

## TCP 기본

- 호스트 안에서 프로세스 식별하여 데이터 전달
- 신뢰성, 순차적
- 흐름제어, 혼잡제어, 오류제어

### TCP Header

- src port
- dst port
- seq num: tcp는 바이트스트림을 쪼개서 여러번 전송하기 때문에..
- ack num: 데이터를 잘 받았다고 알리기 위해 받은 seq+1를 ack로 전송 seq+1: 다음에 전송받을 데이터의 seq num
- header len: 기본적으로 20바이트, 값의 단위는 4바이트
- reserbe field
- TCP Flag
- Window Size 송신 window: 수신측의 ack없이 한번에 전달할 수 있는 바이트 크기 수신 window: 수신측에서 수신할 수 있는 바이트 크기
- Checksum
- Urgent Pointer

## 서버 / 클라이언트 분석

### TCP Socket

```
socket(AF_INET / PF_INET, SOCK_STREAM, IPPROTO_TCP / 0)
```

connect()를 이용해 소켓 연결 시, TCP 상태는 ESTABLISH로 변환

## 서버 모델

### Concurrent 서버 모델

동시에 여러 클라이언트에게 서비스를 제공함

### Iterative 서버 모델

다수의 클라이언트들에게 순차적인 서비스를 제공함



▶ 실습 01: tcpfirstsrv.c

```

1  #include <stdio.h>
2  #include <sys/socket.h>
3  #include <sys/types.h>
4  #include <netinet/in.h>
5  #include <string.h>
6  #include <errno.h>
7  #include <stdlib.h>
8  #include <arpa/inet.h>
9
10 void error_proc();
11
12 int main(int argc, char** argv)
13 {
14     int srvSd, clntSd;
15     struct sockaddr_in srvAddr, clntAddr;
16     int clntAddrLen, readLen, strLen;
17     char rBuff[BUFSIZ];
18
19     if(argc != 2) {
20         printf("Usage: %s [port] \n", argv[0]);
21         exit(1);
22     }
23
24     printf("Server start...\n");
25
26     //서버 소켓 생성 IPv4, TCP
27     srvSd = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);

```

```
28     if(srvSd == -1 ) error_proc();
29
30     //주소 설정
31     memset(&srvAddr, 0, sizeof(srvAddr));
32     srvAddr.sin_addr.s_addr = htonl(INADDR_ANY);
33     srvAddr.sin_family = AF_INET;
34     srvAddr.sin_port = htons(atoi(argv[1]));
35
36     //소켓에 주소 할당
37     if(bind(srvSd, (struct sockaddr *) &srvAddr, sizeof(srvAddr)) == -1)
38         error_proc();
39
40     //client 요청 기다림
41     if(listen(srvSd, 2) < 0)
42         error_proc();
43     clntAddrLen = sizeof(clntAddr);
44
45     //반복문으로 계속 주관하기
46     while(1)
47     {
48         //client 연결
49         clntSd = accept(srvSd, (struct sockaddr *)
50                        &clntAddr, &clntAddrLen);
51         if(clntSd == -1)
52             error_proc();
53
54         //client의 ip, port 출력
55         printf("client %s:%d is connected...\n",
56               inet_ntoa(clntAddr.sin_addr),
57               ntohs(clntAddr.sin_port));
58
59         while(1)
60         {
61             //client가 보낸 값 읽기
62             readLen = read(clntSd, rBuff, sizeof(rBuff)-1);
63             if(readLen == 0) break;
64             rBuff[readLen] = '\0';
65
66             //서버에 출력
67             printf("Client(%d): %s\n",
68                   ntohs(clntAddr.sin_port), rBuff);
69
70             //다시 client로 보내기
71             write(clntSd, rBuff, strlen(rBuff));
72         }
73         close(clntSd);
74     }
75     close(srvSd);
76     return 0;
77 }
78
79 void error_proc()
80 {
81     fprintf(stderr, "Error: %s \n", strerror(errno));
82     exit(1);
83 }
84
```

Colored by Color Scripter

## ▶ 실습 01: tcpfirstclnt.c

```
1  #include <stdio.h>
2  #include <netinet/in.h>
3  #include <sys/socket.h>
4  #include <stdlib.h>
5  #include <string.h>
6  #include <errno.h>
7  #include <arpa/inet.h>
8
9  void err_proc();
10 int main(int argc, char** argv)
11 {
12     int clntSd;
13     struct sockaddr_in clntAddr;
14     int clntAddrLen, readLen, recvByte, maxBuff;
15     char wBuff[BUFSIZ];
16     char rBuff[BUFSIZ];
17     if(argc != 3) {
18         printf("Usage: %s [IP Address] [Port]\n", argv[0]);
19     }
20 }
21
22 //연결할 소켓 할당
23 clntSd = socket(AF_INET, SOCK_STREAM, 0);
24 if(clntSd == -1) err_proc();
25 printf("==== client program =====\n");
26
27 memset(&clntAddr, 0, sizeof(clntAddr));
28 clntAddr.sin_family = AF_INET;
29 clntAddr.sin_addr.s_addr = inet_addr(argv[1]);
30 clntAddr.sin_port = htons(atoi(argv[2]));
31
32 //서버에 연결
33 if(connect(clntSd, (struct sockaddr *) &clntAddr,
34           sizeof(clntAddr)) == -1)
35 {
36     close(clntSd);
37     err_proc();
38 }
39
40 while(1)
41 {
42     //터미널로 입력받기
43     fgets(wBuff, BUFSIZ-1, stdin);
44     readLen = strlen(wBuff);
45
46     if(readLen < 2) continue;
47
48     //서버로 전송
49     write(clntSd, wBuff, readLen-1);
50     recvByte = 0;
51     maxBuff = BUFSIZ-1;
52
53     //끝까지 읽어오기
54     do{
```

```
55         recvByte += read(clntSd, rBuff, maxBuff);
56         maxBuff -= recvByte;
57     }while(recvByte < (readLen-1));
58
59     //읽어온 값에 null 붙이기
60     rBuff[recvByte] = '\0';
61     printf("Server: %s\n", rBuff);
62     wBuff[readLen-1]='\0';
63
64     //만약 terminal에 END입력하면 서버로 보낸 후, client 종료
65     if(!strcmp(wBuff, "END")) break;
66 }
67 printf("END ^^ \n");
68 close(clntSd);
69
70 return 0;
71 }
72
73 void err_proc()
74 {
75     fprintf(stderr, "Error: %s\n", strerror(errno));
76     exit(errno);
77 }
78
```

Colored by Color Scripter