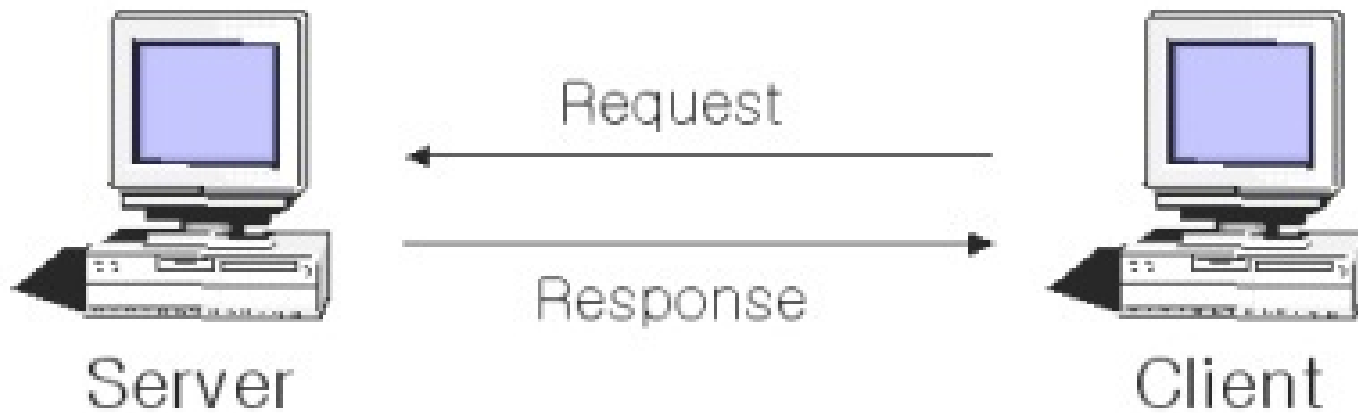


소켓 프로그래밍 1

컴퓨터 통신 실습
홍석준

Client/Server 모델

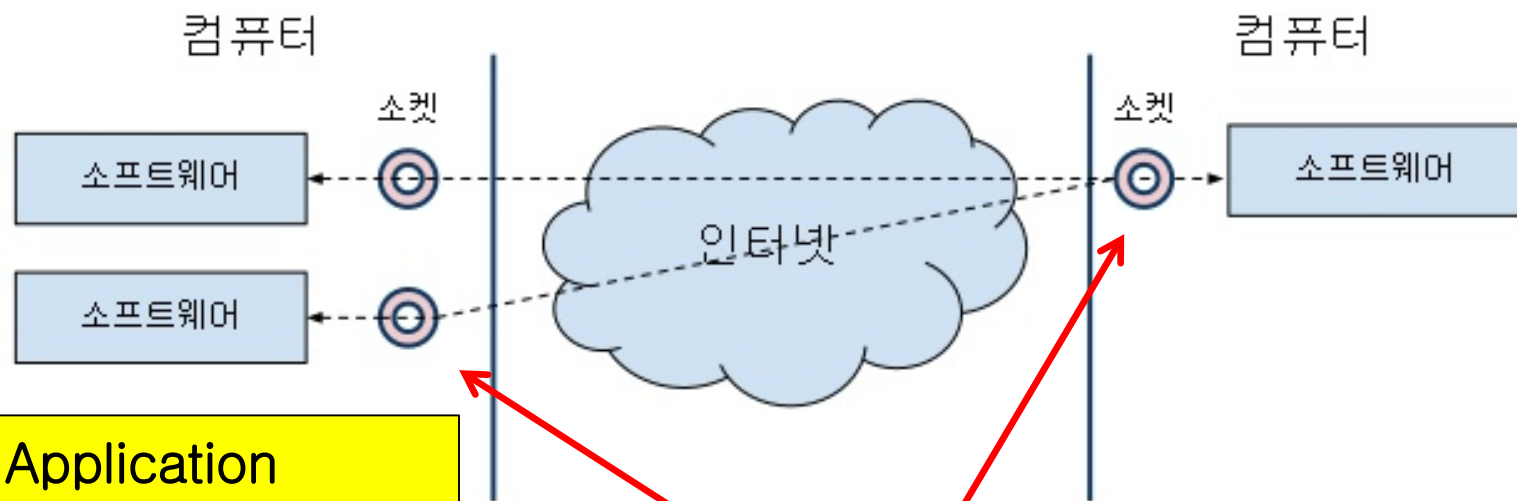
1. Server와 Client는 프로그램이다. Machine이 아니다!
2. Server는 Client의 연결요청을 기다린다.
3. Client는 Server에 요청을 하고 응답을 기다리는 호스트를 의미한다.



