

---

# OS/NW 과제-5

단일 서버, 멀티 클라이언트 연결 & 최대,최소값 및 주소 반환

32184074 컴퓨터공학과 정민준 - 2022년 10월 28일

---



## 본문

### <cal\_client.c 코드>

```
osnw_hw_05 | cal_client.c
Project | cal_server.c | cal_client.c | Unnamed | Nothing to run on
1 | #include ...
10 |
11 | #define PORT 3600
12 | #define IP "127.0.0.1"
13 |
14 | struct cal_data
15 | {
16 |     int left_num;
17 |     int right_num;
18 |     char op;
19 |     int result;
20 |     int min_result;
21 |     int max_result;
22 |     u_long min_add,max_add; //각 최대,최소값의 ip주소를 담을 변수
23 |     u_short min_pt,max_pt; //각 최대,최소값의 포트번호를 담을 변수
24 |     short int error;
25 | };
26 |
27 | int main(int argc, char **argv)
28 | {
29 |     struct sockaddr_in addr;
30 |
31 |     //ip주소를 할당받을 sockaddr_in의 구조체 변수 선언
32 |     struct sockaddr_in min_addr,max_addr;
33 |     int s;
34 |     int len;
35 |     int sbyte, rbyte;
36 |     struct cal_data sdata;
37 |     if (argc < 4)
38 |     {
39 |         printf("Usage : %s [num1] [num2] [op]\n", argv[0]);
40 |         return 1;
41 |     }
42 |
43 |     memset((void *)&sdata, 0x00, sizeof(sdata));
44 |     sdata.left_num = atoi(argv[1]);
45 |     sdata.right_num = atoi(argv[2]);
46 |
47 |     //입력시 문자 '*'를 문자 'c'로 인식하는 경우가 발생해서 수정함.
48 |     if(argv[3][0] == 'c')
49 |         sdata.op = '*';
50 |     else
51 |         sdata.op = argv[3][0];
52 |
53 |     s = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
54 |     if (s == -1)
55 |     {
56 |         return 1;
57 |     }
58 |
59 |     addr.sin_family = AF_INET;
```

### <cal\_client.c - 1>

```
osnw_hw_05 > cal_client.c
cal_server.c x cal_client.c x
57 }
58
59     addr.sin_family = AF_INET;
60     addr.sin_port = htons(PORT);
61     addr.sin_addr.s_addr = inet_addr(IP);
62
63     if ( connect(s, (struct sockaddr *)&addr, sizeof(addr)) == -1 )
64     {
65         printf("fail to connect\n");
66         close(s);
67         return 1;
68     }
69
70     len = sizeof(sdata);
71     sdata.left_num = htonl(sdata.left_num);
72     sdata.right_num = htonl(sdata.right_num);
73
74     sbyte = write( fd: s, buf: (char *)&sdata, nbyte: len);
75     if(sbyte != len)
76     {
77         return 1;
78     }
79
80     rbyte = read(s, (char *)&sdata, len);
81     if(rbyte != len)
82     {
83         return 1;
84     }
85     if (ntohs(sdata.error != 0))
86     {
87         printf("CALC Error %d\n", ntohs(sdata.error));
88     }
89
90
91     //서버로 부터 받은 각 최대,최소의 ip주소값을 ntohl을 통해서
92     // 변환해서 구조체 변수에 할당
93     min_addr.sin_addr.s_addr = ntohl(sdata.min_addr);
94     max_addr.sin_addr.s_addr = ntohl(sdata.max_addr);
95
96     printf("min = %d from %s:%d\n",ntohl(sdata.min_result),
97           inet_ntoa(min_addr.sin_addr),ntohs(sdata.min_pt));
98
99     printf("max = %d from %s:%d\n",ntohl(sdata.max_result),
100           inet_ntoa(max_addr.sin_addr),ntohs(sdata.max_pt));
101
102     close(s);
103     return 0;
104 }
105
```

