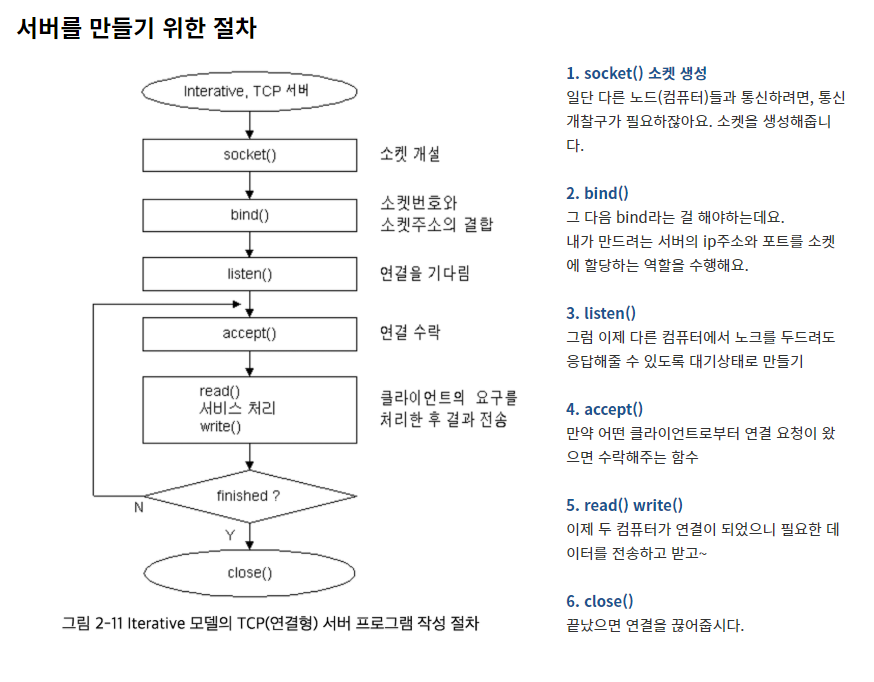
소켓 프로그래밍 (C언어)

서버, 클라이언트, 소켓 관계도

* 소켓 socket이란???
* 소프트웨어적인 데이터 송수신 방법을 이미 운영체제에서 제공해주고 있는데 이게‘소켓(Socket)’입니다. 모든 운영체제에서 지원해준다. 이는 물리적으로 연결된 네트워크에서의 송수신에 사용할 수 있는 소프트웨어적인 장치를 의미한다.
* 데이터를 주고 받기 위해서는 소켓 디스크립터라는 파일시스템을 이용해야하고, 그것을 소켓이라고 한다. 이미 운영체제에서 제공해주고 있는 함수로 우리는 원하는 통신 프로그램을 소스로 짜서 실현시킬 수 있다.



**1. 소켓 생성 함수**

* EX> #include<sys/socket.h>
* int socket(int domain, int type, int protocol);
* int domain:
* 어떤 영역에서 통신할 것인지에 대한 영역을 지정한다.(protocol family 지원)
* 올 수 있는 값 : AF\_UNIX, IF\_INET, IF\_INET6 등
* AF\_UNIX는 프로세스끼리 통신할 때, IF\_INET은 IPv4, IF\_INET6은 IPv6를 의미한다.
* int type:
* 어떤 서비스 타입의 소켓을 생성할건지 적는다. 값의 내용은 아래와 같다.
* SOCK\_STREAM(TCP), SOCK\_DGRAM(UDP), SOCK\_RAW(Raw 방식 TCP나 UDP를 거치지 않고 바로 IP계층 사용시) 우리는 TCP연결 지향형 통신을 생성하고자 하니 SOCK\_STREAM을 적어준다.
* int protocol:
* 소켓에서 사용할 프로토콜
* IPPROTO\_TCP : TCP 방식
* IPPROTO\_UDP : UDP 방식
* 0 : type에서 미리 정해진 경우.
* 리턴값 : 소켓을 가리키는 소켓 디스크립터를 반환합니다.

- -1 : 소켓 생성 실패

- 0이상의 값 : 디스크립터

* EX> TCP연결지향형이고 IPv4 도메인을 위한 소켓을 생성
* serv\_socket(PF\_INET, SOCK\_STREAM,0);
* if(serv\_sock == -1)
* printf(“socket error\n”);

**2. bind()**

* #include <sys/socket.h>
* int bind(int sockfd, struct sockaddr \*myaddr, socklen\_t addrlen);
* 소켓이랑 서버의 정보를 묶어주는 함수이다. 이 과정을 거치는 이유는 다른 외부의 컴퓨터가 서버에 연결하려고 요청을 하면 IP주소를 기반으로 찾지만 통신을 위해서는 소켓 디스크립터 번호를 알아야 한다. IP주소를 안다고 이것을 아는 것은 아니다 그래서 두 개를 묶어주는 작업을 해야한다.
* int sockfd:
* fd가 파일디스크립터의 약자이다. 즉 1번 함수 리턴값으로 받은 소켓 디스크립터를 이곳에 넣어주면 된다.
* struct sockaddr \*myaddr:
* 서버의 주소를 넣어준다.
* socklen\_t addrlen:
* 주소의 길이를 넣어준다.
* 리턴값: 성공시 0, 실패시 –1이다.
* EX> //소켓과 서버 주소를 바인딩
* if(bind(serv\_sock, (struct sockaddr\*) &serv\_addr, sizeof(serv\_addr)) == -1)
* printf(“bind error”);

**3. listen()**

* 어떤 컴퓨터로부터 요청이 와도 수락할 수 있게 대기 상태에 들어가는 함수이다.