소켓(socket)의 개념

- 1982년 BSD(Berkeley Software Distribution)에서 소개
- 프로그래밍 인터페이스를 제공
- 소켓은 통신의 종단점이며, 메시지를 주고 받기 위해 프로세스가 사용할 수 있는 추상적인 개체
- 각 운영체제에서 제공 (Linux, 윈도우 모두 제공하고 있음)
- BSD socket은 C언어를 위해 개발되었으나, 그 인터페이스가 표준으로 사용되고 있는고로 다른 언어들도 C언어의 인터페이스를 비슷하게 따르고 있음
- BSD socket은 internet 영역에서 뿐만 아니라, 동일한 system에서의 IPC(프로세스간 내부 통신) 인터페이스도 제공하고 있음

소켓(socket)의 개념



Application
(messenger, FTP, etc ..)

소켓은 떨어져 있는 두 개체를 연결시켜주는 도구
(소프트웨어적인 장치)



네트워크 프로그래밍

- 네트워크 프로그래밍
 - 네트워크로 연결되어 있는 호스트들이 서로 데이터를 주고 받을 수 있도록 프로그램을 구현하는 것
 - 파일 입출력과 차이점은 데이터를 주고 받는 대상에 있다
- 소켓(socket)
 - 소프트웨어 차원에서 호스트들간에 연결해주는 매개체 기능을 하는 장치
 - 운영체제에서 제공 (리눅스, 윈도우 소켓)
- 소켓(socket)이라는 장치를 사용하여 프로그래밍
 - 일반적으로 소켓 프로그래밍과 네트워크 프로그래밍은 같은 의미로 사용

서버 소켓 구현의 이해

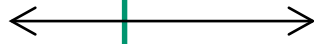
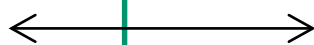
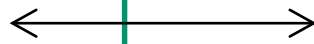
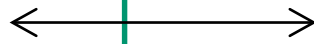
Telephone

1. 전화기 구입

2. 전화번호 할당

3. 케이블 연결

4. 수화기 든다.



Server Socket

1. 소켓 생성

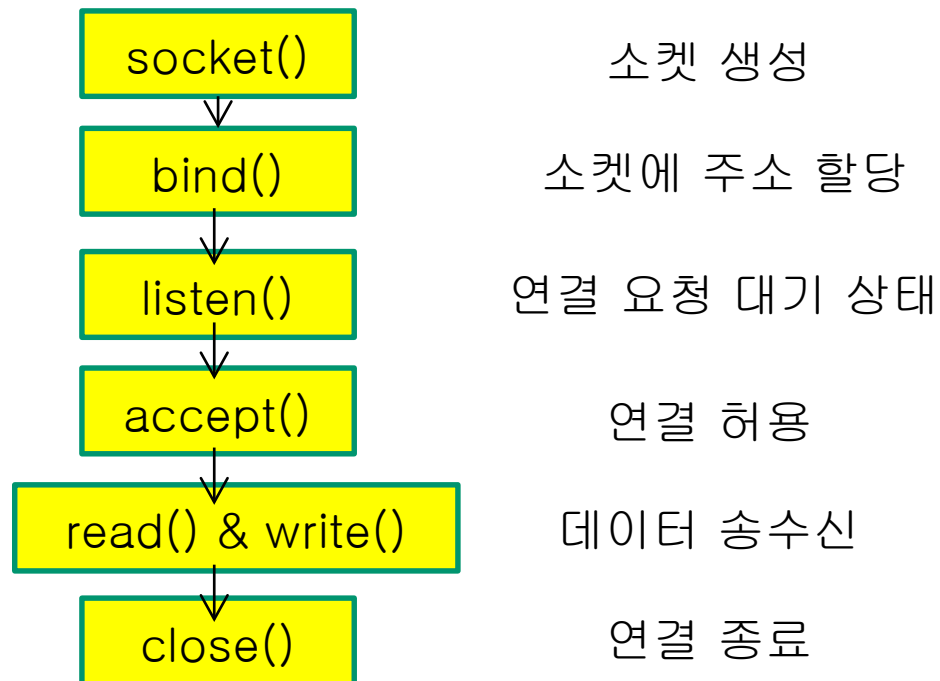
2. IP 주소와 포트 주소 할당

3. 연결 요청 대기 상태

4. 연결 수락

TCP 기반 서버의 구현

- 서버 구현을 위한 기본적인 함수 호출 순서



Server 프로그램 구현 1- 소켓의 생성

1. 소켓은 기본적인 통신의 도구이다.
2. 프로토콜에 독립적이다.
 - 소켓을 생성할 때 환경을 고려하여 프로토콜 체계를 지정해주면 그 환경에서 사용 가능한 소켓이 생성됨
3. 데이터 전송 타입을 지정해줘야 한다.
 - 연결 지향 소켓, 비연결 지향 소켓.

```
#include <sys/types.h>
#include <sys/socket.h>
```

```
Int socket(int domain, int type, int protocol)
```

