Systems Programming

Spring 2023

9 주차

- Q&A
  - 12-network-programming
  - 13-webservices

중간고사 2

- 범위
  - 06-vm-concepts ~ 13-webservices
  - 강의자료 및 Q&A Session Materials
  - 실습내용 (Malloc Lab)

< 중간고사 2 범위 >

- 6주차 (4/13 목)
  - 06-vm-concepts
  - 07-vm-systems
- 7주차 (4/20 목)
  - 08-allocation-basic
  - 09-allocation-advanced
- 8주차 (4/27 목)
  - 10-internet
  - 11-NAT-PAT
- 9주차 (5/4 목)
  - 12-network-programming
  - 13-webservices

- 5월 11일 목요일 오후 2시 ~

- 장소
  - 302동 105호
  - 302동 209호
  - 개인별 시험장소 배정 공지 예정 (eTL, ~5/7)

중간고사 2

- Q&A Session 시험 범위 가이드라인
  - 강의 내용에 대한 부연설명은 시험범위에 해당
  - 강의 내용의 advanced topic 에 대한 내용은 시험 범위에 해당되지 않음

- 예를 들면,
  - SP-session08.pdf 의 p.9~p.10 질문의 경우,
  - outside local 과 global 이 구분되어 NAT Table 에 저장되어야 하는 이유에 관한 것은 **시험 범위에 해당**
  - 이해를 돕기 위해 설명된 p.10 의 Router 에 대한 내용은 **시험 범위에 해당되지 않음**

중간고사 2

- 오후 1시 50분까지 착석 완료
  - 강의실 시계 기준 2시 이후 입실자는 감점 조치 취해질 예정

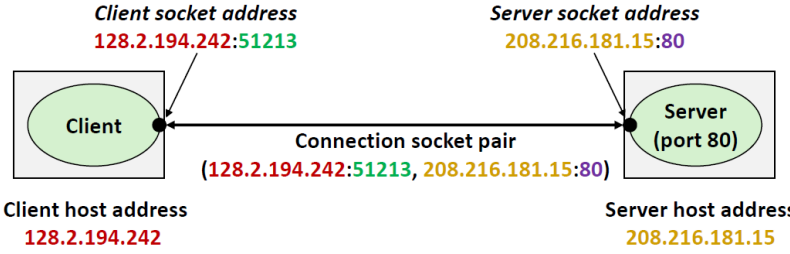
- 신분증 검사 진행
  - 실물 학생증 또는 신분증 지참
  - 모바일 학생증 인정하지 않음

- 11주차 ~ 15주차 강의자료 etl 업로드 예정
  - 각자 확인하고 학습 진행할 것

Questions

1. 12-network-programming 5p, 16p
사소한 질문이기는 한데, 일반적으로 socket address라고 하면 16p에 나온 것처럼 48bit 전체 struct를 말하는 것인지, 아니면 5p에 나와있는 IP address + port number를 말하는 것인지 궁금합니다.

Putting it all Together: Anatomy of an Internet Connection



51213 is an ephemeral port allocated by the kernel

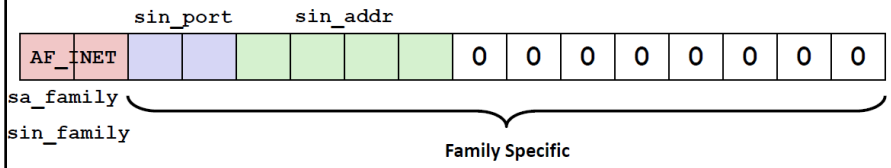
80 is a well-known port associated with Web servers

5

Socket Address Structures

- **Internet-specific socket address:**
  - Must cast (`sockaddr_in *`) to (`sockaddr *`) for `connect`, `bind`, and `accept`

```
struct sockaddr_in  {
  unsigned short  sin_family;  /* address family (always AF_INET) */
  unsigned short  sin_port;    /* port num in network byte order */
  struct in_addr  sin_addr;    /* IP addr in network byte order */
  unsigned char   sin_zero[8]; /* pad to sizeof(struct sockaddr) */
};
```

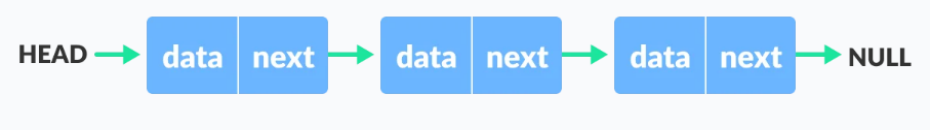


17

Questions

1. 12-network-programming 5p, 16p
사소한 질문이기는 한데, 일반적으로 socket address라고 하면 16p에 나온 것처럼 48bit 전체 struct를 말하는 것인지, 아니면 5p에 나와있는 IP address + port number를 말하는 것인지 궁금합니다.

