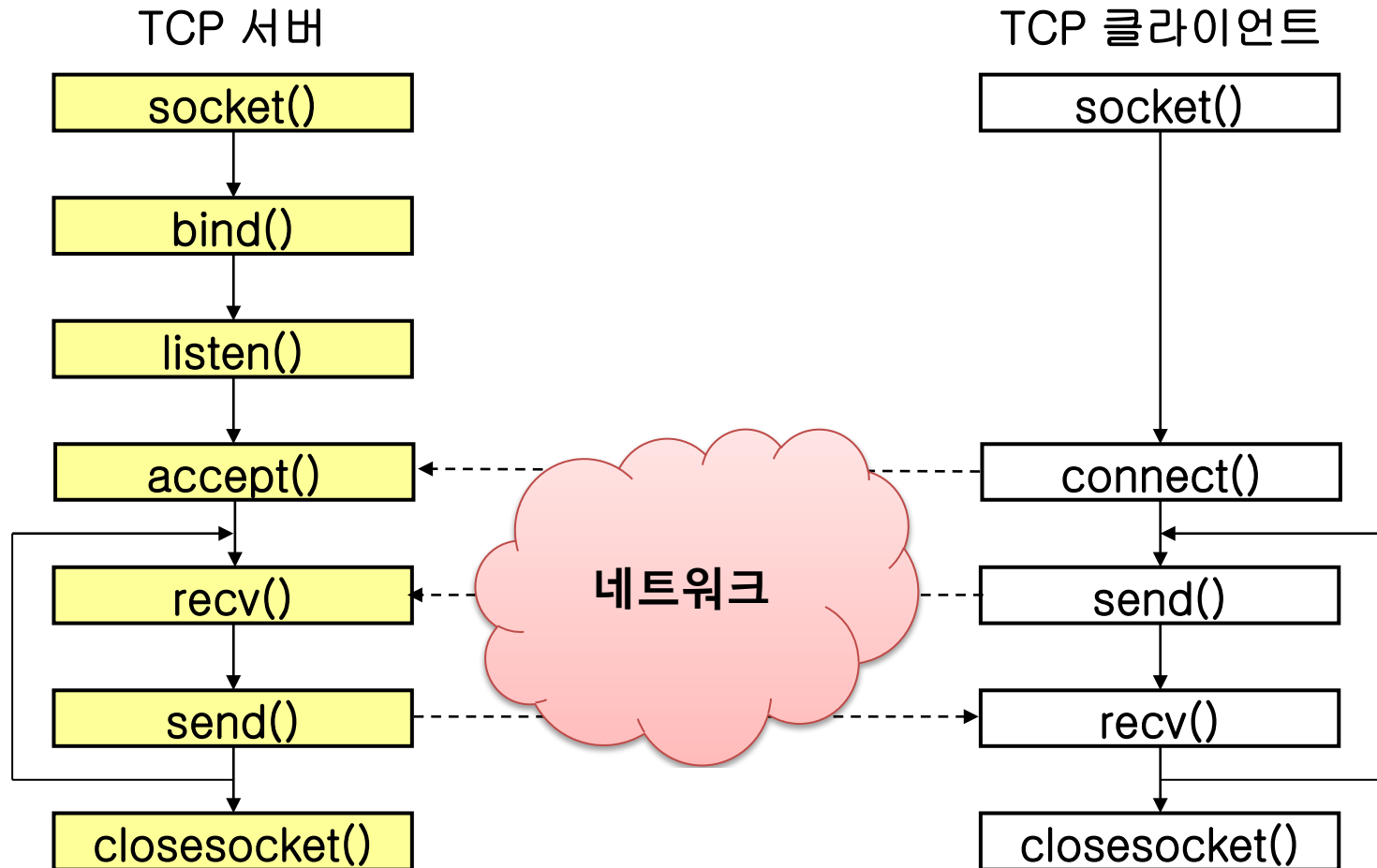

기본 서버 프로그래밍

편집: 김혜영

TCP 서버 함수 (1)

- TCP 서버 함수



서버 프로그램 만들기

bind 함수

- 소켓을 인터넷 주소에 묶어준다. (IP 주소 & Port 번호)

```
int bind(int sockfd, struct sockaddr *my_addr, socklen_t addrlen);
```

```
struct sockaddr_in addr;
```

```
addr.sin_family = AF_INET;
```

```
addr.sin_addr.s_addr = htonl(INADDR_ANY);
```

```
addr.sin_port = htons(8080);
```

```
state = bind(sockfd , (struct sockaddr *)&addr, sizeof(addr));
```