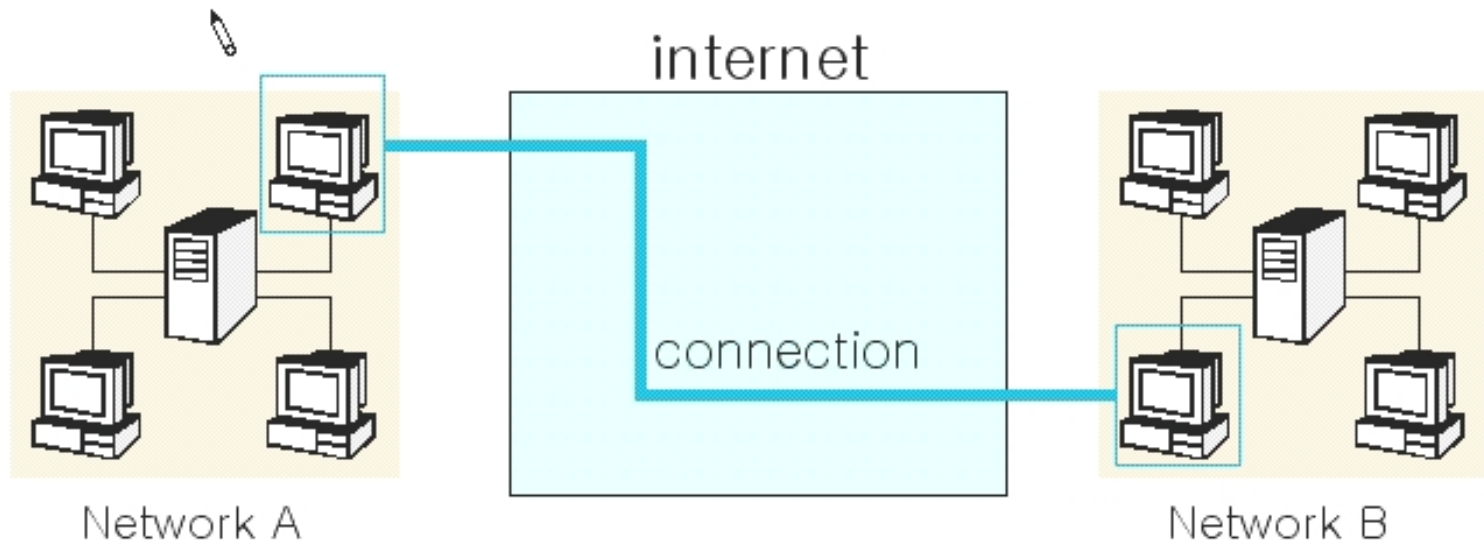

프로토콜 체계(domain 파라미터)

프로토콜 체계	정 의
PF_INET	IPv4 인터넷 프로토콜
PF_INET6	IPv6 인터넷 프로토콜
PF_LOCAL	Local 통신을 위한 UNIX 프로토콜
PF_PACKET	Low level socket을 위한 인터페이스
PF_IPX	IPX 노벨 프로토콜

소켓의 타입

1. 연결 지향형 소켓(SOCK_STREAM, TCP 소켓)

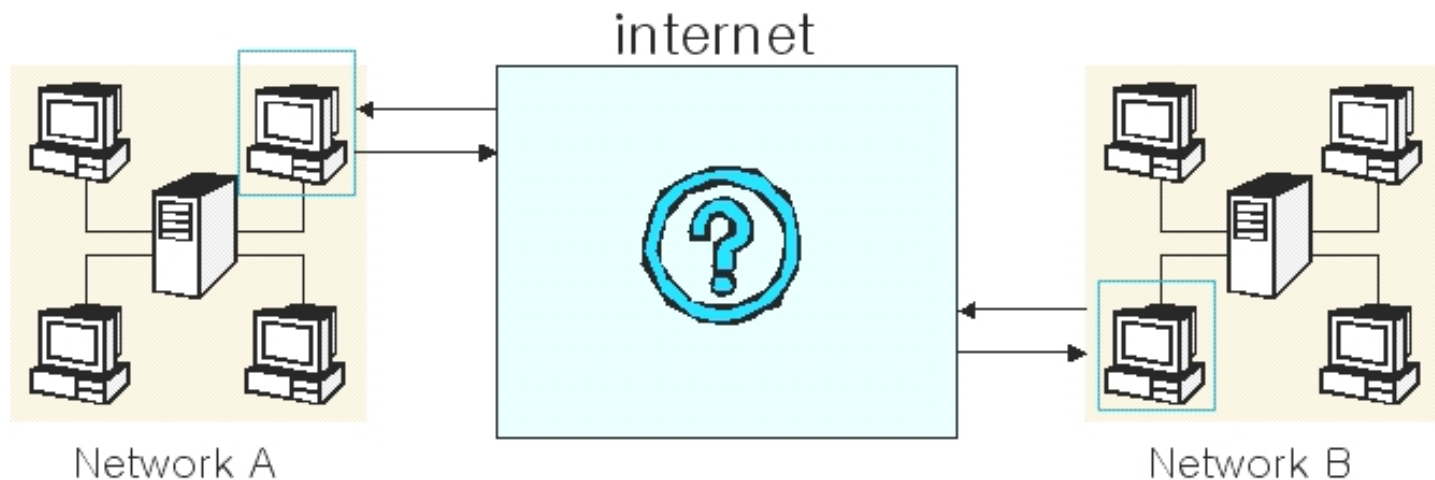
- 전송하는 순서대로 데이터가 전달된다.
- 에러나 데이터의 손실 없이 무사히 전달된다.
- 전송되는 데이터의 경계가 존재하지 않는다.



