

Internet Address Conversion Functions



NETWORKING LAB

Department of Computer Engineering

Kyung Hee University.

Choong Seon Hong

포트번호

□ IP 주소

- ◆ IP 데이터그램을 목적지 호스트까지 전달하는데 사용
- ◆ 특정 호스트를 찾는데 사용

□ 포트번호

- ◆ 16 비트로 표현 (0~65535)
- ◆ IP 데이터그램에 실린 데이터를 최종적으로 전달할 프로세스를 구분
- ◆ 호스트 내의 통신 접속점 (소켓)을 구분하는데 사용
- ◆ 같은 포트번호를 TCP와 UDP가 동시에 사용 가능

□ Well-known 포트

- ◆ 1023번 이하가 배정되어 사용됨
- ◆ 널리 사용되는 서비스를 위해 미리 지정되어 있는 포트번호
- ◆ 예) ftp, telnet, mail, http 등

□ /etc/services

- ◆ TCP/IP가 지원하는 응용 서비스와 포트번호가 정리된 파일

포트번호 확인

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

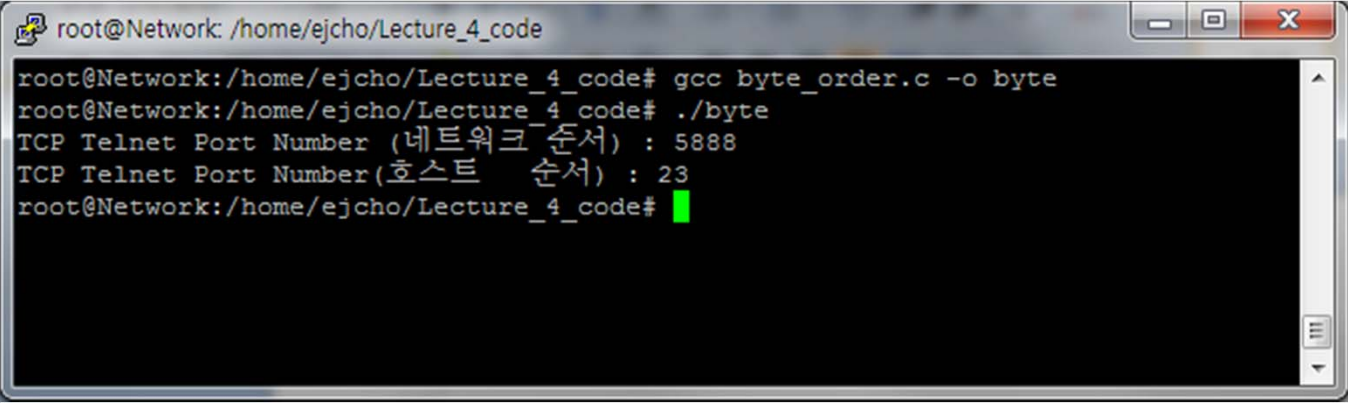
```
struct servent * getservbyname (string service, string protocol)
```

- ❑ getservbyname()은 well-known 포트를 사용하는 TCP/IP 응용 프로그램의 포트번호를 알아내는 함수이다.
- ❑ 서비스 이름과 프로토콜을 인자로 하여 호출하면 서비스 관련 각종 정보를 포함하고 있는 servent라는 구조체의 포인터를 리턴한다.

```
struct servent { char *s_name;      // 서비스 이름
                 char **s_aliases; // 별명 목록
                 int s_port;        // 포트번호
                 char *s_proto;     // 사용하는 프로토콜
};
```

