

## 소켓통신

### 소켓(SOCKET)이란?

프로세스가 직접 데이터를 주고 받는 것이 아님.

소켓(Socket)은 프로세스가 드넓은 네트워크 세계로 데이터를 보내거나 혹은 그 세계로부터 데이터를 받기 위한 실제적인 창구 역할을 한다. 그러므로 프로세스가 데이터를 보내거나 받기 위해서는 반드시 소켓을 열어서 소켓에 데이터를 써보내거나 소켓으로부터 데이터를 읽어들여야 한다.

소켓은 프로토콜, IP 주소, 포트 번호로 정의된다.

#### \* 프로토콜

프로토콜은 원래 외교상의 언어로써 의례나 국가간에 약속을 의미하며, 통신에서는 어떤 시스템이 다른 시스템과 통신을 원활하게 수용하도록 해주는 통신 규약, 약속

#### \* IP

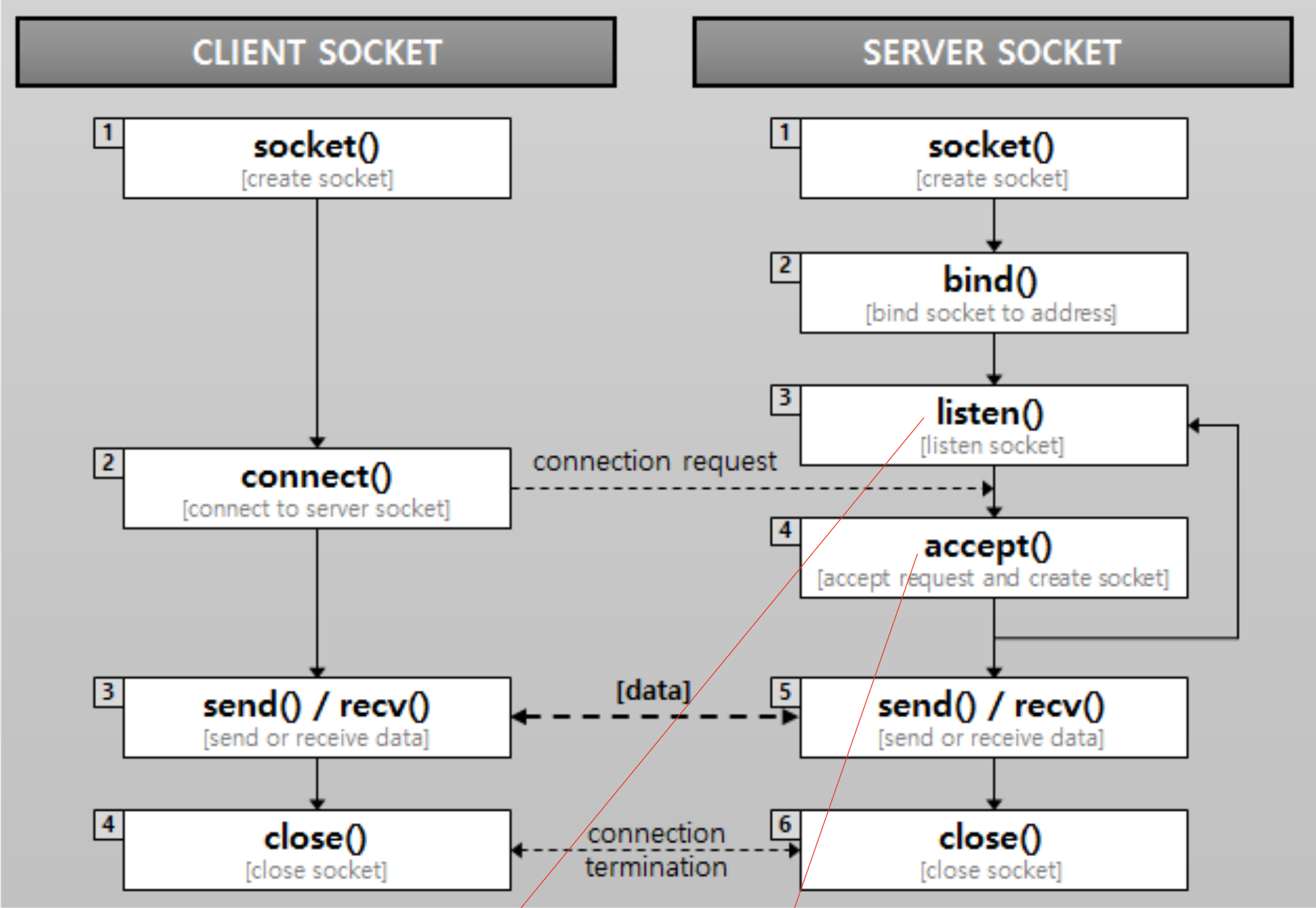
전 세계 컴퓨터에 부여된 고유의 식별 주소

#### \* 포트

포트(Port)는 네트워크 상에서 통신하기 위해서 호스트 내부적으로 프로세스가 할당받아야 하는 고유한 숫자이다. 한 호스트 내에서 네트워크 통신을 하고 있는 프로세스를 식별하기 위해 사용되는 값이므로, 같은 호스트 내에서 서로 다른 프로세스가 같은 포트 번호를 가질 수 없다. 즉, 같은 컴퓨터 내에서 프로그램을 식별하는 번호이다.

다시말해 소켓이 연결되어 있는 두 호스트를 연결해주는 도구로서 인터페이스의 역할을 하는데 데이터를 주고 받을 수 있는 구조체로 소켓을 통해 데이터 통로가 만들어 진다. 이러한 소켓은 역할에 따라 <sup>①</sup>서버 소켓, <sup>②</sup>클라이언트 소켓으로 구분된다.

소켓통신의 흐름



서버 (Server)

클라이언트 소켓의 연결 요청을 대기하고, 연결 요청이 오면 클라이언트 소켓을 생성하여 통신이 가능하게 한다

- 1) `socket()` 함수를 이용하여 <sup>서버 소켓</sup> 소켓을 생성
- 2) `bind()` 함수로 <sup>서버 주소와 포트번호</sup> ip와 port 번호를 설정하게 됩니다.
- 3) `listen()` 함수로 클라이언트의 접근 요청에 수신 대기열을 만들어 몇 개의 클라이언트를 대기 시킬지 결정
- 4) `accept()` 함수를 사용하여 클라이언트와의 연결을 기다림

클라이언트 (Client)

실제로 데이터 송수신이 일어나는 것은 클라이언트 소켓이다.

- 1) `socket()` 함수로 가장먼저 소켓을 연다.
- 2) `connect()` 함수를 이용하여 통신 할 서버의 설정된 ip와 port 번호에 통신을 시도
- 3) 통신을 시도 시, 서버가 `accept()` 함수를 이용하여 클라이언트의 socket descriptor를 반환
- 4) 이를 통해 클라이언트와 서버가 서로 `read()`, `write()` 를 하며 통신 (이 과정이 반복)