소켓의 타입


2. 비 연결 지향형 소켓(SOCK_DGRAM, UDP 소켓)

- 전송되는 순서에 상관없이 가장 빠른 전송을 지향한다.
- 전송되는 데이터는 손실 될 수도 있고, 에러가 발생할 수도 있다.
- 전송되는 데이터의 경계가 존재한다.
- 한번에 전송되는 데이터의 크기는 제한된다.



프로토콜의 선택

CASE	Domain	Type	Protocol
1	PF_INET	SOCK_STREAM	IPPROTO_TCP(0)
2	PF_INET	SOCK_DGRAM	IPPROTO_UDP(0)

< 표 2-2 > 

Server 프로그램 구현 2- 주소 정보 할당

- IP주소와 Port주소를 할당한다.

```
#include <sys/socket.h>
```

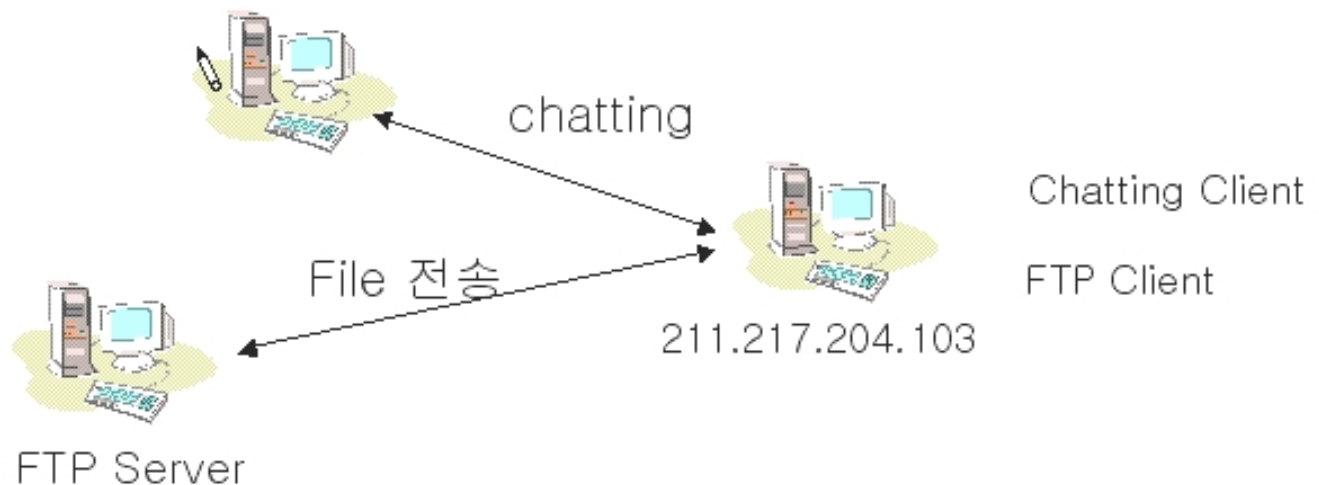
```
int bind(int sockfd, struct addr * myaddr, int addrlen)
```

Internet Address

- 인터넷 상에 존재하는 호스트를 구분하기 위한 32비트 주소체계
- 일반적으로 점이 찍힌 십진수 표현 방식을 사용해서 IP주소를 표현
- 점에 의해 구분되는 각각의 십진수값은 1바이트로 표현
 - 따라서, 총 4바이트를 사용

Port

1. 호스트 내에서 실행되고 있는 프로세스(Process)를 구분 짓기 위한 16 비트의 논리적 할당(소켓에 할당된다).
- 16비트 이므로 값의 범위가 0 ~ 65536 이다.
2. Well-known ports : 0 ~ 1023