## <cal\_client.c - 2>

## <cal\_server.c 코드>

```
osnw_hw_05 cal_server.c
cal_server.c cal_client.c
1 #include ...
10
11 #define PORT 3600
12
13 struct cal_data
14 {
15     int left_num;
16     int right_num;
17     char op;
18     int result;
19     int min_result;
20     int max_result;
21     u_long min_add,max_add; //각 최대,최소값의 ip주소를 담을 변수
22     u_short min_pt,max_pt; //각 최대,최소값의 포트번호를 담을 변수
23     short int error;
24 };
25
26 int main(int argc, char **argv)
27 {
28     //클라이언트 3개를 받아야 하기 때문에 구조체 배열을 선언
29     struct sockaddr_in client_addr[3];
30     struct sockaddr_in sock_addr;
31     int listen_sockfd;
32     //클라이언트 3개를 받아야 하기 때문에 구조체 배열을 선언
33     int client_sockfd[3];
34     int addr_len;
35     struct cal_data rdata;
36     int left_num, right_num, cal_result;
37     short int cal_error;
38
39     int min = 2147483647; //최소값을 임시로 담을 변수
40     int max = -2147483648; //최대값을 임시로 담을 변수
41
42     u_long min_address; //min값의 ip주소를 임시로 담을 변수
43     u_long max_address; //max값의 ip주소를 임시로 담을 변수
44     u_short max_port; //max값의 포트주소를 임시로 담을 변수
45     u_short min_port; //min값의 포트주소를 임시로 담을 변수
46
47     if( (listen_sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1)
48     {
49         perror("Error ");
50         return 1;
51     }
52
53     memset((void *)&sock_addr, 0x00, sizeof(sock_addr));
54     sock_addr.sin_family = AF_INET;
55     sock_addr.sin_addr.s_addr = htonl(INADDR_ANY);
56     sock_addr.sin_port = htons(PORT);
57
58     if( bind(listen_sockfd, (struct sockaddr *)&sock_addr,
59             sizeof(sock_addr)) == -1)
```

## <cal\_server.c - 1>

```
osnw_hw_05 | cal_server.c
Project | cal_server.c | cal_client.c | Unnamed | Nothing to run on
57
58     if( bind(listen_sockfd, (struct sockaddr *)&sock_addr,
59             sizeof(sock_addr)) == -1)
60     {
61         perror("Error ");
62         return 1;
63     }
64
65     if(listen(listen_sockfd, 5) == -1)
66     {
67         perror("Error ");
68         return 1;
69     }
70
71     //3개의 클라이언트를 받기위해 반복문으로 3번 진행한다.
72     for(int i=0; i<3; i++)
73     {
74         addr_len = sizeof(client_addr[i]);
75         client_sockfd[i] = accept(listen_sockfd,
76                                  (struct sockaddr *)&client_addr[i], &addr_len);
77         if(client_sockfd[i] == -1)
78         {
79             perror("Error ");
80             return 1;
81         }
82         printf("New Client Connect : %s\n",
83               inet_ntoa(client_addr[i].sin_addr));
84
85         read(client_sockfd[i], (void *)&rdata, sizeof(rdata));
86
87         char *ptr = (char *) &rdata;
88
89         for(int i=0; i< sizeof(rdata); i++)
90             printf("%02x ", *(ptr+i) & 0xFF);
91         printf("\n");
92
93         cal_result = 0;
94         cal_error = 0;
95
96         left_num = ntohs(rdata.left_num);
97         right_num = ntohs(rdata.right_num);
98
99         switch(rdata.op)
100         {
101             case '+':
102                 cal_result = left_num + right_num;
103                 break;
104             case '-':
105                 cal_result = left_num - right_num;
106                 break;
107             case '*':
108
109         }
110     }
111 }
main
Structure | Bookmarks
Version Control | Python Packages | TODO | Problems | Terminal | CMake | Services
Use of undeclared identifier 's' | clang-tidy | 121:18 | LF | UTF-8 | 4 spaces | C: osnw_hw_05 | Debug
```

