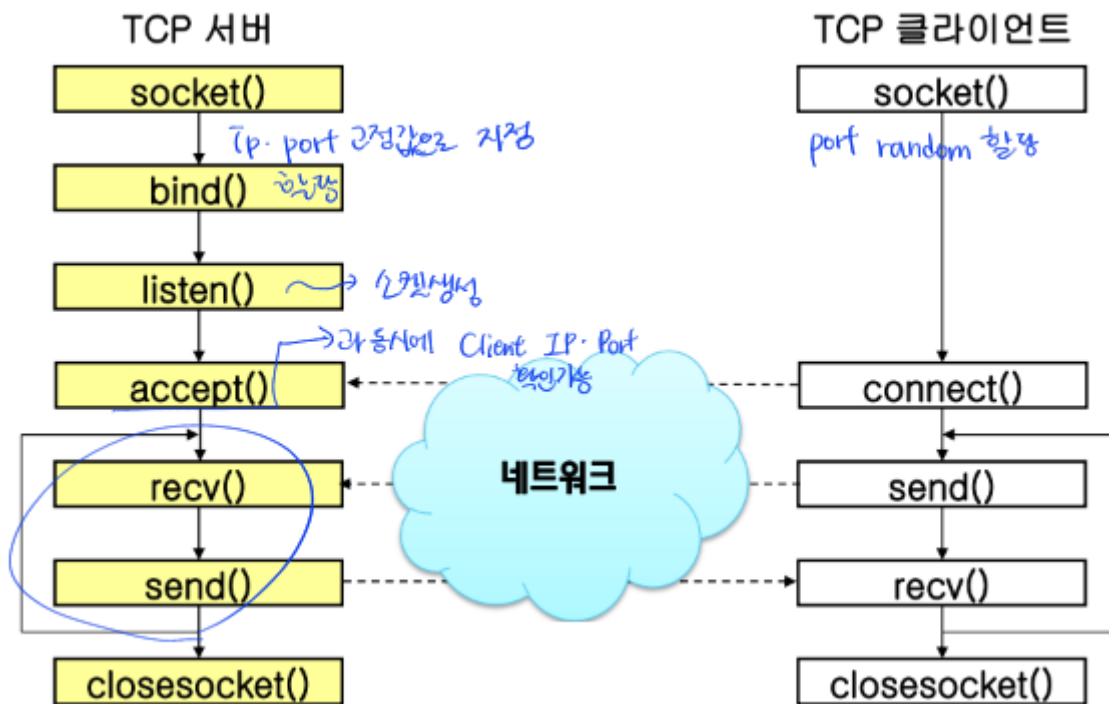


TCP Server - Client



Server: listen()

클라이언트 접속 요청받음, 최대 접속 지정 가능

Client: connect()

클라이언트가 서버에게 접속

Server: accept()

접속한 클라이언트의 정보 제공

send(), recv() == write(), read()

send(fd, buf, len, flag) recv(fd, buf, len, flag) 데이터 송수신용

▶ 실습 01: Basiccpsrv.c

```

1  #include <stdio.h>
2  #include <netinet/in.h>
3  #include <sys/socket.h>
4  #include <string.h>
5  #include <arpa/inet.h>
6  #define PORT 9001
7
8  int main()
9  {
10     //실제 소켓
11     int srvSd, clntSd;

