

[ Computer network ]

## 소켓 프로그램 만들기

## 1. 개발환경

Ubuntu 18.04.3

## 2. 사용언어

C언어

## 3. 프로그램 내용

'소켓 통신'을 이용하여, 채팅 프로그램을 구현한다.

server를 실행시킨 상태에서, client는 총 5개까지 연결이 가능하도록 구현하였다.

연결 후 각 client에서 작성한 내용을 server가 읽고, 연결된 client들에게 broadcast한다.

## 4. 구현 방법.

### 4.1 server.c

```
server_socket = socket(AF_INET, SOCK_STREAM, 0);

rc = setsockopt(server_socket, SOL_SOCKET, SO_REUSEADDR, (char *)&on, sizeof(on));
if (rc < 0) {
    printf("setsockopt error\n");
    exit(1);
}

memset(&server_addr, 0, sizeof(server_addr));
server_addr.sin_family = AF_INET;
server_addr.sin_addr.s_addr = htonl(INADDR_ANY);
server_addr.sin_port = htons(4000);
```

server에서 소켓을 생성한다.

```
rc = bind(server_socket, (struct sockaddr *)&server_addr, sizeof(server_addr));

if (rc < 0){
    printf("bind() error\n");
    exit(1);
}

rc = listen(server_socket, 5);
if (rc < 0) {
    printf("listen() error\n");
    exit(1);
}
```

bind와 listen함수를 이용하여 client와 연결 가능 상태로 setting한다..

```
rc = setsockopt(server_socket, SOL_SOCKET, SO_REUSEADDR, (char *)&on, sizeof(on));
```

이때, 옵션을 SO\_REUSEADDR 로 주어, 이미 사용했던 주소를 time-wait 없이 재사용 가능하도록 한다.

```

void init_set()
{
    FD_ZERO(&origin_set);
    max_sd=0;
}

void add_to_set(int new_sd)
{
    FD_SET(new_sd, &origin_set);
    max_sd=MAX(new_sd, max_sd);
}

```

```
memcpy(&select_result, &origin_set, sizeof(origin_set));
```

소켓의 변경내용을 감지하도록 origin\_set을 set해주고, 이를 select\_result에 복사해준다.

While loop을 수행 할 때 마다, select\_result를 초기화하여 소켓의 변경 내용을 계속 감지하게 한다.

```
rc = select(max_sd + 1, &select_result, NULL, NULL, NULL);
```

이후 select함수를 사용하여, 소켓이 새롭게 연결되거나, client로부터 전송된 내용이 있는지 감지한다.

```
new_sd = accept(server_socket, NULL, NULL);
```

accept 함수를 이용하여 새로운 연결된 소켓을 만든다.

```

void broadcast(int sender, char* msg)
{
    int j;
    int len=strlen(msg);
    for (j=0; j <= max_sd ; ++j)
    {
        if (FD_ISSET(j, &origin_set) && j!=server_socket && j!=sender)
        {
            send(j, msg, len, 0);
        }
    }
    printf("Broadcasting '%s' finished\n" , msg);
}

```

그 상황과 내용을 전체 client들에게 send함수를 이용하여 보낸다.

## 4.2 client.c

```
if((client_socket = socket(AF_INET, SOCK_STREAM, 0)) < 0){
    printf("socket create error\n");
    exit(1);
}

memset(&server_addr, 0, sizeof(server_addr));
server_addr.sin_family = AF_INET;
server_addr.sin_port = htons(4000);
server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");

if(connect(client_socket, (struct sockaddr *)&server_addr, sizeof(server_addr)) < 0){
    printf("connected error\n");
    exit(1);
}
```

client에서 소켓을 생성하고, connect함수를 이용하여 서버로 접속을 요청한다.

```
FD_ZERO(&origin_set);
FD_SET(fileno(stdin), &origin_set);
FD_SET(client_socket, &origin_set);
max_sd=MAX(fileno(stdin), client_socket);
```

```
memcpy(&select_result, &origin_set, sizeof(origin_set));
rc = select(max_sd+1, &select_result, NULL, NULL, NULL);
```

소켓에 변화가 생기거나, stdin event를 감지하도록 origin\_set을 설정하고, 이를 select\_result에 복사해준다.

이후, select함수를 이용하여 서버에서 클라이언트에 send한 내용이 있거나(소켓에 변화)

사용자 입력 event가 있으면 이를 감지해준다.

```
if(FD_ISSET(fileno(stdin), &select_result))
{
    gets(send_buf);

    if(!strcmp(send_buf, "quit")){
        printf("finished\n");
        exit(1);
    }

    send(client_socket, send_buf, strlen(send_buf), 0);
    prompt();
}
```

stdin event가 발생하면 이를 send함수를 통해 연결된 소켓에 전송해준다.

입력된 메시지 내용이 "quit"이라면 프로그램을 종료한다.

```

else if(FD_ISSET(client_socket, &select_result))
{
    rc = recv(client_socket, recv_buf, 4096, 0);
    if (rc < 0){
        printf("recv() error.\n");
        exit(1);
    }
    if (rc == 0)
    {
        printf("disconnected\n");
        close(client_socket);
        exit(0);
    }

    recv_buf[rc]=0;
    printf("\r%s\n", recv_buf);
    prompt();
}

```

socket에 변화가 생기면, 서버에서 소켓에 전송한 내용이 있다는 것 이므로,  
recv함수를 이용하여 그 내용을 받고 화면에 출력해준다.

## 5. 실행 방법

1. Server.c 파일과 client.c파일을 각각 gcc를 이용하여 컴파일 해준다.  
(여기선 server.c의 실행파일을 server, client.c 의 실행파일을 client라고 하겠다.)
2. Server를 먼저 실행시킨 후 "server start"메시지를 확인한다.
3. Client를 실행 시킨 후 connected 메시지를 확인한다.
4. 여러 터미널 창을 띄워 여러 개의 client실행파일을 실행시킨 뒤 채팅 프로그램임을 확인한다.
5. 종료를 원하는 client는 "quit"를 입력한다.

## 6. 추가

Client와 server의 실행파일을 첨부하였으나, 코드와 함께, 컴파일 하는 방법으로 실행방법을 작성하였습니다.