TCP 서버 함수 (2)

- **bind() 함수**
 - 소켓의 지역 IP 주소와 지역 포트 번호를 결정

```
int bind (  
    SOCKET s,  
    const struct sockaddr *name,  
    int namelen  
);
```

성공: 0, 실패: SOCKET_ERROR

TCP 서버 함수 (3)

- bind() 함수 사용 예

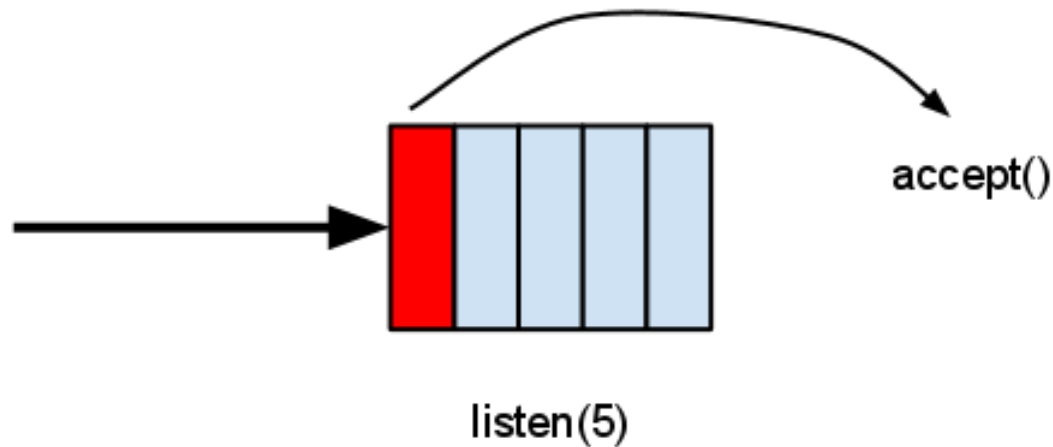
```
050  SOCKADDR_IN serveraddr;  
051  ZeroMemory(&serveraddr, sizeof(serveraddr));  
052  serveraddr.sin_family = AF_INET;  
053  serveraddr.sin_addr.s_addr = htonl(INADDR_ANY);  
054  serveraddr.sin_port = htons(SERVERPORT);  
055  retval = bind(listen_sock, (SOCKADDR *)&serveraddr, sizeof(serveraddr));  
056  if(retval == SOCKET_ERROR) err_quit("bind()");
```

서버 프로그램 흐름 만들기

listen 함수 : 수신 대기열 생성

- 클라이언트의 요청은 먼저 수신 대기열에 들어간다.

```
int listen(int queue_size);
```



TCP 서버 함수 (4)

- listen() 함수
 - 소켓의 TCP 포트 상태를 LISTENING으로 변경

```
int listen (  
    SOCKET s,  
    int backlog  
);
```

성공: 0, 실패: SOCKET_ERROR

TCP 서버 함수 (5)

- listen() 함수 사용 예

```
059     retval = listen(listen_sock, SOMAXCONN);  
060     if(retval == SOCKET_ERROR) err_quit("listen()");
```


서버 프로그램 흐름 만들기

accept 함수

- 수신 대기열의 맨 앞에 있는 클라이언트 요청을 읽는다.
- 클라이언트 요청이 있다면, 클라이언트와의 통신을 담당할 소켓 지정번호를 반환한다.

```
int accept(int s, struct sockaddr *addr,  
           socklen_t *addrlen);
```

- **s** : 듣기 소켓의 소켓 지정 번호
- **addr** : 클라이언트의 주소 정보
- **addrlen** : 두번째 매개 변수의 데이터 크기.

TCP 서버 함수 (6)

- `accept()` 함수
 - 접속한 클라이언트와 통신할 수 있도록 새로운 소켓을 생성해서 리턴
 - 접속한 클라이언트의 IP 주소와 포트 번호를 알려줌

```
SOCKET accept (  
    SOCKET s,  
    struct sockaddr *addr,  
    int *addrlen  
);
```

성공: 새로운 소켓, 실패: **INVALID_SOCKET**

TCP 서버 함수 (7)

