* 과제 목표 (도출해야 할 결과)
* 소켓 프로그래밍을 통한 서버 / 클라이언트 구현
* 코드 설명과 과제 해결 방법

1. 서버

* 소켓을 만들고, bind를 통해 ip 주소와 포트 번호를 바인딩한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* 소켓으로 오는 connect 요청을 3개까지 listen 하고, accept 한다.
* accept의 결과로 connection이 나오는데, 이를 이용하여 한 소켓에 여러 연결을 할 수 있다. 이 후로는 send / recv를 이용하여 통신할 수 있다.
* 각 연결들을 리스트에 추가한 뒤, 전체에 메시지를 보낸다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* Client가 순번으로 선택할 난수와, Client 별로 할당될 난수를 생성한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* 3개의 Connection 목록을 순회하며 순서대로 숫자 입력을 받아오고, 그에 따른 결과를 돌려준다. 클라이언트가 보낼 때에는 서버는 수신모드여야 하고, 그 반대의 경우도 마찬가지이므로 때에 맞게 모드를 send / receive 를 설정해준다. 무슨 난수가 골라졌는지 서버 입장에서는 알 수 있도록 하기 위해, 출력문을 추가하였다..

텍스트이(가) 표시된 사진

자동 생성된 설명

* 이 또한 마찬가지로, Connection 목록을 순회하며 순서대로 연산 입력을 받아온 후, 그에 따른 결과를 계산한다.

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

* 점수 목록을 순회하며, 최고점인 경우 승리 메시지를 보내고 아닐 경우 패배 메시지를 보낸다. 이후 connection 들을 모두 닫는다.

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 클라이언트

* 소켓을 만들고, 서버에게 connect 요청을 보낸다. 서버가 listen 중이라면 둘이 연결되어 send / recv 로 통신할 수 있다.

텍스트, 표지판이(가) 표시된 사진

자동 생성된 설명

* 서버가 보낼 때에는 recv 하고, 서버에게 보내야 할 때에는 send를 한다.
* 한쪽이 recv 하고 있을 때에는 다른 쪽은 send를 하고 있어야 한다.
* 마지막에는 소켓을 닫아준다.

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 실행 결과

* 서버와 클라이언트 간 상호작용이 잘 일어남을 확인할 수 있었다.
* <https://youtu.be/J-RD6WOJGCc>

텍스트이(가) 표시된 사진

자동 생성된 설명

* 느낀 점 및 하고싶은 말

코드 설명 스크린샷도 찍고 영상도 찍으려니 너무 힘들다