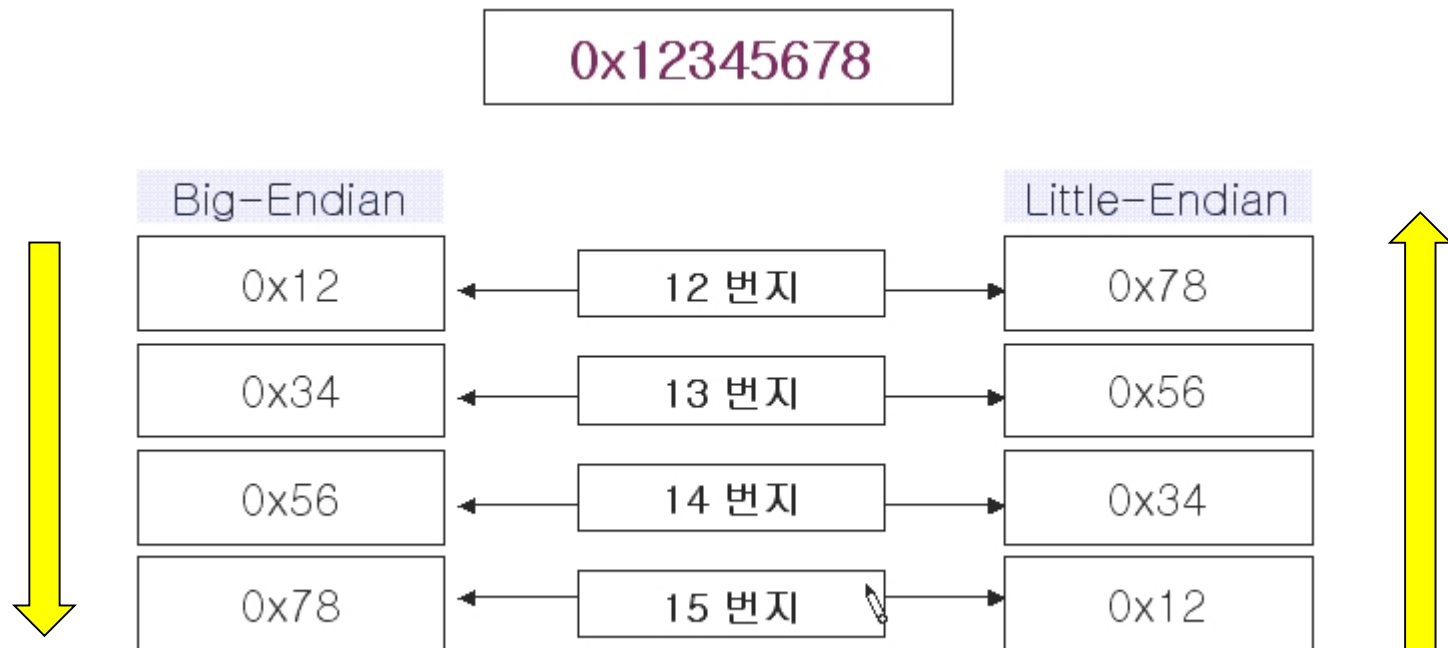
IPv4의 주소체계를 나타내는 구조체

```
struct sockaddr_in {
    sa_family_t      sin_family; /* 주소 체계(AF_INET) */
    uint16_t         sin_port;   /* port 정보 */
    struct in_addr    sin_addr;  /* 32 비트 IP 주소정보 */
    char             sin_zero[8]; /* 사용되지 않음 */
};
```

```
struct in_addr {
    uint32_t         s_addr;     /* 32비트 IP 주소정보 */
};
```

