3) 소켓 바인딩 : 운영체제에 어떤 소켓이 특정 주소와 전송 계층 포트를 쓰겠다고 알려주는 절차

int bind(SOCKET sock, const sockaddr\* address, int address\_len);

sock : **바인딩할 소켓**, address : 보낼 주소가 아니라 **회신 주소**

address\_len : sockaddr 길이

반환값 함수 성공 시 0을 실패 시 -1을 리턴한다.

Sockaddr을 바인딩하는 의미 첫째, 운영체제가 이 주소와 포트를 목적지로 발신된 패킷을 수신하면 운영체제는 이제 이 소켓으로 넘겨준다. 둘째, bind()에서 지정한 주소 및 포트를 이 소켓을 통해 나가는 패킷의 네트워크 계층과 전송 계층 헤더의 발신 주소와 포트로 운영 체제가 쓰게 된다.

4) 리스닝(listening) : 외부에서 들어오는 TCP 핸드셰이킹 첫 단계 요청을 받아 이르 대기열에 저장해 둔다.

Int listen(SOCKET sock, int backlog);

backlog : 연결을 대기열에 둘 최대 숫자 대기열이 가득 차면 그 이후로 들어오려는 연결은 끊어진다. 기본값을 사용하려면 SOMAXONN

성공 시 0, 실패 시 -1을 리턴 한다.

5) accept : 성공하게 되면 원격 호스트와 통신하는 소켓 생성

SOCKET accept(SOCKET sock, sockaddr\* addr, int\* addrlen);

Addr은 accept() 함수가 연결을 요청하는 원격 호스트의 주소를 채워줄 sockaddr 구조체 포인터

Addrlen은 addr 버퍼의 포인터 길이를 반환하는 용도로 사용

6) connect : 에러가 없는 경우 최초 SYN 패킷을 대상 호스트에 전송하여 TCP 핸드셰이킹 개시

Int connect(SOCKET sock, const sockaddr \* addr, int addrlen);

sock : 연결에 사용하고자 하는 소켓

addr : 연결하고자 하는 원격 호스트의 주소를 가리키는 포인터

addrlen : addr 인자의 길이

성공 시 0, 에러가 있는 경우 -1 리턴

7) send : 데이터 전송

Int send(SOCKET sock, const char \* buf, int len, int flags)

sock : 데이터를 보내는데 사용할 소켓

buf : 데이터가 담긴 버퍼

len : 전송할 데이터의 바이트 수

flags : 데이터 전송을 제어하는 비트 플래그, 게임 코드에선 대개 0

에러가 있으면 -1 리턴

성공하면 전송한 데이터의 길이를 리턴한다.

11. recv : 데이터를 받는다.

Int recv(SOCKET sock, char \* buf, int len , int flags);

buf : 데이터를 복사해 넣을 버퍼

len : 버퍼에 넣을 수 있는 데이터 크기의 상한선

flags : 데이터 수신을 제어하는 비트 플래그

에러가 있으면 -1 리턴

성공하면 수신한 바이트 길이를 리턴