프로그램 예제 확인

□ byte_order.c



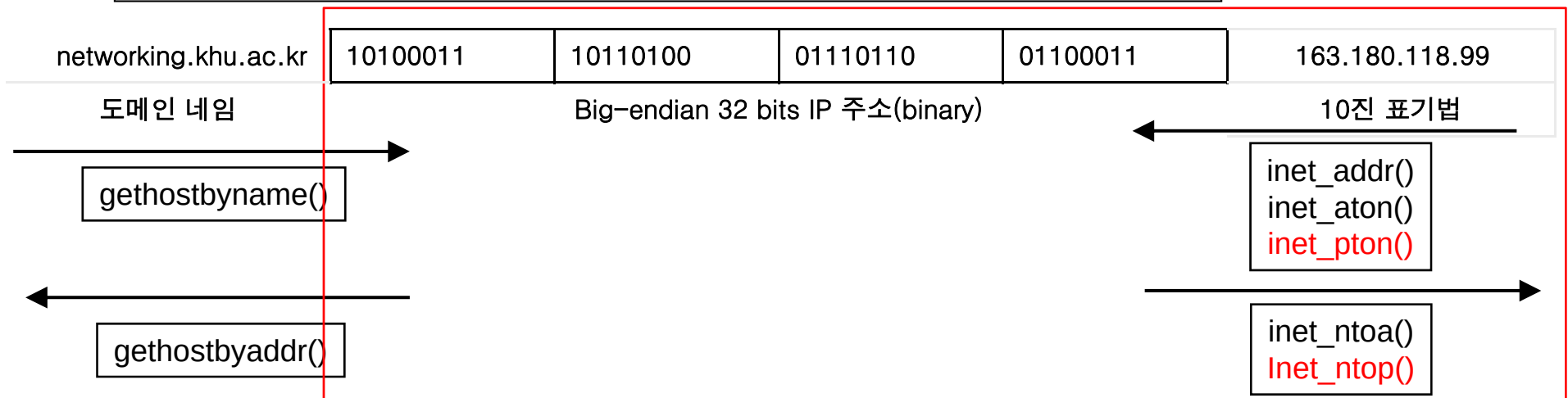
```
root@Network: /home/ejcho/Lecture_4_code
root@Network:/home/ejcho/Lecture_4_code# gcc byte_order.c -o byte
root@Network:/home/ejcho/Lecture_4_code# ./byte
TCP Telnet Port Number (네트워크 순서) : 5888
TCP Telnet Port Number (호스트 순서) : 23
root@Network:/home/ejcho/Lecture_4_code#
```

IP 주소 변환

- IP 주소를 도메인 네임과 dotted decimal 방식으로 표현
 - Dotted decimal은 15개의 문자로 구성된 스트링 변수
- 주소 표현법의 상호 변환 함수

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

unsigned long inet_addr(const char *string);
int inet_aton(const char *string, struct in_addr *addr);
