12 소켓을 이용한 네트워크 프로그래밍

쉽게 배우는 데이터 통신과 컴퓨터 네트워크

학습목표

- 소켓의 주소 체계와 서비스를 이해
- 소켓 기본 함수의 문법과 사용법.
- 소켓을 이용한 연결형 서버-클라이언트 구조를 이해
- 소켓을 이용한 비연결형 서버-클라이언트 구조를 이해

1절. 소켓의 주소 체계와 서비스

소켓

- 네트워크 프로그램을 쉽게 개발할 수 있도록 운영 체제에서 제공하는 인터페이스(운영체제의 한 부분)

소켓 주소

- 프로토콜의 종류에 따라 사용하는 주소 체계가 다름

- 유닉스 주소 체계
  - AF_UNIX
    - 한 호스트에 존재하는 프로세스 사이의 통신을 지원
    - 같은 시스템이므로 주소 체계는 파일 시스템의 경로명을 기반으로 함
  - 주소 체계

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

유닉스 주소체계를 의미하는 AF_UNIX 값

파일 시스템 경로를 기록

1절. 소켓의 주소 체계와 서비스

소켓 주소

- **인터넷 주소 체계**
  - **AF_INET**
    - 다른 호스트에 존재하는 프로세스 사이의 통신을 지원
    - 주소 체계는 32비트 IP 주소와 16 비트 포트 번호를 기반으로 한 인터넷 주소체계
  - **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;
};
```

인터넷 주소체계를 의미하는 AF_INET 값

포트번호(2바이트)

IP 주소(4바이트)

사용하지 않음(8바이트)

32비트

1절. 소켓의 주소 체계와 서비스

소켓 주소

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
- 인터넷 주소와 포트 주소를 쉽게 입력하여 사용함(프로그래밍)

- 주소 체계

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

Socketaddr_in 구조체의
포트(2), IP 주소(4), 패딩주소와 호환(8)

1절. 소켓의 주소 체계와 서비스

소켓 주소

- **통합 주소 체계**
  - **사용 예**
    - **addr: 주소 공간 자체는 해당 프로토콜의 주소 체계로 선언 (인터넷 주소체계 이용)**
    - **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 스크립트 번호

시스템 콜에서는 문법적으로 sockaddr 구조로 표현
즉, 통합 주소체계를 사용하는 형식

bind, connect, accept, recvfrom, sendto 함수를 사용시 통합 주소 체계를 사용
결국은 인터넷 주소를 통합 주소 체계로 형변환해서 사용

1절. 소켓의 주소 체계와 서비스

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 사용해 통신할 때 사용

1절. 소켓의 주소 체계와 서비스

소켓의 기본 함수(7가지)

- s = socket (int domain, int type, int protocol)
  - 매개 변수로 지정된 유형을 지원하는 소켓을 생성
  - 생성된 소켓을 가리키는 소켓 디스크립터를 리턴(파일 오픈 기능과 유사)

- bind (int s, struct sockaddr *name, socklen_t *namelen)
  - s가 가리키는 소켓에 자신의 소켓 주소를 부여함
  - name: 바인드될 소켓 주소 값

- listen (int s, int backlog)
  - 소켓을 활성화 시키는 것으로 보통 서버에서 사용
  - 연결을 거부하지 않고 대기할 수 있는 수를 표시

- accept (int s, struct sockaddr *addr, socklen_t *addrlen)
  - 클라이언트/서버 환경에서 서버가 대기하는 역할을 함
  - 클라이언트의 connect() 함수와 만나면 소켓 연결을 설정하고 새로운 디스크립트를 반환(클라이언트와 통신시 이 소켓을 사용함)

1절. 소켓의 주소 체계와 서비스

소켓 함수

- connect (int s, struct sockaddr *name, socklen_t namelen)
  - 클라이언트/서버 환경에서 클라이언트의 연결 설정 요청을 수행함
  - name 이 가리키는 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

1절. 소켓의 주소 체계와 서비스

연결형 서비스

- TCP를 이용한 통신 절차

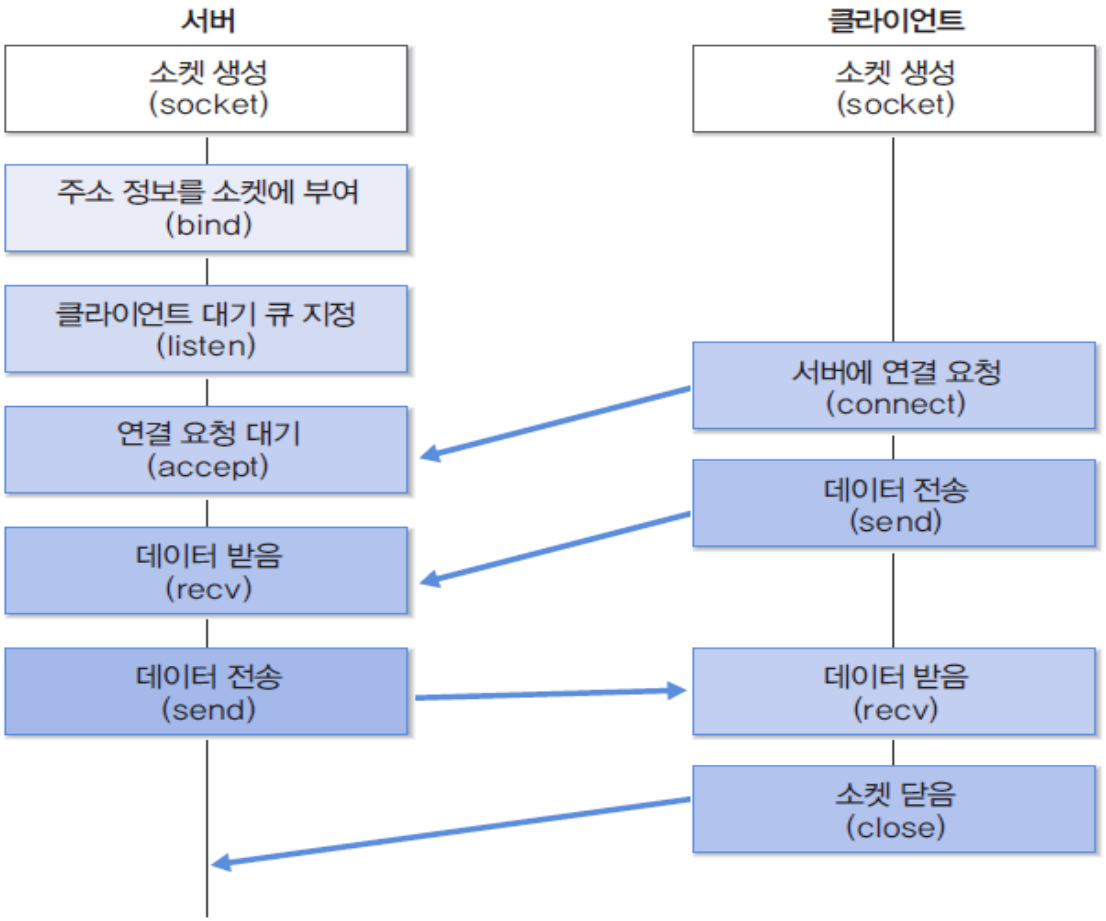

그림 12-3 TCP를 이용한 통신 절차