해 죽은 플레이어의 번호를 전달한다.

4. 밤 트리거 메시지

|  |  |  |
| --- | --- | --- |
| D | 데이터 길이 | 죽은 플레이어 번호 |

서버에서 클라이언트에게 밤임을 알려주고 낮에 투표에 의해 죽은 플레이어의 번호를 전달한다.

5. 투표 메시지

|  |  |  |
| --- | --- | --- |
| E | 데이터 길이 | 투표한 플레이어 번호 |

클라이언트에서 서버에게 투표한 플레이어 번호를 전달한다.

6. 마피아 메시지

|  |  |  |
| --- | --- | --- |
| F | 데이터 길이 | 죽일 플레이어 번호 |

클라이언트에서 서버에게 마피아가 죽인 플레이어 번호를 전달한다.

7. 게임 종료 메시지

|  |  |  |
| --- | --- | --- |
| G | 데이터 길이 | 게임 결과 |

서버에서 클라이언트에게 게임이 종료됨을 알려주고 게임결과를 전달한다.

8. 연결 메시지

|  |  |  |
| --- | --- | --- |
| H | 데이터 길이 | 플레이어 번호 |

서버에서 클라이언트에게 방에 접속한 플레이어 번호를 전달한다.

9. 연결해제 메시지

|  |  |  |
| --- | --- | --- |
| I | 데이터 길이 | 플레이어 번호 |

서버에서 클라이언트에게 방에서 나간 플레이어 번호를 전달한다.

10. 방 인원 메시지

|  |  |  |  |
| --- | --- | --- | --- |
| J | 데이터 길이 | 1번방 현재 인원 | 2번방 현재 인원 |

서버에서 클라이언트에게 현재 방의 인원을 전달한다.

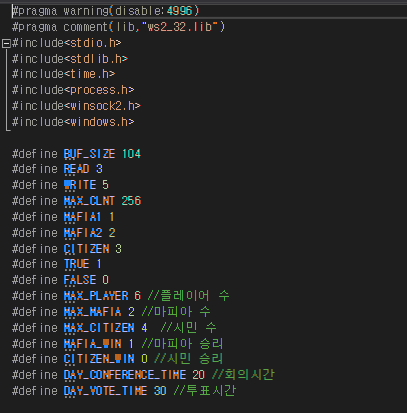
11. 방 선택 메시지

|  |  |  |
| --- | --- | --- |
| K | 데이터 길이 | 방 번호 |

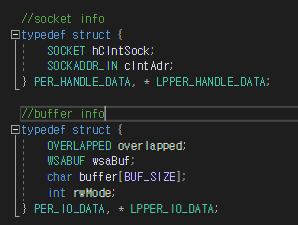
클라이언트에서 서버에게 입장할 방 번호를 전달한다.

코드

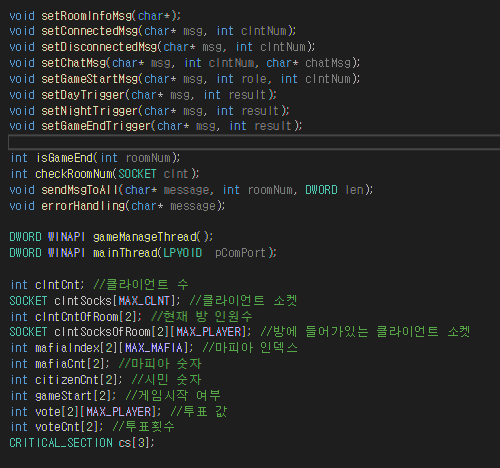
(1) 헤더 파일



구현에 필요한 상수들을 정의 하였다.