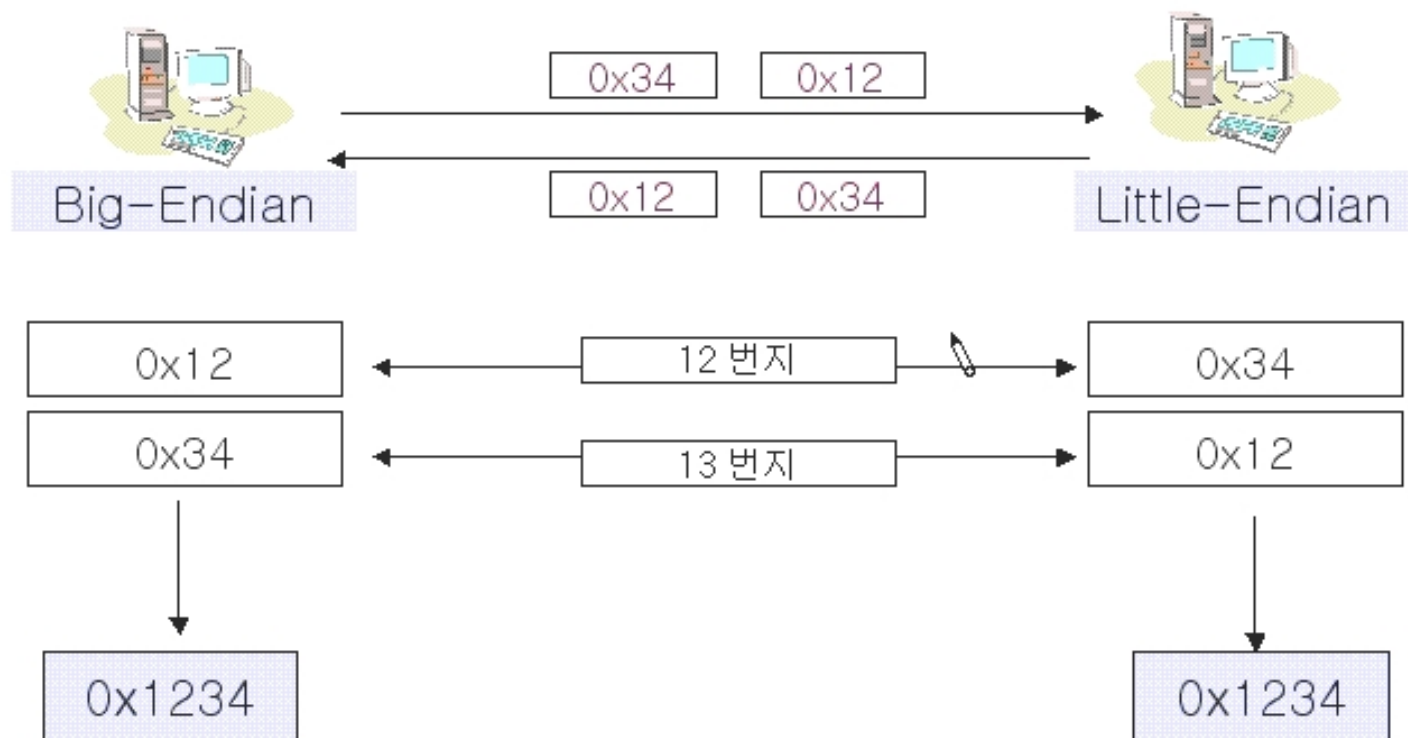
☞ 주의 : 모든 데이터는 네트워크 바이트 순서로 저장 해야 한다.

호스트 바이트 순서



네트워크 바이트 순서

1. 네트워크 바이트 순서는 Big-Endian 방식을 적용하기로 약속



바이트 순서 변환 함수

```
unsigned short htons(unsigned short);  
unsigned short ntohs(unsigned short);  
unsigned long htonl(unsigned long);  
unsigned long ntohl(unsigned long);
```

‘h’ : host byte order

‘n’ : network byte order

‘s’ : short (16 bit)

‘l’ : long (32 bit)

인터넷 주소 초기화 Template

```
1: struct sockaddr_in addr;  
2: char *serv_ip="211.217.168.13";  
3: char *serv_port="9190";  
  
4: memset(&addr, 0, sizeof(addr_len));  
5: addr.sin_family = AF_INET;  
6: addr.sin_addr.s_addr = inet_addr(serv_ip);  
7: addr.sin_port = htons(atoi(serv_port));
```

Server 프로그램 구현 3 - 연결 요청 대기

1. listen 함수는 전달되는 인자의 소켓을 ‘서버 소켓’이 되게 한다.
2. listen 함수는 ‘연결 요청 대기 큐’를 생성 한다.

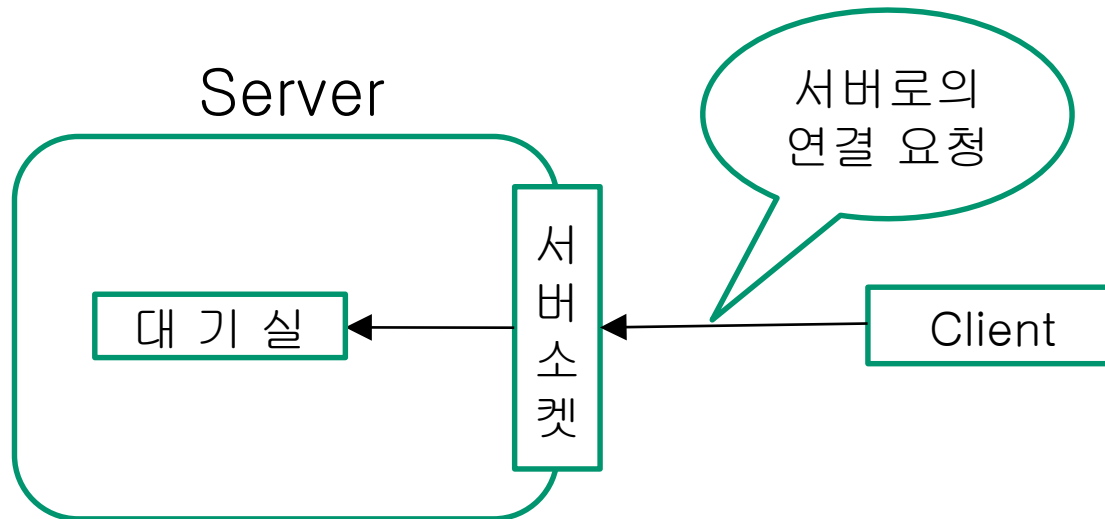
```
#include <sys/types.h>

int listen(int s, int backlog);
```