## <cal\_server.c - 2>

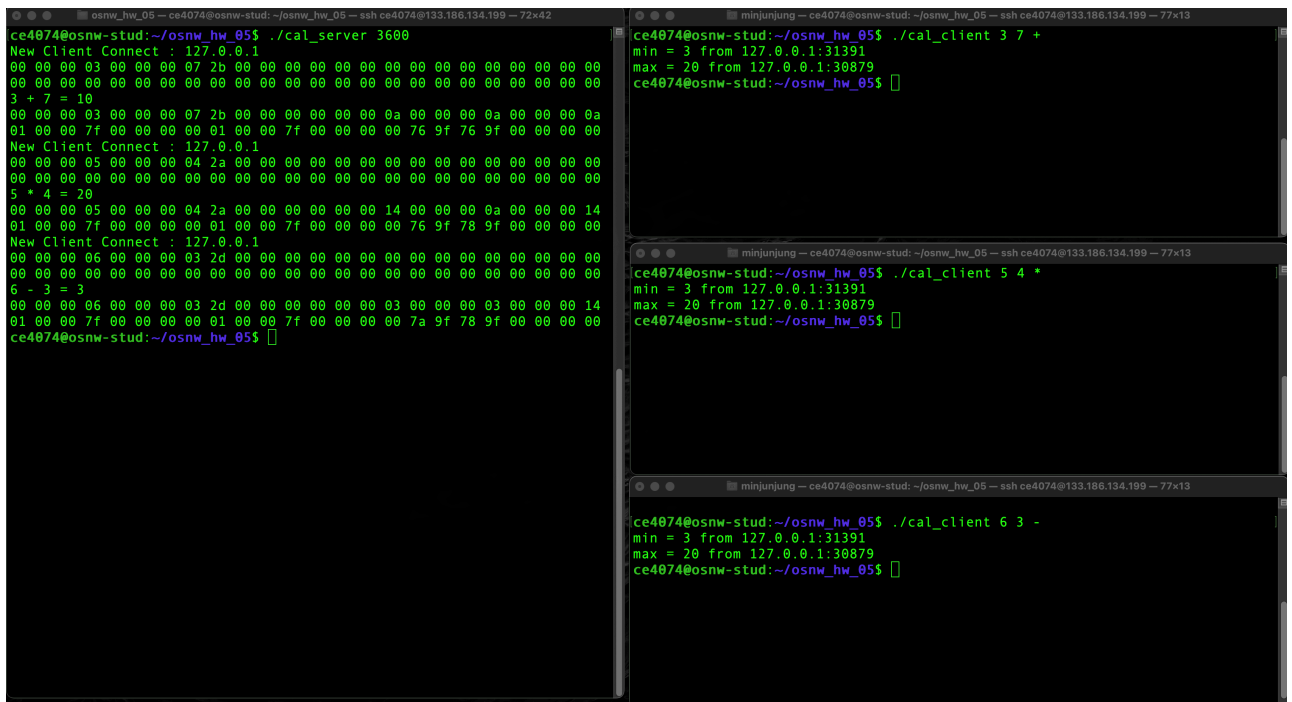
```
osnw_hw_05 > cal_server.c
100 {
107
108     case '*':
109         cal_result = left_num * right_num;
110         break;
111     case '/':
112         if(right_num == 0)
113         {
114             cal_error = 2;
115             break;
116         }
117         cal_result = left_num / right_num;
118         break;
119     default:
120         cal_error = 1;
121 }
122
123 //계산결과가 max 값보다 작다면
124 if(cal_result > max)
125 {
126     //max값을 갱신해주고 max_port,max_address를 갱신한다.
127     max = cal_result;
128     max_address = client_addr[i].sin_addr.s_addr;
129     max_port = client_addr[i].sin_port;
130 }
131
132 //계산결과가 min값 보다 작다면
133 if(cal_result < min)
134 {
135     //min값을 갱신해주고 max_port,max_address를 갱신한다.
136     min = cal_result;
137     min_address = client_addr[i].sin_addr.s_addr;
138     min_port = client_addr[i].sin_port;
139 }
140
141
142
143 rdata.result = htonl(cal_result);
144 rdata.error = htons(cal_error);
145
146 //결과값과 ip주소는 htonl() - "Host to Network Long"로 변환
147 rdata.max_result = htonl(max);
148 rdata.max_add = htonl(max_address);
149
150 //포트넘버는 htons() - "Host to Network Short"로 변환
151 rdata.max_pt = htons(max_port);
152
153 //결과값과 ip주소는 htonl() - "Host to Network Long"로 변환
154 rdata.min_result = htonl(min);
155 rdata.min_add = htonl(min_address);
156
157 //포트넘버는 htons() - "Host to Network Short"로 변환
158 }
```