- accept() 함수 사용 예

```
063  SOCKET client_sock;
064  SOCKADDR_IN clientaddr;
065  int addrlen;
...
068  while(1){
069      // accept()
070      addrlen = sizeof(clientaddr);
071      client_sock = accept(listen_sock, (SOCKADDR *)&clientaddr, &addrlen);
072      if(client_sock == INVALID_SOCKET){
073          err_display("accept()");
074          break;
075      }
076
077      // 접속한 클라이언트 정보 출력
078      printf("\n[TCP 서버] 클라이언트 접속: IP 주소=%s, 포트 번호=%d\n",
079          inet_ntoa(clientaddr.sin_addr), ntohs(clientaddr.sin_port));
```

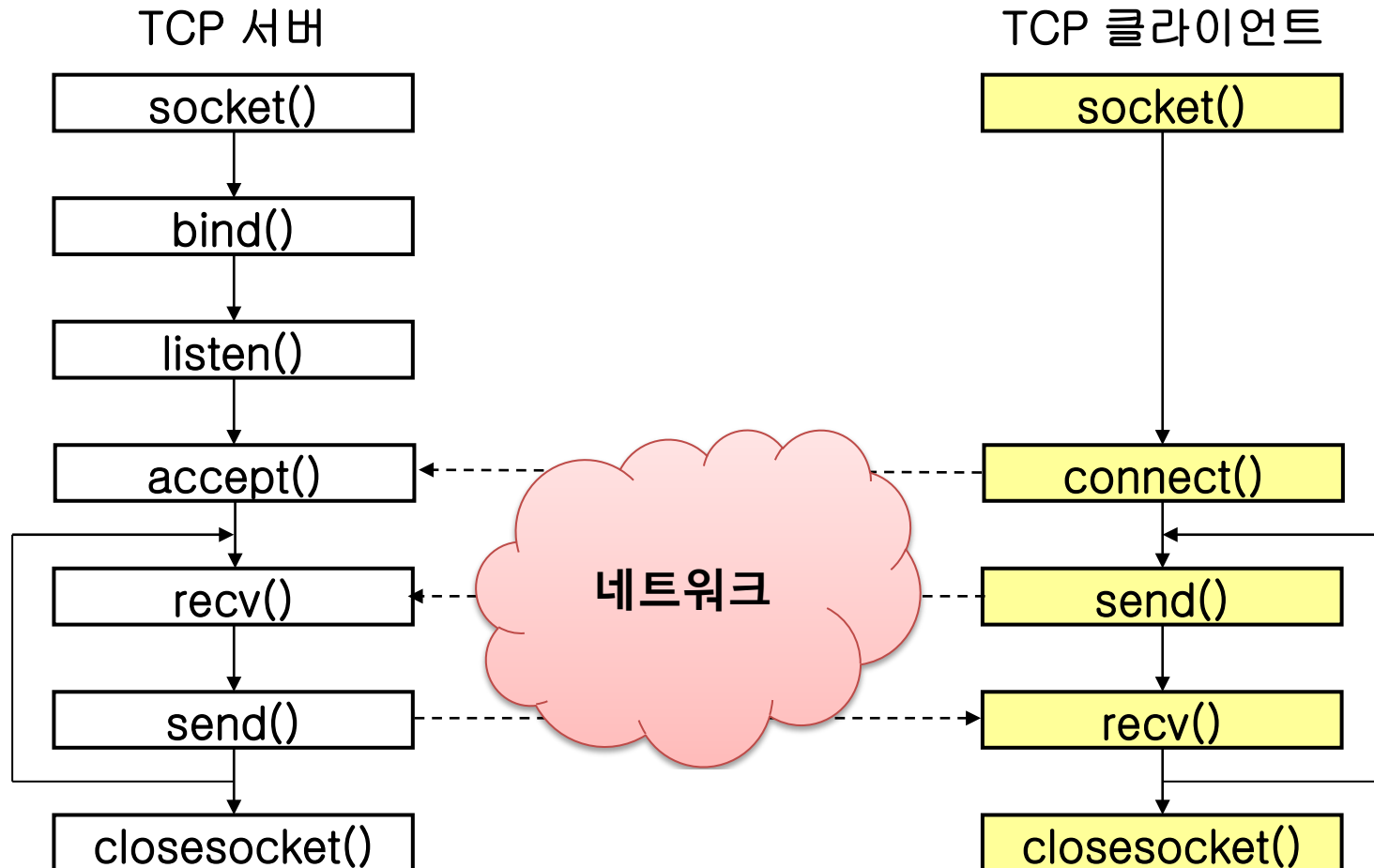
TCP 서버 함수 (8)