char *inet_ntoa(struct in_addr addr);
int inet_pton(int family, const char *strptr, void *addrptr);
const char *inet_ntop(int family, const void *addrptr, char *strptr, size_t len);
```

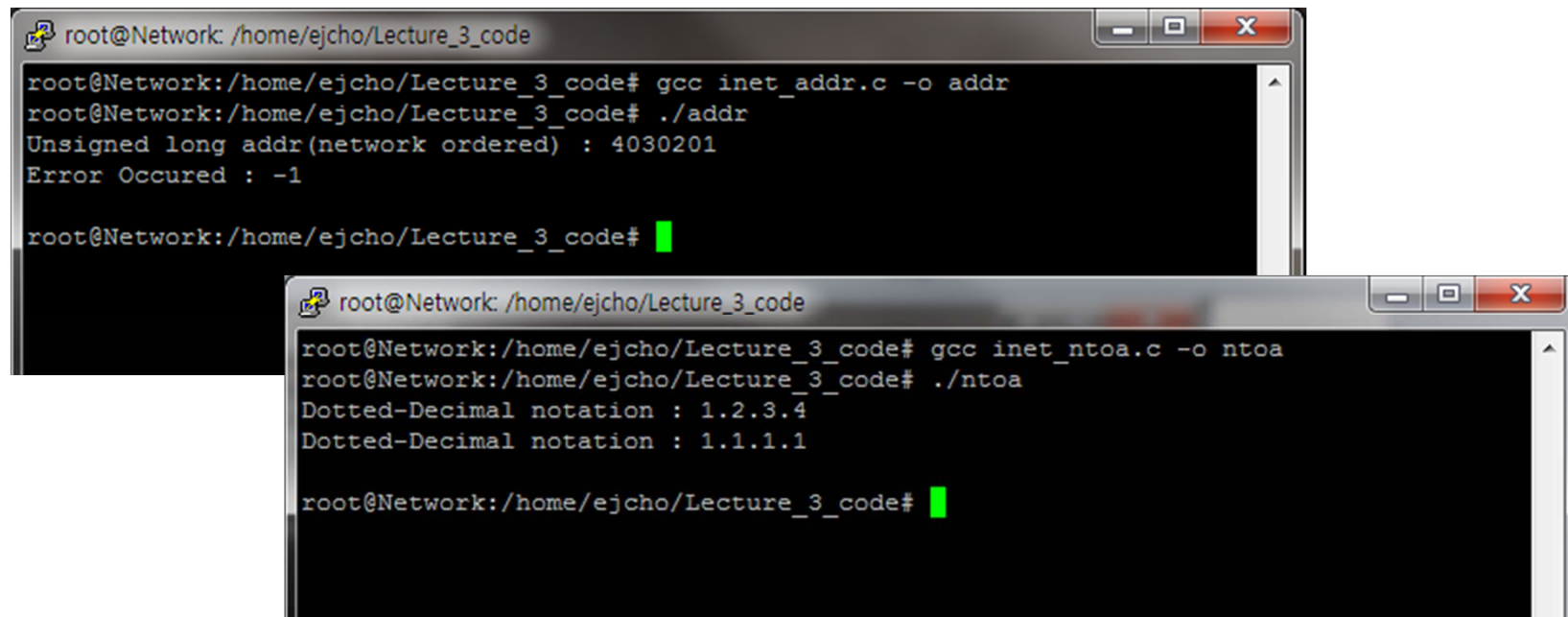


프로그램 예제 확인

□ 프로그램 예제

- ◆ inet_addr.c, inet_aton.c, inet_ntoa.c

□ 2. 실행결과.



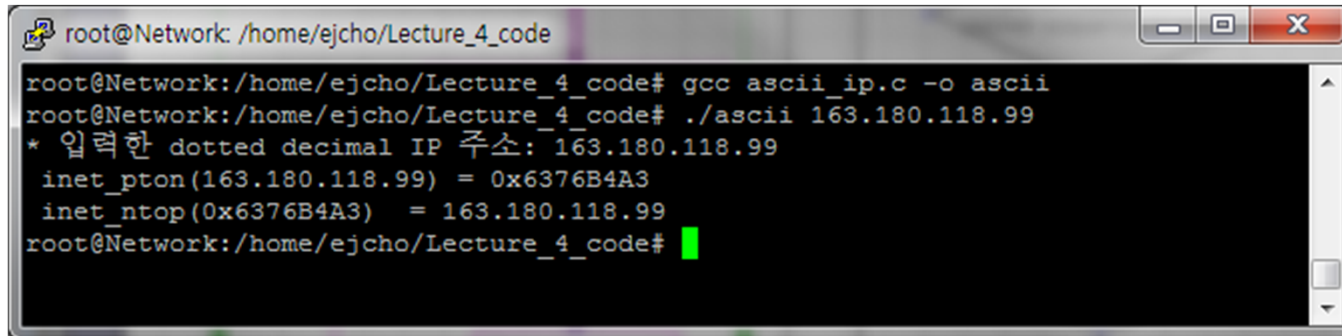
```
root@Network: /home/ejcho/Lecture_3_code
root@Network:/home/ejcho/Lecture_3_code# gcc inet_addr.c -o addr
root@Network:/home/ejcho/Lecture_3_code# ./addr
Unsigned long addr(network ordered) : 4030201
Error Occured : -1
root@Network:/home/ejcho/Lecture_3_code#
```