### <cal\_server.c - 3>

```
136 //min값을 갱신해주고 max_port,max_address를
137 min = cal_result;
138 min_address = client_addr[i].sin_addr.s_addr;
139 min_port = client_addr[i].sin_port;
140
141 }
142
143 rdata.result = htonl(cal_result);
144 rdata.error = htons(cal_error);
145
146 //결과값과 ip주소는 htonl() - "Host to Network Long"로 변환
147 rdata.max_result = htonl(max);
148 rdata.max_add = htonl(max_address);
149
150 //포트넘버는 htons() - "Host to Network Short"로 변환
151 rdata.max_pt = htons(max_port);
152
153 //결과값과 ip주소는 htonl() - "Host to Network Long"로 변환
154 rdata.min_result = htonl(min);
155 rdata.min_add = htonl(min_address);
156
157 //포트넘버는 htons() - "Host to Network Short"로 변환
158 rdata.min_pt = htons(min_port);
159
160 printf("%d %c %d = %d \n", left_num, rdata.op,
161        right_num, cal_result);
162
163 ptr = (char *) &rdata;
164
165 for(int i=0; i< sizeof(rdata); i++)
166     printf("%02x ", *(ptr+i) & 0xFF);
167     printf("\n");
168 }
169
170 for(int i =0; i<3; i++)
171 {
172     write(fd: client_sockfd[i], buf: (void *)&rdata,
173          nbyte: sizeof(rdata));
174     close(client_sockfd[i]);
175 }
176
177 close(listen_sockfd);
178 return 0;
179 }
180
181
```

## <cal\_server.c - 4>

## <결과 화면>



```
ce4074@osnw-stud:~/osnw_hw_05$ ./cal_server 3600
New Client Connect : 127.0.0.1
00 00 00 03 00 00 00 07 2b 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
3 + 7 = 10
00 00 00 03 00 00 00 07 2b 00 00 00 00 00 00 0a 00 00 00 0a 00 00 00 0a 00 00 00 0a 00 00 00 0a 00 00 00 0a
01 00 00 7f 00 00 00 01 00 00 7f 00 00 00 00 76 9f 76 9f 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
New Client Connect : 127.0.0.1
00 00 00 05 00 00 00 04 2a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5 * 4 = 20
00 00 00 05 00 00 00 04 2a 00 00 00 00 00 00 14 00 00 00 0a 00 00 00 14 00 00 00 0a 00 00 00 14 00 00 00 14
01 00 00 7f 00 00 00 01 00 00 7f 00 00 00 00 76 9f 78 9f 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
New Client Connect : 127.0.0.1
00 00 00 06 00 00 00 03 2d 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6 - 3 = 3
00 00 00 06 00 00 00 03 2d 00 00 00 00 00 00 03 00 00 00 03 00 00 00 14 00 00 00 03 00 00 00 14 00 00 00 14
01 00 00 7f 00 00 00 01 00 00 7f 00 00 00 00 7a 9f 78 9f 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
ce4074@osnw-stud:~/osnw_hw_05$

ce4074@osnw-stud:~/osnw_hw_05$ ./cal_client 3 7 +
min = 3 from 127.0.0.1:31391
max = 20 from 127.0.0.1:30879
ce4074@osnw-stud:~/osnw_hw_05$

ce4074@osnw-stud:~/osnw_hw_05$ ./cal_client 5 4 *
min = 3 from 127.0.0.1:31391
max = 20 from 127.0.0.1:30879
ce4074@osnw-stud:~/osnw_hw_05$

ce4074@osnw-stud:~/osnw_hw_05$ ./cal_client 6 3 -
min = 3 from 127.0.0.1:31391
max = 20 from 127.0.0.1:30879
ce4074@osnw-stud:~/osnw_hw_05$
```

각 연산결과에 맞게 연산을 한 뒤 최대 값과 최소값, 그리고 해당 연산을 입력받은 ip주소와 포트 번호를 모두 동일하게 클라이언트 반환하고 있습니다.

1. 3명의 클라이언트를 받아야 하기 때문에 sockaddr\_in 의 구조체 배열을 크기3, 이름을 clientaddr 로 선언합니다.
2. 반복문을 사용해서 클라이언트로 부터 받는 연산을 3번 반복합니다.
3. 연산 결과를 각각 max,min 변수와 비교해서 현재까지 계산한 최대, 최소 값을 갱신해 줍니다.
4. Ip address는 unsigned long, port 번호는 unsigned short 형 변수에 넣어줍니다.
5. 포트 번호는 htons(), IP주소는 htonl() 함수로 변환한뒤 전달합니다.
6. 클라이언트 에서는 변환된 데이터를 ntohs(), ntohl() 함수를 변환해서 읽어드립니다.
7. 위와 같이 결과가 정상적으로 출력되는 것을 볼 수 있습니다.