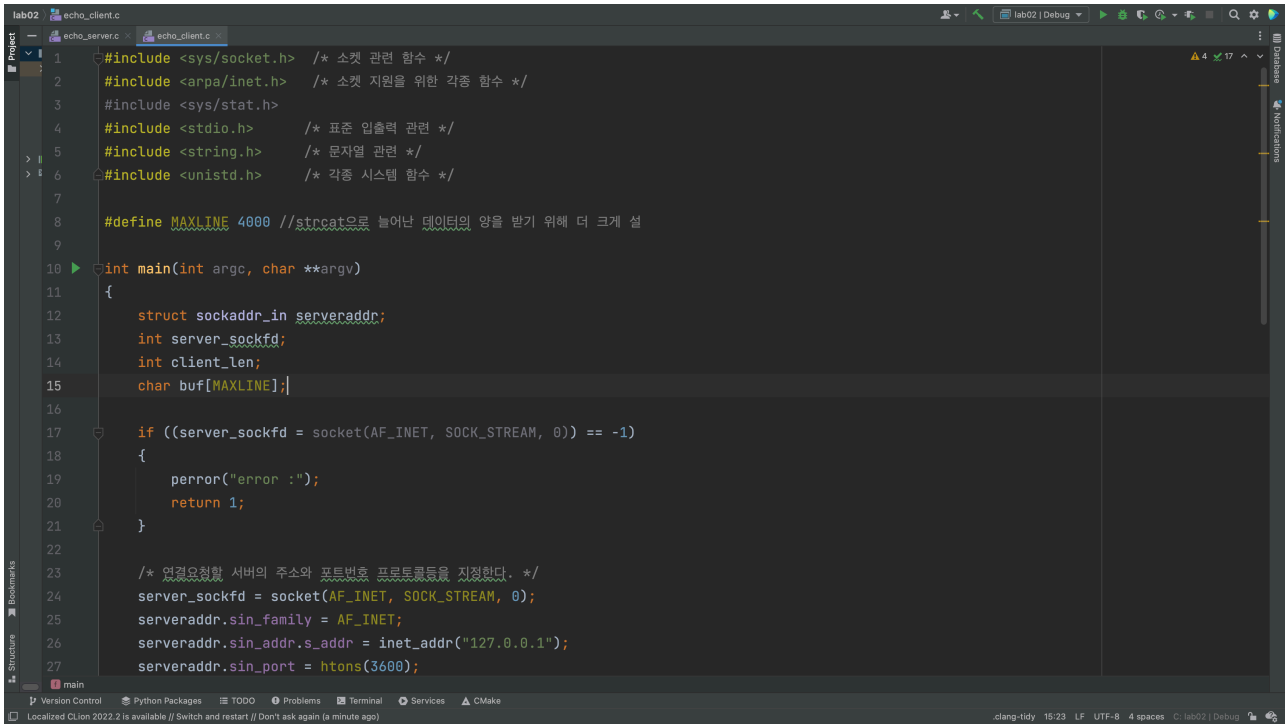
OS/NW 과제-4

단일 서버, 멀티 클라이언트 통신 프로그램

32184074 컴퓨터공학과 정민준 - 2022년 10월 17일

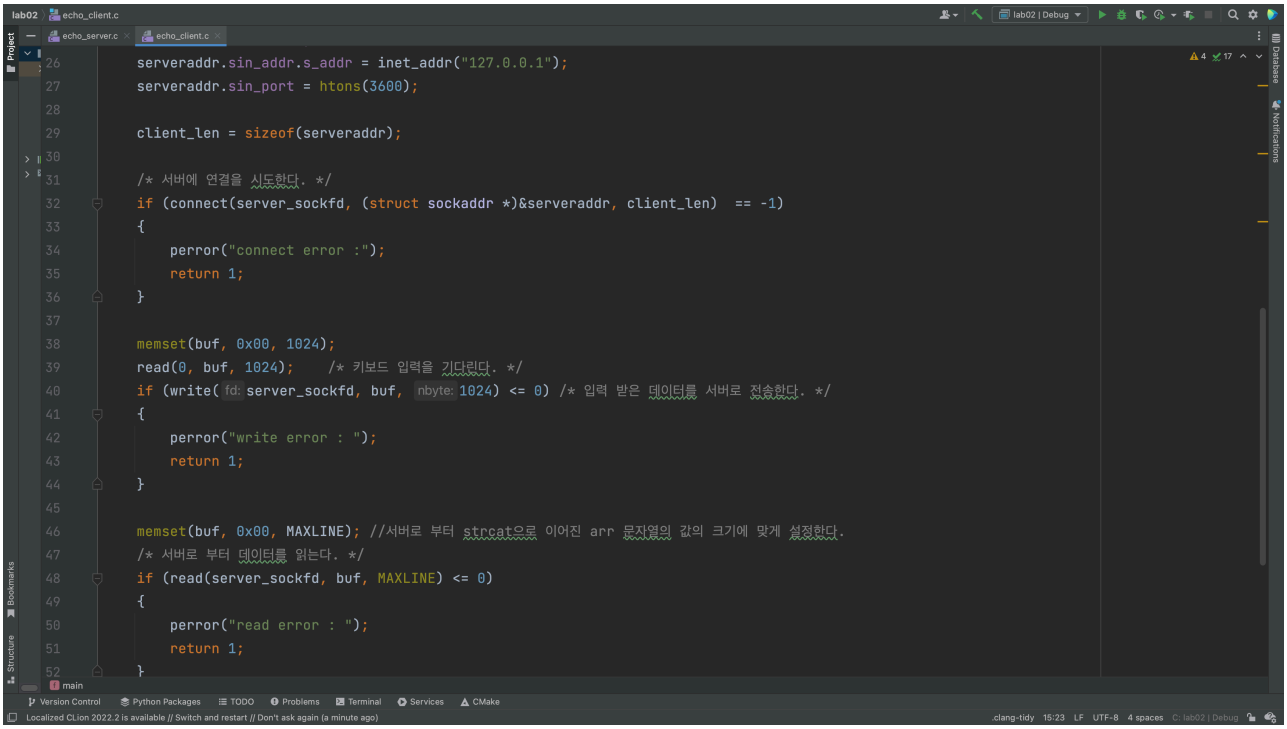
본문

<echo_client.c 코드>



```c
#include <sys/socket.h>  /* 소켓 관련 함수 */
#include <arpa/inet.h>   /* 소켓 지원을 위한 각종 함수 */
#include <sys/stat.h>
#include <stdio.h>       /* 표준 입출력 관련 */
#include <string.h>      /* 문자열 관련 */
#include <unistd.h>      /* 각종 시스템 함수 */

#define MAXLINE 4000 //strcat으로 늘어난 데이터의 양을 받기 위해 더 크게 설

int main(int argc, char **argv)
{
    struct sockaddr_in serveraddr;
    int server_sockfd;
    int client_len;
    char buf[MAXLINE];

    if ((server_sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1)
    {
        perror("error :");
        return 1;
    }

    /* 연결요청할 서버의 주소와 포트번호 프로토콜등을 지정한다. */
    server_sockfd = socket(AF_INET, SOCK_STREAM, 0);
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    serveraddr.sin_port = htons(3600);
```

<echo_client.c 코드 - 1>



```c
    serveraddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    serveraddr.sin_port = htons(3600);

    client_len = sizeof(serveraddr);

    /* 서버에 연결을 시도한다. */
    if (connect(server_sockfd, (struct sockaddr *)&serveraddr, client_len)  == -1)
    {
        perror("connect error :");
        return 1;
    }

    memset(buf, 0x00, 1024);
    read(0, buf, 1024);    /* 키보드 입력을 기다린다. */
    if (write( fd: server_sockfd, buf, nbyte: 1024) <= 0) /* 입력 받은 데이터를 서버로 전송한다. */
    {
        perror("write error : ");
        return 1;
    }

    memset(buf, 0x00, MAXLINE); //서버로 부터 strcat으로 이어진 arr 문자열의 값의 크기에 맞게 설정한다.
    /* 서버로 부터 데이터를 읽는다. */
    if (read(server_sockfd, buf, MAXLINE) <= 0)
    {
        perror("read error : ");
        return 1;
    }
```

