

컴퓨터 네트워크

# 12장. 소켓시스템 (2)

## - 시스템콜

# 이번 시간의 학습 목표

- ▶ 소켓 시스템 콜의 문법 구조와 사용법

# socket() 함수 (1)

- ▶ 소켓을 생성하며, 생성된 소켓의 디스크립터(descriptor)를 반환
- ▶ socket() 함수 사용법
  - ▶ 문법

```
# include <sys/types.h>
# include <sys/socket.h>
int socket(int domain, int type, int protocol);
```

- ▶ 설명
  - ▶ domain: 사용할 도메인을 지정
  - ▶ type: 서비스 유형을 지정
  - ▶ protocol: 보통 0으로 지정

# socket() 함수 (2)

## ▶ socket() 함수 예제

```
sd = socket(AF_UNIX, SOCK_STREAM, 0);  
/* 유닉스 도메인 연결형 서비스 */
```

```
sd = socket(AF_UNIX, SOCK_DGRAM, 0);  
/* 유닉스 도메인 비연결형 서비스 */
```

```
sd = socket(AF_INET, SOCK_STREAM, 0);  
/* 인터넷 도메인 연결형 서비스 */
```

```
sd = socket(AF_INET, SOCK_DGRAM, 0);  
/* 인터넷 도메인 비연결형 서비스 */
```

# bind() 함수 (1)

## ▶ 생성된 소켓에 주소를 부여

### ▶ bind() 함수 사용법

#### ▶ 문법

```
# include <sys/types.h>
# include <sys/socket.h>
int bind(int s, const struct sockaddr *name, socklen_t *namelen);
```

#### ▶ 설명

- ▶ s: socket() 함수가 리턴한 디스크립터
- ▶ name: 바인드할 소켓 주소를 표기
- ▶ namelen: name에 보관된 주소 공간의 크기

# bind() 함수 (2)

## ▶ AF\_UNIX 예제

```
int sd;  
struct sockaddr_un addr;
```

```
sd = socket(AF_UNIX, SOCK_STREAM, 0);  
if(sd == -1) {  
    perror("socket");  
    exit(1);  
}
```

```
addr.sun_family = AF_UNIX;  
strcpy(addr.sun_path, "/tmp/sock_addr");
```

```
if(bind(sd, (struct sockaddr *)&addr, sizeof(addr)) == -1) {  
    perror("bind");  
    exit(1);  
}
```

소켓 주소

# bind() 함수 (3)

## ▶ AF\_INET 예제

```
int sd;  
struct sockaddr_in addr;
```

```
sd = socket(AF_INET, SOCK_STREAM, 0);  
if(sd == -1) {  
    perror("socket");  
    exit(1);  
}
```

```
addr.sin_family = AF_INET;  
addr.sin_addr.s_addr = htonl(INADDR_ANY);  
addr.sin_port = htons(5010);
```

```
if(bind(sd, (struct sockaddr *)&addr, sizeof(addr)) == -1) {  
    perror("bind");  
    exit(1);  
}
```

소켓 주소

# 주소(정수) 변환

- ▶ 컴퓨터마다 정수형 데이터를 처리하는 방법이 다를 수 있음
- ▶ ‘개별 호스트 -> 네트워크’ 변환: htonl(), htons()
- ▶ ‘네트워크 -> 개별 호스트’ 변환: ntohl(), ntohs()
- ▶ 문법

```
#include <sys/types.h>
#include <netinet/in.h>
#include <inttypes.h>
uint32_t htonl(uint32_t hostlong);
uint16_t htons(uint16_t hostshort);
uint32_t ntohl(uint32_t netlong);
uint16_t ntohs(uint16_t netshort);
```

# listen() 함수 (1)

▶ 소켓에서 대기할 수 있는 연결 요청의 개수를 지정

▶ 사용법

▶ 문법

```
#include<sys/types.h>
#include<sys/socket.h>
int listen(int s, int backlog);
```

▶ 설명

- ▶ s: socket() 함수가 생성한 연결형 서비스용 소켓
- ▶ backlog: 일반적인 환경에서 5로 지정

# listen() 함수 (2)

## ▶ listen() 함수 예제

```
int sd;
struct sockaddr_in addr;

sd = socket(AF_INET, SOCK_STREAM, 0);
if(sd == -1) {
    perror("socket");
    exit(1);
}
addr.sin_family = AF_INET;
addr.sin_addr.s_addr = htonl(INADDR_ANY);
addr.sin_port = htons(5010);

if(bind(sd, (struct sockaddr *)&addr, sizeof(addr)) == -1) {
    perror("bind");
    exit(1);
}

if(listen(sd, 5) == -1) {
    perror("listen");
    exit(1);
}
```

