

컴퓨터 네트워크

# 12장. 소켓시스템 (3)

## - 네트워크 프로그래밍

# 이번 시간의 학습 목표

- ▶ 소켓을 이용한 연결형 클라이언트-서버 구조 이해
- ▶ 소켓을 이용한 비연결형 클라이언트-서버 구조 이해

# 연결형 서비스 (1)

- ▶ 소켓: 네트워크 통신을 위한 소프트웨어 교신점
- ▶ 두 개의 독립 프로세스가 네트워크를 통해 통신하려면 논리적인 연관관계를 맺어주는 소켓 필요
- ▶ 서버의 동작
  - ▶ Well-known 포트에서 대기
  - ▶ 동작 과정
    1. 서비스 교신점 (호스트의 IP 주소, 포트 번호) 공개
    2. 클라이언트로부터 발생하는 서비스 요구 대기
    3. 클라이언트에 서비스 제공
    4. 해당 클라이언트에 서비스 제공 완료
    5. 단계 2로 이동

# 연결형 서비스 (2)

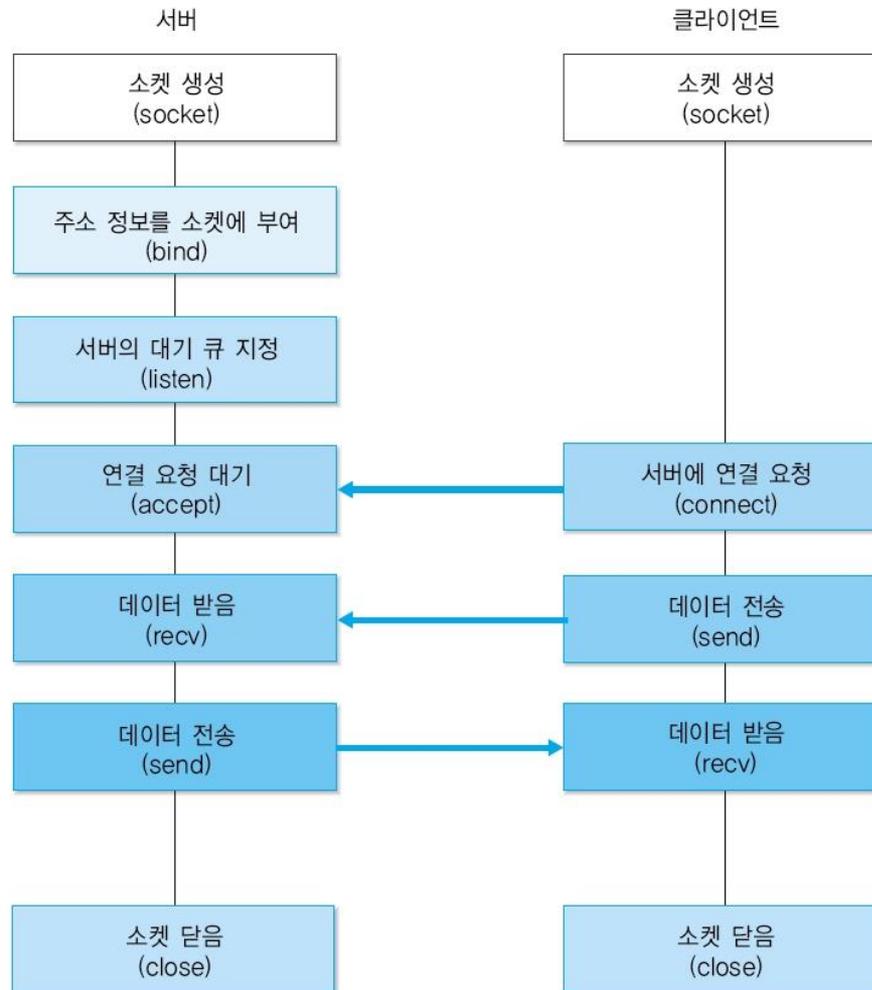
## ▶ 클라이언트의 동작

- ▶ 서버의 Well-known 포트에 접속을 시도

- ▶ 동작 과정

1. 원하는 서비스를 제공하는 서버 확인
2. 해당 서버와 연결 시도
3. 서버에 서비스 요청
4. 서버에 서비스 요구 완료

# 연결형 서비스 (3)



[그림 12-1] TCP를 이용한 통신 절차

# 연결형 서비스 (4)

## ▶ 소켓 함수의 컴파일

### ▶ 소켓 관련 라이브러리를 추가 (solaris)

```
% cc -o time_client time_client.c -lsocket -lnsl
```

```
% cc -o time_server timer_server.c -lsocket -lnsl
```

### ▶ 서버를 먼저 실행하고, 이어서 클라이언트를 실행

# 연결형 서비스 (5)

## ▶ 클라이언트 예제

```
# define TIME_SERVER  "211.223.201.30"
# define TIME_PORT    5010

main ()
{
    int sock;
    struct sockaddr_in server;
    char buf [256];

    sock = socket (AF_INET, SOCK_STREAM, 0);

    server.sin_family = AF_INET;
    server.sin_addr.s_addr = htonl (inet_addr (TIME_SERVER));
    server.sin_port = htons (TIME_PORT);

    connect (sock, (struct sockaddr *)&server, sizeof(server));

    if (recv (sock, buf, sizeof (buf), 0) == -1)
        exit (1);
    printf ("Time information from server is %s", buf);
    close (sock);
}
```