- Consider Linked-list



Linked list Data Structure

```
struct node
{
  int data;
  struct node *next;
};
```

```
/* Initialize nodes */
struct node *head;
struct node *one = NULL;
struct node *two = NULL;
struct node *three = NULL;

/* Allocate memory */
one = malloc(sizeof(struct node));
two = malloc(sizeof(struct node));
three = malloc(sizeof(struct node));

/* Assign data values */
one->data = 1;
two->data = 2;
three->data=3;

/* Connect nodes */
one->next = two;
two->next = three;
three->next = NULL;

/* Save address of first node in head */
head = one;
```

Questions

2. 12-network-programming 20p, 26p
20p에 나온 것처럼 connect에서는 2번째 argument로 (SA *)serveraddr를 보내주면서 3번째 argument로 sizeof()를 이용해 크기를 함께 보내주었는데 26p에서 accept의 경우에는 똑같이 2번째 argument로 (SA*)clientaddr를 받아오지만 3번째 argument로 sizeof()의 결과인 clientlen이 아닌 clientlen의 주소값을 넣도록 되어있는데 connect에서처럼 크기를 값으로 넣어주면 안되는 이유가 있을지 궁금합니다.

man7.org > Linux > man-pages — Linux/UNIX system programming training

connect(2) — Linux manual page

NAME | SYNOPSIS | DESCRIPTION | RETURN VALUE | ERRORS | CONFORMING TO | NOTES | EXAMPLES | SEE ALSO | COLOPHON

Search online pages

CONNECT(2) Linux Programmer's Manual CONNECT(2)

NAME top

connect - initiate a connection on a socket

SYNOPSIS top

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

int connect(int sockfd, const struct sockaddr *addr,
            socklen_t addrlen);
```

DESCRIPTION top

The **connect**() system call connects the socket referred to by the file descriptor *sockfd* to the address specified by *addr*. The *addrlen* argument specifies the size of *addr*. The format of the address in *addr* is determined by the address space of the socket *sockfd*; see socket(2) for further details.

If the socket *sockfd* is of type SOCK_DGRAM, then *addr* is the address to which datagrams are sent by default, and the only address from which datagrams are received. If the socket is of type SOCK_STREAM or SOCK_SEQPACKET, this call attempts to make a connection to the socket that is bound to the address specified by *addr*.

Some protocol sockets (e.g., UNIX domain stream sockets) may

man7.org > Linux > man-pages — Linux/UNIX system programming training

accept(2) — Linux manual page

NAME | SYNOPSIS | DESCRIPTION | RETURN VALUE | ERRORS | VERSIONS | CONFORMING TO | NOTES | EXAMPLES | SEE ALSO | COLOPHON

Search online pages

ACCEPT(2) Linux Programmer's Manual ACCEPT(2)

NAME top

accept, accept4 - accept a connection on a socket

SYNOPSIS top

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

int accept(int sockfd, struct sockaddr *restrict addr,
           socklen_t *restrict addrlen);

#define _GNU_SOURCE             /* See feature_test_macros(7) */
#include <sys/socket.h>

int accept4(int sockfd, struct sockaddr *restrict addr,
            socklen_t *restrict addrlen, int flags);
```

DESCRIPTION top

The **accept**() system call is used with connection-based socket types (SOCK_STREAM, SOCK_SEQPACKET). It extracts the first connection request on the queue of pending connections for the listening socket, *sockfd*, creates a new connected socket, and returns a new file descriptor referring to that socket. The newly created socket is not in the listening state. The original socket *sockfd* is unaffected by this call.

The argument *sockfd* is a socket that has been created with

Questions

3. 12-network-programming 32p

sin_addr에 들어가는 INADDR_ANY의 의미에 대해 제대로 이해하지 못한 것 같아 질문드립니다. 강의에서 "서버는 자신의 주소를 알고 있지만 누구랑 연결될지는 모르기에 누구라도 연결될 수 있다라는 의미에서 INADDR_ANY를 넣어둔다" 라고 되어있는데 찾아보니 INADDR_ANY 값은 0.0.0.0이며 서버의 IP주소를 자동으로 찾아넣어서 사용하라는 의미라고 나오는데 설명이 좀 다른 것 같아 질문드립니다.

man7.org > Linux > man-pages — Linux/UNIX system programming training

ip(7) — Linux manual page

NAME | SYNOPSIS | DESCRIPTION | ERRORS | NOTES | BUGS | SEE ALSO | COLOPHON

Search online pages

IP(7) Linux Programmer's Manual IP(7)

NAME top

ip - Linux IPv4 protocol implementation

SYNOPSIS top

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/ip.h> /* superset of previous */

tcp_socket = socket(AF_INET, SOCK_STREAM, 0);
udp_socket = socket(AF_INET, SOCK_DGRAM, 0);
raw_socket = socket(AF_INET, SOCK_RAW, protocol);
```

DESCRIPTION top

Linux implements the Internet Protocol, version 4, described in RFC 791 and RFC 1122. **ip** contains a level 2 multicasting implementation conforming to RFC 1112. It also contains an IP router including a packet filter.

The programming interface is BSD-sockets compatible. For more information on sockets, see socket(7).

Address format

An IP socket address is defined as a combination of an IP interface address and a 16-bit port number. The basic IP protocol does not supply port numbers, they are implemented by higher level protocols like udp(7) and tcp(7). On raw sockets *sin_port* is set to the IP protocol.

```
struct sockaddr_in {
    sa_family_t    sin_family; /* address family: AF_INET */
    in_port_t      sin_port;   /* port in network byte order */
    struct in_addr sin_addr;   /* internet address */
};

/* Internet address */
struct in_addr {
    uint32_t       s_addr;     /* address in network byte order */
};
```

There are several special addresses: **INADDR_LOOPBACK** (127.0.0.1) always refers to the local host via the loopback device; **INADDR_ANY** (0.0.0.0) means any address for binding; **INADDR_BROADCAST** (255.255.255.255) means any host and has the same effect on bind as **INADDR_ANY** for historical reasons.

13-webservices