```

```
12 //소켓의 주소 구조체
13 struct sockaddr_in srvAddr, clntAddr;
14 int clntAddrLen, readLen;
15 char rBuff[BUFSIZ];
16 char wBuff[] = "I am 20 years old.";
17
18 //IPv4, TCP로 소켓 생성
19 srvSd = socket(AF_INET, SOCK_STREAM, 0);
20 if(srvSd == -1)
21 {
22     printf("Socket Error\n");
23     return -1;
24 }
25
26 //메모리 초기화
27 memset(&srvAddr, 0, sizeof(srvAddr));
28 srvAddr.sin_family = AF_INET;
29 //INADDR_ANY: 어느 IP든 접속 가능
30 srvAddr.sin_addr.s_addr = htonl(INADDR_ANY);
31 //9001
32 srvAddr.sin_port = htons(PORT);
33
34 //소켓에 주소 설정해줌 (실제 소켓 <- 소켓 주소(sockaddr))
35 if(bind(srvSd, (struct sockaddr *) &srvAddr, sizeof(srvAddr)) == -1)
36 {
37     printf("Bind Error");
38     return -1;
39 }
40
41 //최대 5개 connection 허용
42 if(listen(srvSd, 5) == -1)
43 {
44     printf("Listen Error");
45     return -1;
46 }
47
48 clntAddrLen = sizeof(clntAddr);
49
50 //client로부터 연결 오면 clntSd 소켓에 저장
51 clntSd = accept(srvSd, (struct sockaddr*)&clntAddr, &clntAddrLen);
52 if(clntSd == -1)
53 {
54     printf("Accept Error");
55     return -1;
56 }
57
58 //client가 write 값 읽기
59 readLen = read(clntSd, rBuff, sizeof(rBuff)-1);
60 if(readLen == -1)
61 {
62     printf("Read Error");
63     return -1;
64 }
65 //개행문자 추가
66 rBuff[readLen] = '\0';
67 printf("Client: %s \n", rBuff);
68
```

```

69 //client로 값 보내기
70 write(clntSd, wBuff, sizeof(wBuff));
71
72 //소켓 닫기
73 close(clntSd);
74 close(srvSd);
75
76 return 0;
77 }
78

```

Colored by Color Scripter

▶ 실습 01: Bsictcpclnt.c

```

1  #include <stdio.h>
2  #include <netinet/in.h>
3  #include <sys/socket.h>
4  #include <string.h>
5  #define PORT 9001
6
7  int main()
8  {
9      //서버 소켓
10     int clntSd;
11     struct sockaddr_in clntAddr;
12     int clntAddrLen, readLen;
13     char wBuff[] = "How old are you?";
14     char rBuff[BUFSIZ];
15
16     //IPv4, TCP로 소켓 생성
17     clntSd = socket(PF_INET, SOCK_STREAM, 0);
18     if(clntSd == -1)
19     {
20         printf("Socket Creation Error");
21         return -1;
22     }
23     //주소 할당
24     clntAddr.sin_family = AF_INET;
25     clntAddr.sin_addr.s_addr = inet_addr("127.0.0.1");
26     clntAddr.sin_port = htons(PORT);
27
28     //server(clntSd)에 연결 요청
29     if(connect(clntSd, (struct sockaddr *) &clntAddr, sizeof(clntAddr)) == -1)
30     {
31         printf("Connect Error");
32         close(clntSd);
33         return -1;
34     }
35
36
37     //보내기
38     write(clntSd, wBuff, sizeof(wBuff));
39     printf("Client: %s\n", wBuff);
40
41     //읽기
42     readLen = read(clntSd, rBuff, sizeof(rBuff)-1);
43     if(readLen == -1)
44     {

```

```

45     printf("Read Error");
46     return -1;
47 }
48 rBuff[readLen] = '\0';
49
50 printf("Server: %s\n", rBuff);
51
52 close(cIntSd);
53 return 0;
54 }
55

```

Colored by Color

Q1. 서버 프로그램에서 최대 sizeof(rBuff)-1를 읽고 rBuff[readLen]에 '\0'을 추가하는 이유?

printf는 문장 종결 단위인 null을 만날 때까지 출력하기 때문에..

-> 널 추가하지 않으면 메모리 leak, 버퍼뒤퍼쓰워짐 등

혹은 null 추가 말고

```
printf("%.s", readLen, rBuff);
```

형식으로도 가능

Summary

- 네트워크 주소 Network Classes (A/B/C/D..) CIDR
- 소켓 주소 Bind
- 바이트 순서 Network order = Big Host order = Little
- 주소 변환 inet_*

- addr = aton //ip to network
- network //ip to host
- ntoh //network to ip

-- IPv6

- pton //ip to network
- ntop //network to ip

-- same

- inet_addr() + ntohl() == inet_network()
- inet_network() + htonl() == inet_addr()

- 소켓 프로그램 **server** socket -> bind -> listen -> accept -> read/write -> close

client socket -> connect -> read/write -> close