- accept() 함수 사용 예

```
080
081 // 클라이언트와 데이터 통신
082 while(1){
...
103 }
104
105 // closesocket()
106 closesocket(client_sock);
107 printf("[TCP 서버] 클라이언트 종료: IP 주소=%s, 포트 번호=%d\n",
108        inet_ntoa(clientaddr.sin_addr), ntohs(clientaddr.sin_port));
109 }
```

TCP 클라이언트 함수 (1)

- TCP 클라이언트 함수



TCP 클라이언트 함수 (2)

- connect() 함수
 - TCP 프로토콜 수준에서 서버와 논리적 연결을

설정

```
int connect (  
    SOCKET s,  
    const struct sockaddr *name,  
    int namelen  
);
```

성공: 0, 실패: SOCKET_ERROR

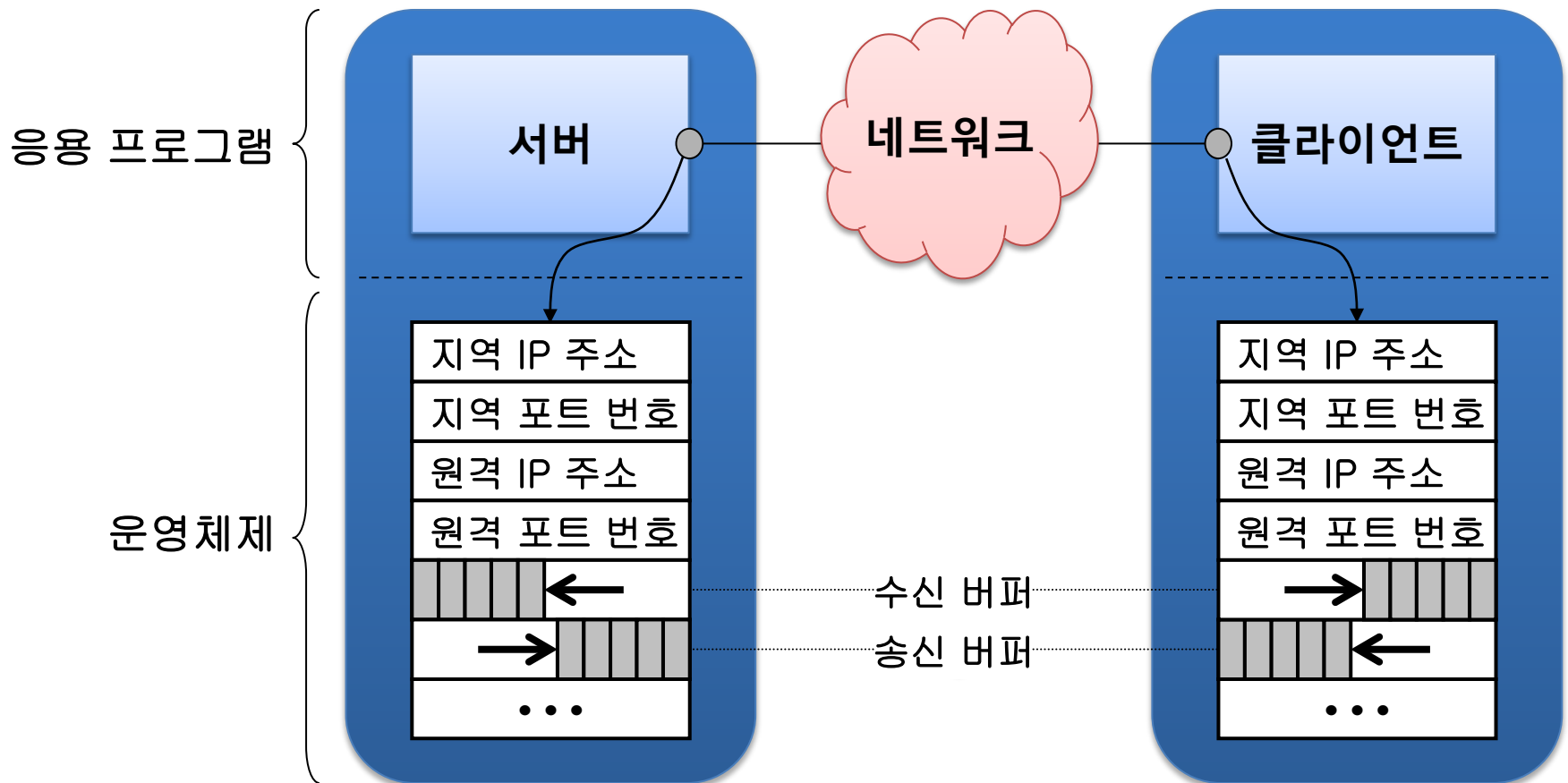
TCP 클라이언트 함수 (3)