<리눅스>

서버의 역할과 연결 요청 대기 상태

- 서버 소켓은 일종의 문지기 역할



Server 프로그램 구현 4 - 연결 요청 수락

1. 연결요청 대기 큐(queue)에 존재하는 클라이언트의 연결 요청 수락.

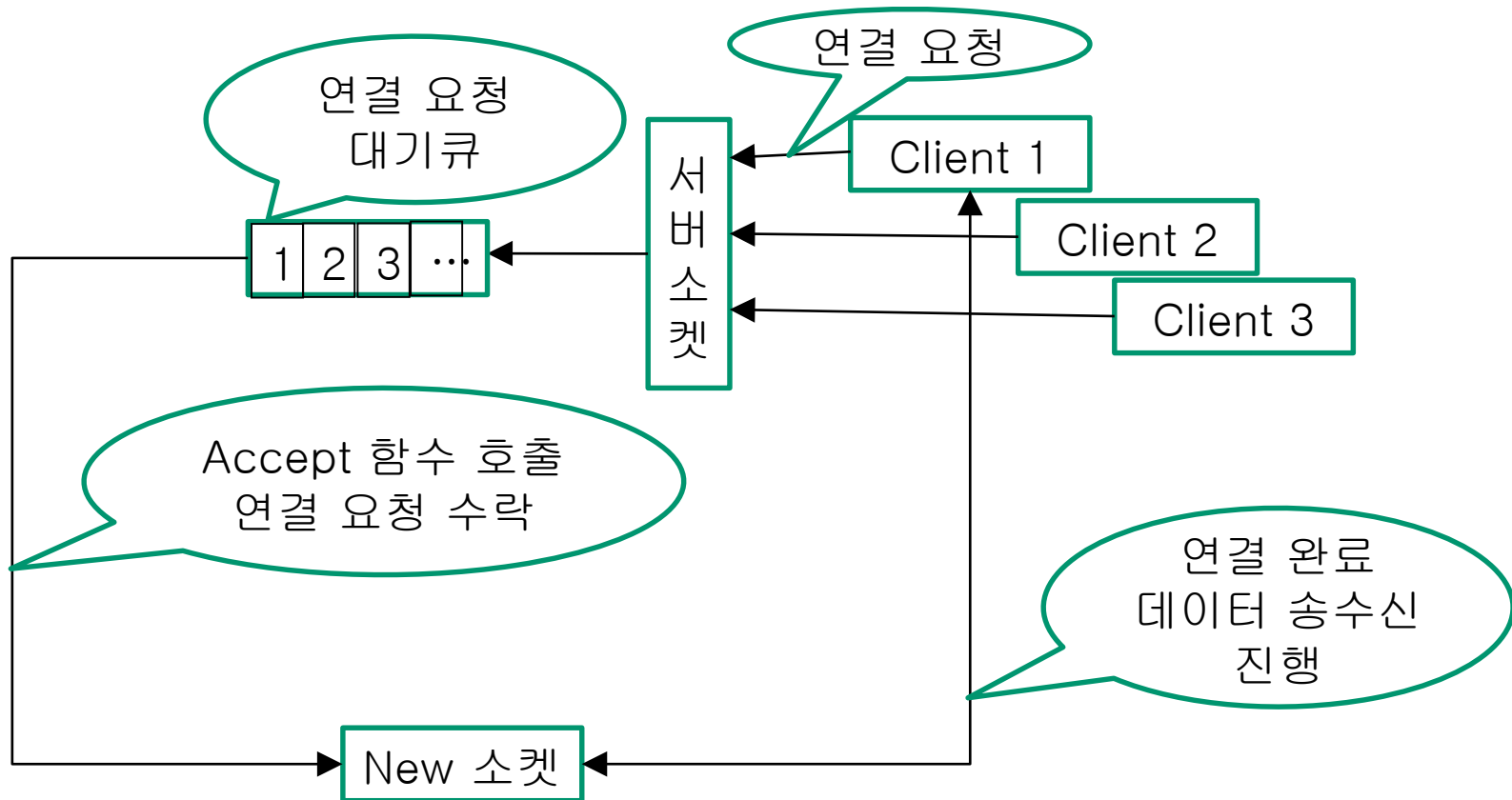
```
#include <sys/types.h>
#include <sys/socket.h>
int accept(int s, struct sockaddr * addr, int * addrlen);
```