# 연결형 서비스 (6)

## ▶ 서버 예제

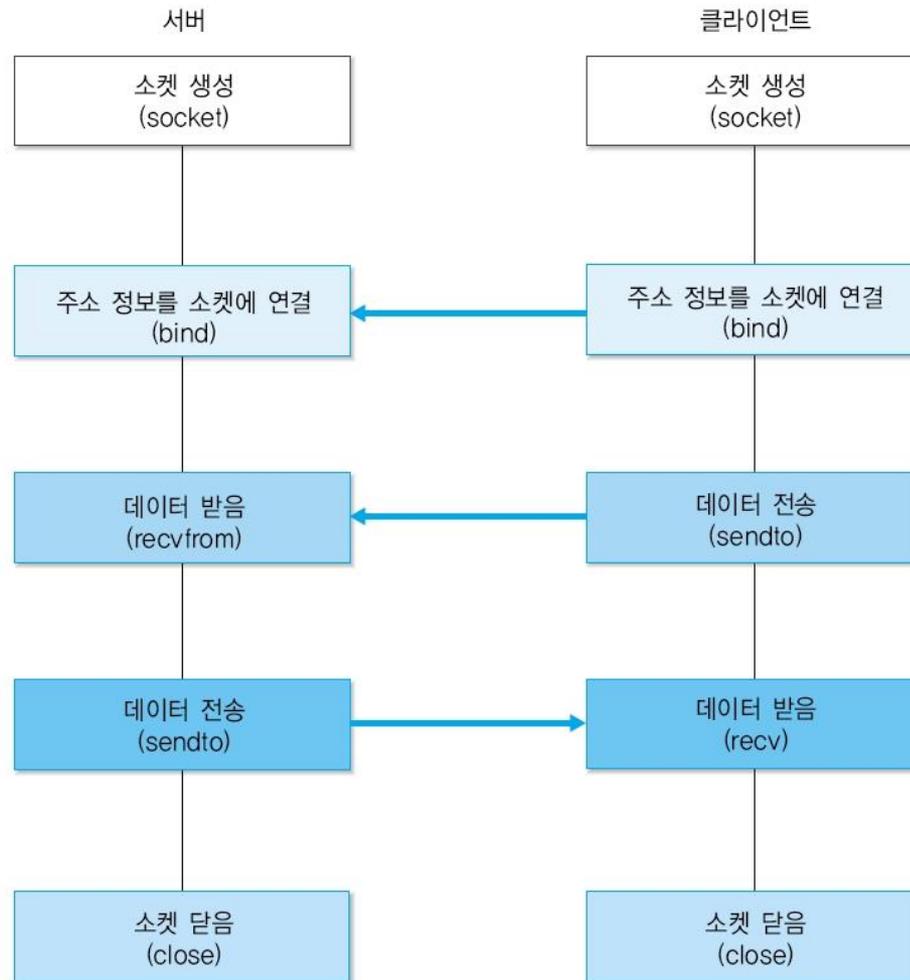
```
# define TIME_PORT 5010
main ()
{
    int sock, sock2;
    struct sockaddr_in server, client;
    int len;
    char buf [256];
    time_t today;

    sock = socket (AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = htonl (INADDR_ANY);
    server.sin_port = htons (TIME_PORT);

    bind (sock, (struct sockaddr *)&server, sizeof (server));
    listen (sock, 5);

    while (1) {
        sock2 = accept (sock, (struct sockaddr *)&client, &len);
        time (&today);
        strcpy (buf, ctime (&today));
        send (sock2, buf, strlen (buf) + 1, 0);
        close (sock2);
    }
}
```

# 비연결형 서비스 (1)



[그림 12-2] UDP를 이용한 통신 절차

# 비연결형 서비스 (2)

## ▶ 클라이언트 예제

```
main ()
{
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = htonl (inet_addr (TIME_SERVER));
    server.sin_port = htons (TIME_PORT);

    sock = socket (AF_INET, SOCK_DGRAM, 0);

    client.sin_family = AF_INET;
    client.sin_addr.s_addr = htonl (INADDR_ANY);
    client.sin_port = htons (0);
    bind (sock, (struct sockaddr *)&client, sizeof (client));

    buf[0] = '?'; buf[1] = '\0';
    buf_len = sendto (sock, buf, strlen(buf) + 1, 0, (struct sockaddr *)&server, server_len);
    if (buf_len < 0) exit (1);

    buf_len = recvfrom (sock, buf, 256, 0, (struct sockaddr *) 0, (int *) 0);
    if (buf_len < 0) exit (1);

    printf ("Time information from server is %s", buf);
}
```

# 비연결형 서비스 (3)

## ▶ 서버 예제

```
main ()
{
    time_t today;

    sock = socket (AF_INET, SOCK_DGRAM, 0);

    server.sin_family = AF_INET;
    server.sin_addr.s_addr = htonl (INADDR_ANY);
    server.sin_port = htons (TIME_PORT);

    bind (sock, (struct sockaddr *)&server, sizeof (server));

    while (1) {
        buf_len = recvfrom (sock, buf, 256, 0, (struct sockaddr *)&client, &client_len);
        if (buf_len < 0)
            exit (1);
        printf ("Server: Got %s\n", buf);

        time (&today);
        strcpy (buf, ctime (&today));
        sendto (sock, buf, strlen (buf) + 1, 0, (struct sockaddr *)&client, client_len);
    }
}
```

# 질의 / 응답