```
root@Network: /home/ejcho/Lecture_3_code
root@Network:/home/ejcho/Lecture_3_code# gcc inet_ntoa.c -o ntoa
root@Network:/home/ejcho/Lecture_3_code# ./ntoa
Dotted-Decimal notation : 1.2.3.4
Dotted-Decimal notation : 1.1.1.1
root@Network:/home/ejcho/Lecture_3_code#
```

프로그램 예제 확인

❑ ascii_ip.c

- ◆ Dotted decimal 표현의 주소를 4바이트의 IP 주소로 출력
- ◆ 변환된 4바이트의 IP를 dotted decimal 표현으로 출력



```
root@Network: /home/ejcho/Lecture_4_code
root@Network:/home/ejcho/Lecture_4_code# gcc ascii_ip.c -o ascii
root@Network:/home/ejcho/Lecture_4_code# ./ascii 163.180.118.99
* 입력한 dotted decimal IP 주소: 163.180.118.99
inet_pton(163.180.118.99) = 0x6376B4A3
inet_ntop(0x6376B4A3) = 163.180.118.99
root@Network:/home/ejcho/Lecture_4_code#
```

도메인 주소변환

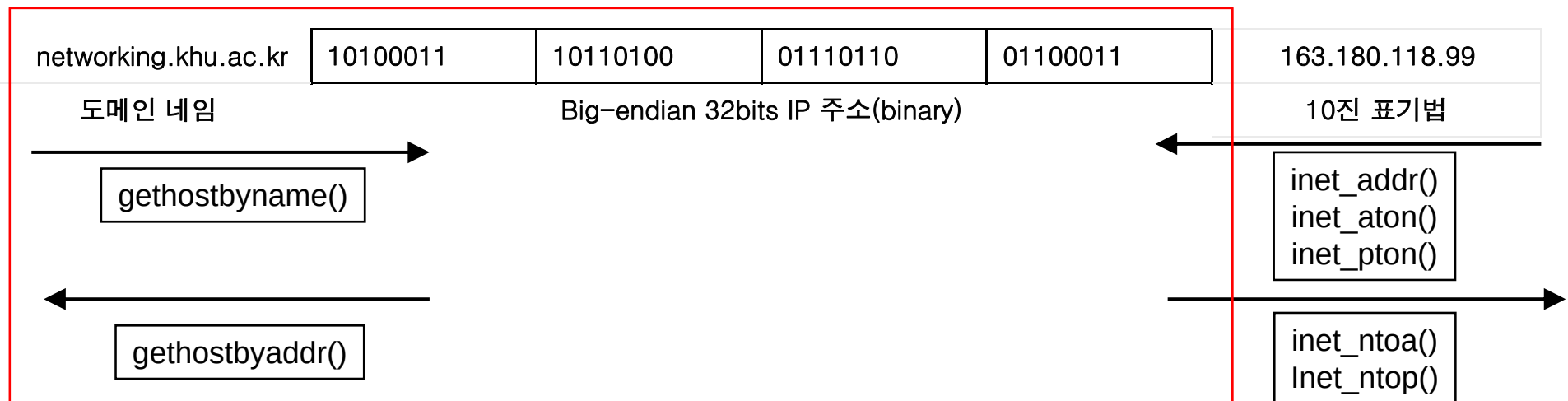
❑ DNS (Domain Name Service)

- ◆ 도메인 네임으로부터 IP 주소를 반환
- ◆ IP 주소로부터 도메인 네임을 반환

❑ 도메인 주소 변환 함수

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

struct hostent *gethostbyname(const char *hname);
struct hostent *gethostbyaddr(const char *in_addr, int len, int family);
```



도메인 주소변환

❑ gethostbyname()

- ◆ hname에 해당하는 호스트의 정보를 hostent 구조체 포인터로 반환

❑ gethostbyaddr()

- ◆ in_addr, 길이, 주소 타입으로부터 hostent 구조체 포인터 반환

```
struct hostent { char *h_name;           // 호스트 이름
                 char **h_aliases;      // 호스트 별칭
                 int h_addrtype;        // 호스트 주소의 종류
                 int h_length;          // 주소의 크기
                 char **h_addr_list;    // IP 주소 리스트
};
#define h_addr haddr_list[0];          // 첫 번째(대표) 주소
```

Member	Description
h_name	호스트의 공식이름
h_aliases	호스트의 별명을 배열의 포인터로 반환
h_addrtype	주소형태 (IPv4:AF_INET IPv6:AF_INET6)
h_length	주소길이 (IPv4:4 IPv6:16)
h_addr_list	IP주소를 배열의 포인터로 반환 (IPv4:각 포인터는 4바이트 IP주소를 가리킨다.)

프로그램 예제 확인

❑ get_hostent.c , get_host_byaddr.c

```
root@Network: /home/ejcho/Lecture_4_code
root@Network:/home/ejcho/Lecture_4_code# gcc get_hostent.c -o host
root@Network:/home/ejcho/Lecture_4_code# ./host www.daum.net
호스트 이름      : daumtop.daum.akadns.net
호스트 주소타입 번호: 2
호스트 주소의 길이 : 4 바이트
IP 주소 (1 번째) : 211.115.77.213
IP 주소 (2 번째) : 211.115.77.214
IP 주소 (3 번째) : 211.115.115.211
IP 주소 (4 번째) : 211.115.115.212
IP 주소 (5 번째) : 222.231.51.77
IP 주소 (6 번째) : 222.231.51.78
IP 주소 (7 번째) : 211.32.117.30
IP 주소 (8 번째) : 211.115.77.212
호스트 별명 (1 번째) : www.daum.net
root@Network:/home/ejcho/Lecture_4_code#
```

```
root@Network: /home/ejcho/Lecture_4_code
root@Network:/home/ejcho/Lecture_4_code# gcc get_host_byaddr.c -o hostaddr
root@Network:/home/ejcho/Lecture_4_code# ./hostaddr 163.180.118.99
호스트 이름 : corba.kyunghee.ac.kr.180.163.in-addr.arpa
root@Network:/home/ejcho/Lecture_4_code#
```

IPv4의 주소체계를 나타내는 구조체

□ 클라이언트 또는 서버의 구체적인 주소를 표현을 위해 사용

- 주소체계
- IP 주소
- 포트번호

```
struct sockaddr_in {  
    sa_family_t    sin_family;    // 주소 체계  
    uint16_t       sin_port;      // 16비트 포트번호  
    struct in_addr sin_addr;      // 32 비트 IP 주소  
    char           sin_zero[8];   // 사용되지 않음  
};
```

```
struct in_addr {  
    uint32_t       s_addr;        // 32비트 IP 주소를 저장하는 구조체  
};
```

☞ 주의 : 모든 데이터는 네트워크 바이트 순서로 저장 해야 한다.