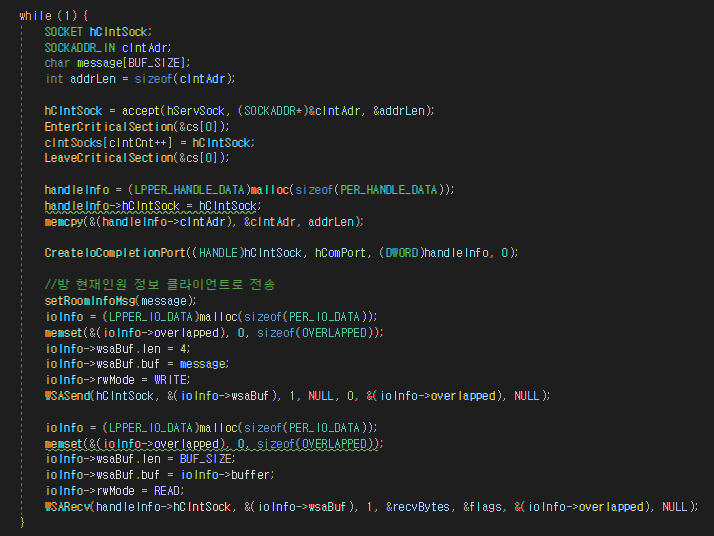
소켓정보와 버퍼정보를 담을 구조체를 선언하였다.



구현에 사용되는 변수와 함수들을 선언하였다.

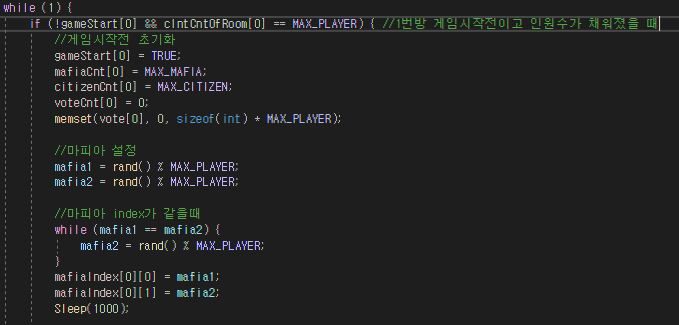
(2) 소스 파일

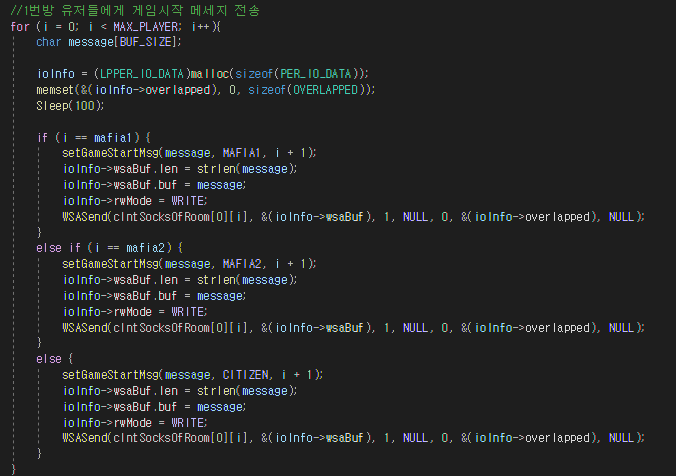
- main 함수



서버는 클라이언트 접속이 있을 경우 setRoomInfoMsg 함수를 통해 방의 현재 인원에 대한 정보 메시지를 설정하고 클라이언트에게 전송한다.