- connect() 함수 사용 예

```
071  SOCKADDR_IN serveraddr;  
072  ZeroMemory(&serveraddr, sizeof(serveraddr));  
073  serveraddr.sin_family = AF_INET;  
074  serveraddr.sin_addr.s_addr = inet_addr(SERVERIP);  
075  serveraddr.sin_port = htons(SERVERPORT);  
076  retval = connect(sock, (SOCKADDR *)&serveraddr, sizeof(serveraddr));  
077  if(retval == SOCKET_ERROR) err_quit("connect()");
```

데이터 전송 함수 (1)

- 소켓 데이터 구조체



데이터 전송 함수 (2)

- `send()` 함수
 - 응용 프로그램 데이터를 운영체제의 송신 버퍼에 복사함으로써 데이터를 전송

```
int send (  
    SOCKET s,  
    const char *buf,  
    int len,  
    int flags  
);
```

성공: 보낸 바이트 수, 실패: `SOCKET_ERROR`

데이터 전송 함수 (3)

- **recv() 함수**
 - 운영체제의 수신 버퍼에 도착한 데이터를 응용 프로그램 버퍼에 복사

```
int recv (  
    SOCKET s,  
    char *buf,  
    int len,  
    int flags  
);
```

성공: 받은 바이트 수 또는 0(연결 종료시)
실패: **SOCKET_ERROR**

데이터 전송 함수 (4)

- recvn() 함수 정의

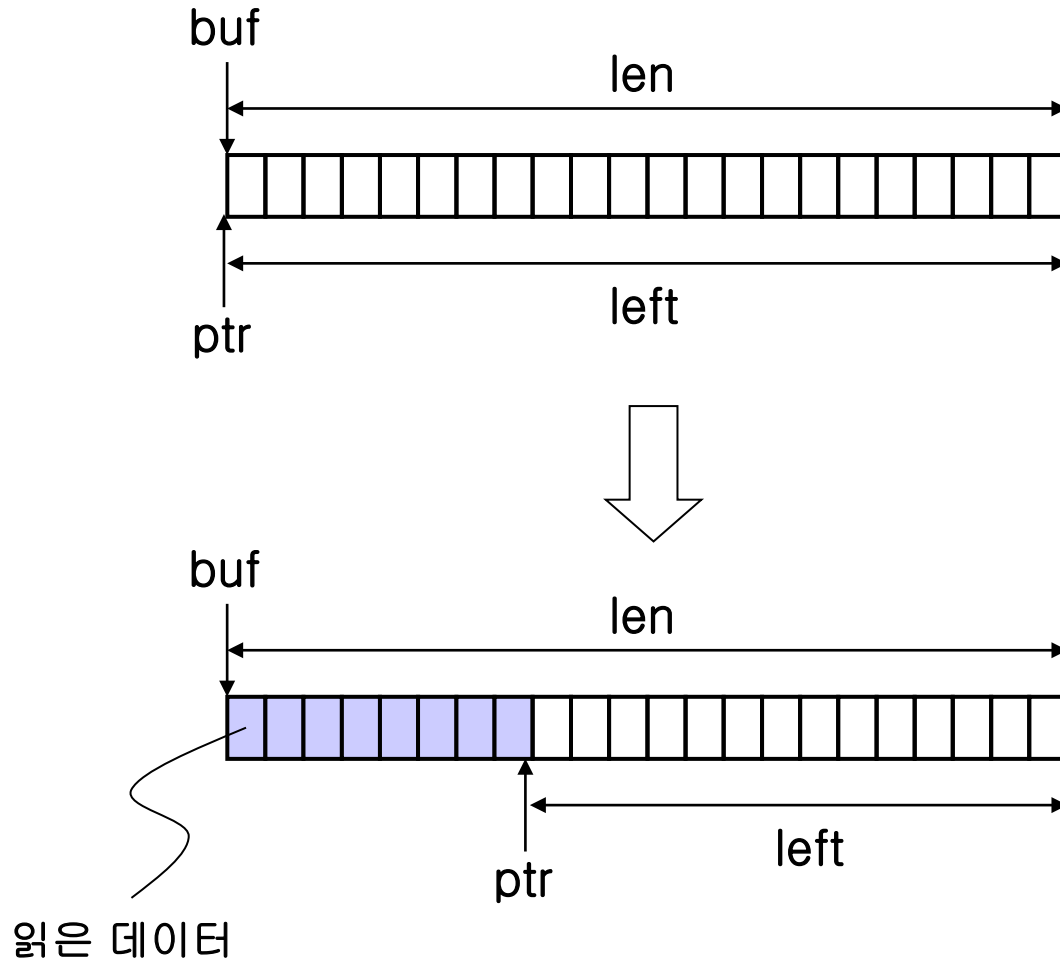
```
038 int recvn(SOCKET s, char *buf, int len, int flags)
039 {
040     int received;
041     char *ptr = buf;
042     int left = len;
043
044     while(left > 0){
045         received = recv(s, ptr, left, flags);
046         if(received == SOCKET_ERROR)
047             return SOCKET_ERROR;
048         else if(received == 0)
049             break;
050         left -= received;
051         ptr += received;
052     }
053
054     return (len - left);
055 }
```

