

맥북 사이에 소켓 통신해보기

IP 주소 확인하기	2
Code를 다운로드 합니다.	2
Code 수정 및 실행하기	2
Server code 수정 및 실행하기	2
서버와 클라이언트의 port 번호는 일치해야 합니다.	2
client로 부터 온 메시지를 수신하는 부분	3
Client code 수정 및 실행하기	3
서버 IP 주소 설정하기	3
메시지를 서버로 전송하는 부분	4

1. IP 주소 확인하기

- 1.1. cmd 창 활성화
- 1.2. ifconfig 명령어를 통해 IP 주소를 확인한다. (서버 역할을 할 컴퓨터의

```
Last login: Thu Nov  2 12:58:16 on ttvs000
gimcheolmin-ui-MacBook-Air:~ gimcheolmin$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
    inet 127.0.0.1 netmask 0xff000000
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
    nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 94:94:26:00:5b:14
    inet6 fe80::c22:6434:cd16:beed%en0 prefixlen 64 secured scopeid 0x4
    inet 192.168.0.7 netmask 0xffffffff broadcast 192.168.0.255
    nd6 options=201<PERFORMNUD,DAD>
    media: autoselect
    status: active
```

2. Code를 다운로드 합니다.

https://github.com/Godopu/KSH_Arduino 에 이동하여 Lesson_06 폴더에 있는 server.c 와 client.c를 다운로드 합니다.

3. Code 수정 및 살펴보기

3.1. Server code 수정 및 살펴보기

- 3.1.1. 서버와 클라이언트의 port 번호는 일치해야 합니다.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include <sys/socket.h>

#define PORT 9999

#define MAXLEN 1024
void error_handling(char *message);
```

3.1.2. client로 부터 온 메시지를 수신하는 부분

```
while(1)
{
    puts("Server] Listening...");
    clnt_sock=accept(serv_sock, (struct sockaddr*)&clnt_addr,&clnt_addr_size);
    if(clnt_sock==-1)
        error_handling("accept() error");
    puts("Server] Accepted!!");

    message[0] = 0;
    idx = 0;
    while(read_len=read(clnt_sock, &message[idx], 1))
    {
        if(read_len==-1){
            error_handling("read() error!");
            exit(1);
        }

        if(message[idx++] == '\n'){
            message[idx] = 0;

            //-----write code to do-----
            printf("Message from Client: %s", message);
            //-----
            message[0] = 0 ;
            idx = 0;
        }
    }
}
```

3.2. Client code 수정 및 살펴보기

3.2.1. 서버 IP 주소 설정하기

```
4  #include <unistd.h>
5  #include <arpa/inet.h>
6  #include <sys/types.h>
7  #include <sys/socket.h>
8
9  #define IP_ADR "localhost"
10 #define PORT 9999
11
12 void error_handling(char *message);
13
```

IP_ADR 상수에 위에서 확인한 서버의 주소를 넣어주세요 (ex. #define IP_ADR "192.168.0.7")

이때 Server 역할을 할 노트북과 Client 역할을 할 노트북은 같은 공유기에 연결되어 있어야 합니다.

3.2.2. 메시지를 서버로 전송하는 부분

```
int main(int argc, char* argv[])
{
    int sock;
    struct sockaddr_in serv_addr;
    char message[30];

    sock=socket(PF_INET, SOCK_STREAM, 0);
    if(sock == -1)
        error_handling("socket() error");

    memset(&serv_addr, 0, sizeof(serv_addr));
    serv_addr.sin_family=AF_INET;
    serv_addr.sin_addr.s_addr=inet_addr("192.168.72.128");
    serv_addr.sin_port=htons(9999);

    if(connect(sock, (struct sockaddr*)&serv_addr, sizeof(serv_addr))==-1)
        error_handling("connect() error!");

    message[0] = 0;
    while(strcmp(message, "exit") != 0)
    {
        printf("Enter a message : ");
        fgets(message, sizeof(message), stdin);
        write(sock, message, strlen(message));
    }
    close(sock);
    return 0;
}
```