

12장. 소켓시스템 (1)

- 주소의 표현

컴퓨터 네트워크

이번 시간의 학습 목표

- 다양한 소켓 주소 이해
- 소켓 시스템의 구조와 기능 이해

소켓 주소 (1)

- 프로토콜의 종류에 따라 사용하는 주소 체계가 다름
 - AF_UNIX: 한 호스트에 존재하는 프로세스 사이의 통신을 지원
 - AF_INET: 다른 호스트에 존재하는 프로세스 사이의 통신을 지원
- 유닉스 주소 체계
 - AF_UNIX
 - 한 호스트에 존재하는 프로세스 사이의 통신을 지원
 - 주소 체계는 파일 시스템의 경로명을 기반으로 함
 - 주소 체계

```
struct sockaddr_un {  
    short sun_family;    /* AF_UNIX */  
    char sun_path[108]; /* pathname */  
};
```

소켓 주소 (2)

- 인터넷 주소 체계
 - AF_INET
 - 다른 호스트에 존재하는 프로세스 사이의 통신을 지원
 - 주소 체계는 32비트 IP 주소와 16 비트 포트 번호를 기반으로 함
 - 주소 체계

```
struct sockaddr_in {
    short sin_family;           /* AF_INET */
    u_short sin_port;          /* Port Number */
    struct in_addr sin_addr;    /* IP Address */
    char sin_zero[8];          /* Padding */
};

struct in_addr {
    u_long s_addr;
};
```

소켓 주소 (3)

- 통합 주소 체계
 - 프로토콜마다 주소 체계를 지원하는 문법 구조가 다름
 - 문법 구조상 하나의 함수에서 다양한 주소 체계를 지원하는데 어려움이 있음
 - 따라서 모든 주소 체계를 수용할 수 있는 공통 주소 체계가 필요함
- 주소 체계

```
struct sockaddr {  
    u_short sa_family;           /* AF_UNIX, AF_INET, ... */  
    char sa_data[14];  
};
```

소켓 주소 (4)

- 통합 주소 체계

- 사용 예

- addr: 주소 공간 자체는 해당 프로토콜의 주소 체계로 선언 (인터넷 주소 체계)
 - bind() 함수의 두 번째 매개 변수는 문법적으로 공통 주소 체계만 수용

```
struct sockaddr_in addr;                /* 인터넷 주소 체계로 변수 선언 */
addr.sin_family = AF_INET;
addr.sin_addr.s_addr = htonl(INADDR_ANY); /* IP 주소 */
addr.sin_port = htons(5010);           /* 포트 번호 */
bind(socket, (struct sockaddr *)&addr, sizeof(addr));
```

소켓 서비스 (1)

- 소켓 유형
 - SOCK_STREAM
 - 연결형 서비스를 지원
 - AF_INET에서는 TCP 프로토콜을 사용
 - SOCK_DGRAM
 - 비연결형 서비스를 지원
 - AF_INET에서는 UDP 프로토콜을 사용
 - SOCK_RAW
 - IP 프로토콜을 직접 사용

소켓 서비스 (2)

- 소켓 함수
 - `s = socket (int domain, int type, int protocol)`
 - 매개 변수로 지정된 유형을 지원하는 소켓을 생성
 - 생성된 소켓을 가리키는 파일 디스크립터를 리턴
 - `bind (int s, struct sockaddr *name, socklen_t *namelen)`
 - `s`가 가리키는 소켓에 소켓 주소를 부여함
 - `name`: 소켓 주소
 - `listen (int s, int backlog)`
 - 소켓을 활성화 시킴
 - `accept (int s, struct sockaddr *addr, socklen_t *addrlen)`
 - 클라이언트/서버 환경에서 서버가 대기하는 역할을 함
 - 클라이언트의 `connect()` 함수와 만나면 소켓 연결을 설정함

소켓 서비스 (3)

- 소켓 함수
 - connect (int s, struct sockaddr *name, socklen_t namelen)
 - 클라이언트/서버 환경에서 클라이언트의 연결 설정 요청을 수행함
 - 서버의 accept() 함수와 만나면 소켓 연결을 설정함
 - send (int s, void *msg, size_t len, int flags)
 - 연결이 설정된 소켓에 데이터를 송신
 - 전송 데이터는 msg가 가리킴
 - recv (int s, void *buf, size_t len, int flags)
 - 연결이 설정된 소켓에서 데이터를 수신
 - 수신 데이터는 buf가 가리키는 공간에 저장됨