실습 1

- 간단한 서버-클라이언트 소켓 프로그램
- 1) 강의록의 TCPServer.cpp, TCPClient.cpp
분석 후 실행해 보기
- 2) 1번의 소스를 분석 후 클래스화 시키기

데이터 전송 함수 (5)

- recvn() 함수 동작 원리



데이터 전송 함수 (6)

- 데이터 전송 함수 사용 예 – TCP 클라이언트

```
079 // 데이터 통신에 사용할 변수
080 char buf[BUFSIZE+1];
081 int len;
082
083 // 서버와 데이터 통신
084 while(1){
085     // 데이터 입력
086     printf("Wn[보낼 데이터] ");
087     if(fgets(buf, BUFSIZE+1, stdin) == NULL)
088         break;
089
090     // 'Wn' 문자 제거
091     len = strlen(buf);
092     if(buf[len-1] == 'Wn')
093         buf[len-1] = 'W0';
094     if(strlen(buf) == 0)
095         break;
```

데이터 전송 함수 (7)

- 데이터 전송 함수 사용 예 – TCP 클라이언트

```
096
097 // 데이터 보내기
098 retval = send(sock, buf, strlen(buf), 0);
099 if(retval == SOCKET_ERROR){
100     err_display("send()");
101     break;
102 }
103 printf("[TCP 클라이언트] %d바이트를 보냈습니다.\n", retval);
104
105 // 데이터 받기
106 retval = recv(sock, buf, retval, 0);
107 if(retval == SOCKET_ERROR){
108     err_display("recv()");
109     break;
110 }
111 else if(retval == 0)
112     break;
```

데이터 전송 함수 (8)

- 데이터 전송 함수 사용 예 – TCP 클라이언트

```
113
114     // 받은 데이터 출력
115     buf[retval] = '\0';
116     printf("[TCP 클라이언트] %d바이트를 받았습니다.\n", retval);
117     printf("[받은 데이터] %s\n", buf);
118 }
```


데이터 전송 함수 (9)

- 데이터 전송 함수 사용 예 – TCP 서버

```
066 char buf[BUFSIZE+1];
067
068 while(1){
...
081 // 클라이언트와 데이터 통신
082 while(1){
083     // 데이터 받기
084     retval = recv(client_sock, buf, BUFSIZE, 0);
085     if(retval == SOCKET_ERROR){
086         err_display("recv()");
087         break;
088     }
089     else if(retval == 0)
090         break;
091
```

데이터 전송 함수 (10)

- 데이터 전송 함수 사용 예 – TCP 서버

```
092      // 받은 데이터 출력
093      buf[retval] = '\0';
094      printf("[TCP/%s:%d] %s\n", inet_ntoa(clientaddr.sin_addr),
095             ntohs(clientaddr.sin_port), buf);
096
097      // 데이터 보내기
098      retval = send(client_sock, buf, retval, 0);
099      if(retval == SOCKET_ERROR){
100          err_display("send()");
101          break;
102      }
103  } ← 안쪽 while 루프의 끝
...
109  } ← 바깥쪽 while 루프의 끝
```

실습 1

- 2개 이상의 클라이언트가 접속하는 서버에서 클라이언트에게 받은 문자열에 자신의 IP를 문자열로 만들어서 함께 보내는 프로그램을 작성하시오

실습 2

- 서버에 하나의 클라이언트가 문자열을 입력하면 서버는 받은 문자열에 "from server"를 추가하여 하나의 문자열로 만들어서 클라이언트에게 보낸다.
- 클라이언트는 서버로부터 받은 문자열에 "from client"를 추가하여 서버로 보낸다.
- 이를 두 개이상의 클라이언트가 실행하도록 프로그램하시오