<리눅스>

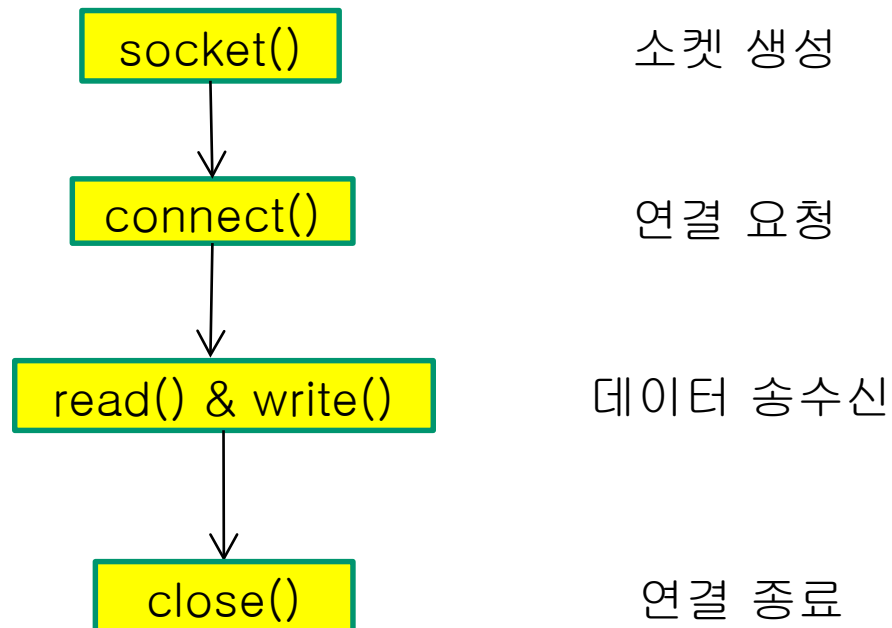
Client와 통신하기 위한 새로운 소켓의 fd가 리턴됨.

서버의 연결 요청 수락



TCP 기반 클라이언트의 구현

- 클라이언트 구현을 위한 기본적인 함수 호출 순서



Client 소켓 구현

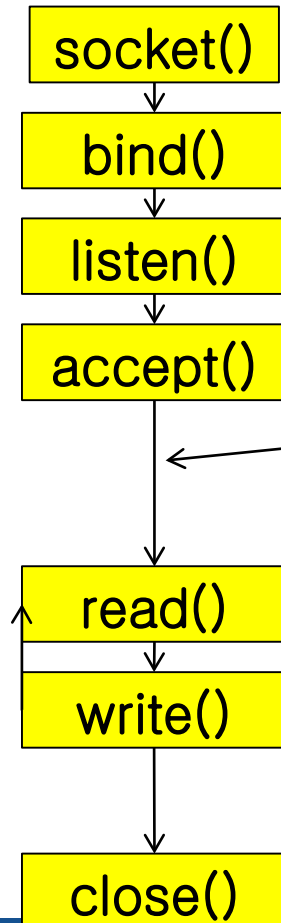
- 순서 1. 소켓의 생성
 - Socket함수 사용
- 순서 2. 연결 요청
 - 소켓과 목적지 주소에 대한 정보를 마련해 두고 나서 연결 요청을 시도함

```
#include <sys/socket.h>
```

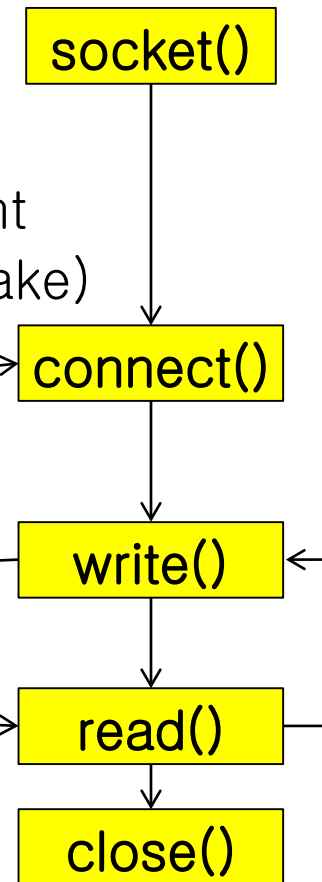
```
Int connect(int sockfd, struct sockaddr *addr, int *addrlen)
```

Socket Functions for simple TCP socket Program

TCP Server



TCP Client



Connection establishment
(TCP three-way handshake)

Data (request)

Data (reply)