인터넷 주소 초기화 Template

```
1:  struct sockaddr_in addr;  
2:  char *serv_ip="163.180.118.99";  
3:  char *serv_port="9999";  
  
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));           //스트링 →정수
```

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

주소 정보 할당하기

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

int bind(int sockfd, struct sockaddr * myaddr, int addrlen);
```

- ◆ sockfd : 주소를 할당하고자 하는 소켓의 파일 디스크립터를 인자로 전달
- ◆ myaddr : 할당하고자 하는 주소 정보를 지니고 있는 sockaddr_in 구조체 변수의 포인터를 인자로 전달
- ◆ addrlen : 인자로 전달될 주소 정보 구조체의 길이를 전달
- ◆ 함수 호출이 성공하면 sockfd가 가리키는 소켓에 myaddr이 가리키는 주소 정보가 할당됨

소켓 주소 구조체

```
struct sockaddr {  
    sa_family_t    sin_family;    // address_family  
    char           sa_data[14];   // 주소  
};
```

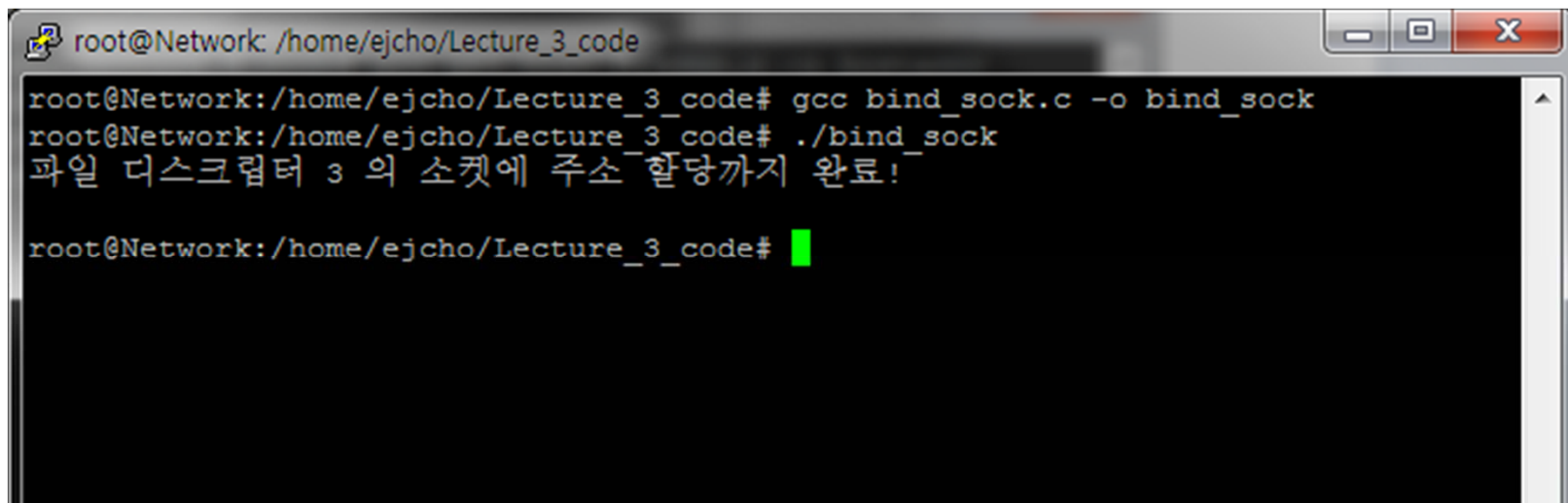
- ❑ 프로토콜 체계에 따라 주소 체계가 다름
- ❑ socket 함수와 bind 함수는 특정 프로토콜 체계를 위한 함수가 아님
 - ◆ 프로토콜에 독립적
- ❑ 따라서, 일반적으로 선언되어야 함
- ❑ Local Unix 프로토콜을 위한 sockaddr_un 구조체의 포인터도 인자값으로 사용할 수 있어야 함

프로그램 예제 확인

□ 프로그램 예제

- ◆ bind_sock.c

□ 실행결과.



```
root@Network: /home/ejcho/Lecture_3_code
root@Network:/home/ejcho/Lecture_3_code# gcc bind_sock.c -o bind_sock
root@Network:/home/ejcho/Lecture_3_code# ./bind_sock
파일 디스크립터 3 의 소켓에 주소 할당까지 완료!
root@Network:/home/ejcho/Lecture_3_code#
```