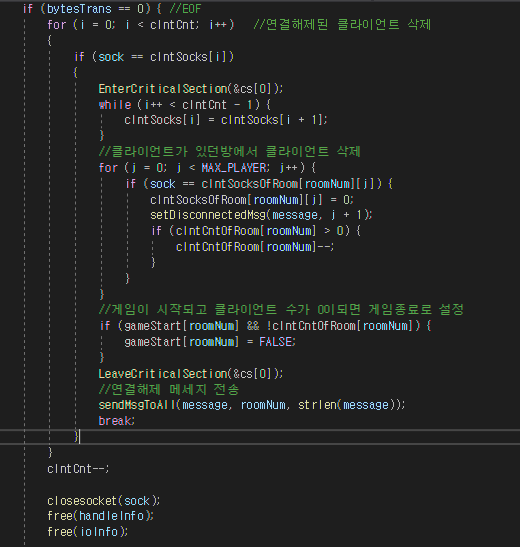
- gameManageThread



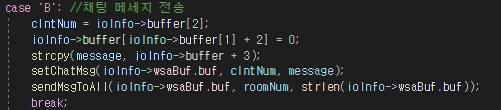


방에 일정 인원이 차면 게임 진행에 필요한 변수들을 초기화 해주며 마피아를 랜덤하게 설정해준다. 그리고 게임시작을 알리는 메시지를 setGameStartMsg를 통해 설정하고 각 플레이이어들에게 전송해준다. 위의 코드는 1번방에 대한 처리이며 2번방도 동일하게 처리된다.

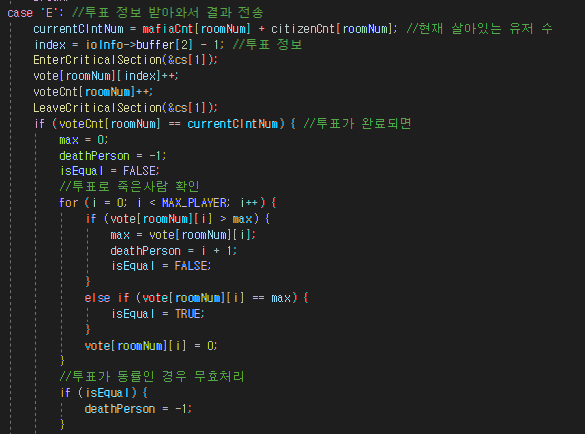
- mainThread



클라이언트에서 연결 해제를 하는 부분이다. 해당 클라이언트 소켓을 배열에서 삭제해주며 count도 감소시킨다. 또한 클라이언트가 방에 입장한 상태에서 연결 해제한 경우 해당 방에서 클라이언트 소켓을 삭제해주고 count도 감소시킨다.



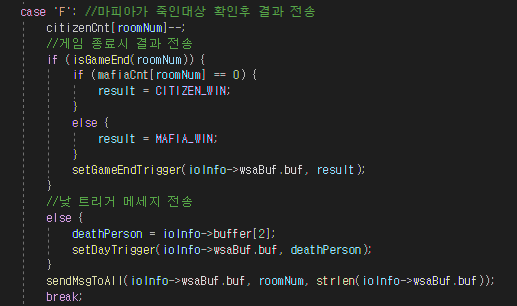
채팅 메시지가 왔을 경우 setChatMsg를 통해 메시지를 설정해주고 그 방에 있는 클라이언트들에게 모두 전송해준다.



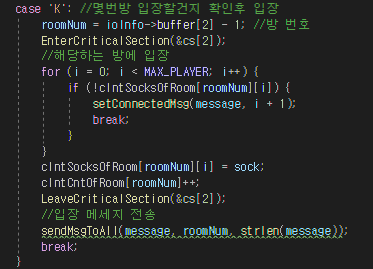


클라이언트에서 투표 정보 메시지를 보낼 경우 가장 많은 투표를 받은 플레이어를 계산한다.

투표한 후 isGameEnd함수를 통해 게임종료 조건을 확인하고 만족하면 setGameEndTrigger을 통해 메시지를 설정하고 만족하지 않으면 setNightTrigger을 통해 메시지를 설정한 후 해당 방의 플레이어들에게 전송해준다.



클라이언트에서 마피아가 죽인 플레이어 번호를 보낼 경우 시민수를 줄이고 게임종료 조건을 확인한 후 메시지를 설정해 게임종료 메시지나 낮 트리거 메시지를 해당 방의 플레이어들에게 전송한다.



클라이언트가 몇번방에 입장할 건지 확인 후 그 방에 해당하는 배열에 소켓을 추가해주고 count를 증가시킨다.

- 사용되는 함수 정의