# accept() 함수 (1)

- ▶ 서버 프로그램에서 클라이언트의 연결 요청을 대기
- ▶ 사용법
  - ▶ 문법

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

- ▶ 설명

- ▶ s: socket() 함수가 생성한 연결형 서비스용 소켓
- ▶ addr: 연결을 요청한 클라이언트의 소켓 주소를 반환

# accept() 함수 (2)

## ▶ accept() 함수 예제

```
int sd, new;
struct sockaddr_in addr;
struct sockaddr_in client;
int client_len;

sd = socket (AF_INET, SOCK_STREAM, 0);
addr.sin_family = AF_INET;
addr.sin_addr.s_addr = htonl (INADDR_ANY);
addr.sin_port = htons (5010);
bind (sd, (struct sockaddr *)&addr, sizeof (addr));
listen (sd, 5);

while ((new = accept (sd, (struct sockaddr *)&client, &client_len)) != -1) {
    if (fork() == 0) { /* 자식 프로세스 */
        close (sd);
        work (new); /* new를 이용해 클라이언트와 통신 */
        close (new);
        exit (0);
    }
    close (new) /* 부모 프로세스 */
}
```

# connect() 함수 (1)

- ▶ 클라이언트 프로그램에서 서버에게 연결 요청을 수행
- ▶ 사용법
  - ▶ 문법

```
#include<sys/types.h>
#include<sys/socket.h>
int connect(int s, const struct sockaddr *name, socklen_t namelen);
```

- ▶ 설명

- ▶ s: socket() 함수가 생성한 연결형 서비스용 소켓
- ▶ name: 연결하고자 하는 서버의 소켓 주소

# connect() 함수 (2)

## ▶ connect() 함수 예제

```
int sd;
struct sockaddr_in addr;

sd = socket(AF_INET, SOCK_STREAM, 0);
if(sd == -1) {
    perror("socket");
    exit(1);
}

addr.sin_family = AF_INET;
addr.sin_addr.s_addr = htonl(inet_addr("211.223.201.30"));
addr.sin_port = htons(5010);

if(connect(sd, (struct sockaddr *)&addr, sizeof(addr)) == -1) {
    perror("connect");
    exit(1);
}
```

# connect() 함수 (3)

## ▶ 주소 변환

### ▶ IP 주소의 표기 방식

- ▶ 10진수 표기 방식: 사람들의 편의를 위하여 211.223.201.30 등의 형식을 사용
- ▶ 2진수 표기 방식: IP 프로토콜에서 사용

### ▶ inet\_addr(): 10진수 형식을 2진수 형식으로 변환

### ▶ inet\_ntoa(): 2진수 형식을 10진수 형식으로 변환

### ▶ 문법

```
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<arpa/inet.h>
unsigned long inet_addr(const char *cp);
char *inet_ntoa(const struct in_addr in);
```

# send() 함수 (1)

- ▶ 연결형 서비스에서 데이터를 송신
- ▶ sendto() : 비연결형 서비스에서 데이터를 송신
- ▶ 함수 사용법
  - ▶ 문법

```
#include<sys/types.h>
#include<sys/socket.h>
ssize_t send(int s, const void *msg, size_t len, int flags);
ssize_t sendto(int s, const void *msg, size_t len, int flags,
               const struct sockaddr *to, socklen_t tolen);
```

- ▶ 설명
  - ▶ s: socket() 함수가 생성한 소켓
  - ▶ msg: 송신할 데이터
  - ▶ to: 비연결형 서비스에서 수신자 주소

# send() 함수 (2)

## ▶ send() 함수 예제

```
int sd;
struct sockaddr_in addr;
char *data = "Test Message";
int length = strlen (data) + 1;

sd = socket (AF_INET, SOCK_STREAM, 0);
addr.sin_family = AF_INET;
addr.sin_addr.s_addr = htonl (inet_addr ("211.223.201.30"));
addr.sin_port = htons (5010);
connect (sd, (struct sockaddr *)&addr, sizeof (addr));

if (send (sd, data, length, 0) == -1) {
    perror ("send");
    exit (1);
}
```

# recv() 함수 (1)

- ▶ 연결형 서비스에서 데이터를 수신
- ▶ recvfrom() : 비연결형 서비스에서 데이터를 수신
- ▶ 사용법
  - ▶ 문법

```
#include<sys/types.h>
#include<sys/socket.h>
#include<sys/uio.h>
ssize_t recv(int s, void *buf, size_t len, int flags);
ssize_t recvfrom(int s, void *buf, size_t len, int flags,
                 struct sockaddr *from, socklen_t *fromlen);
```

- ▶ 설명
  - ▶ s: socket() 함수가 생성한 소켓
  - ▶ buf: 수신할 데이터를 저장할 공간
  - ▶ from: 비연결형 서비스에서 송신자 주소

# recv() 함수 (2)

## ▶ recv() 함수 예제

```
int sd;
struct sockaddr_in addr;
char data[100];
int length = sizeof(data);

sd = socket(AF_INET, SOCK_STREAM, 0);
if (sd == -1) {
    perror("socket");
    exit(1);
}

addr.sin_family = AF_INET;
addr.sin_addr.s_addr = htonl(inet_addr("211.223.201.30"));
addr.sin_port = htons(5010);

if (connect(sd, (struct sockaddr *)&addr, sizeof(addr)) == -1) {
    perror("connect");
    exit(1);
}

if (recv(sd, data, length, 0) == -1) {
    perror("recv");
    exit(1);
}
```

# 질의 / 응답