<echo_client.c 코드 - 2>

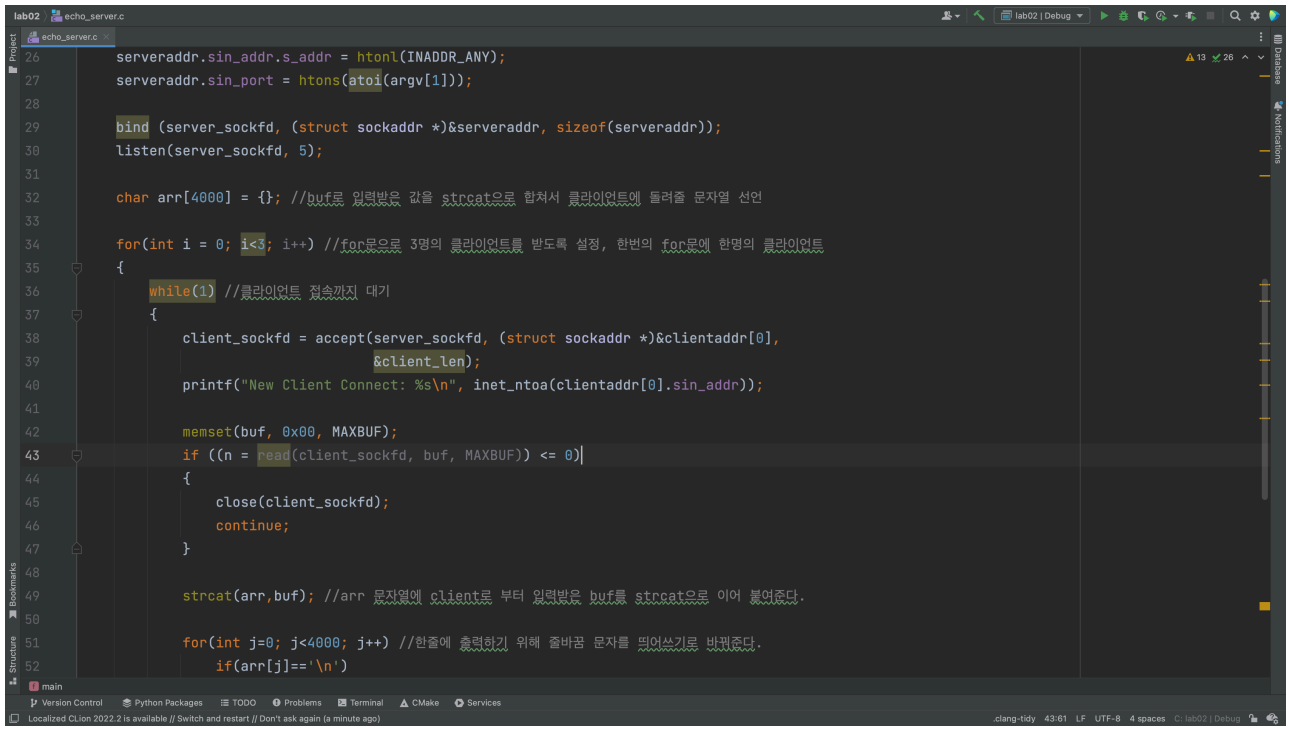
```c
    memset(buf, 0x00, 1024);
    read(0, buf, 1024);    /* 키보드 입력을 기다린다. */
    if (write( fd: server_sockfd, buf,  nbyte: 1024) <= 0) /* 입력 받은 데이터를 서버로 전송한다. */
    {
        perror("write error : ");
        return 1;
    }

    memset(buf, 0x00, MAXLINE); //서버로 부터 strcat으로 이어진 arr 문자열의 값의 크기에 맞게 설정한다.
    /* 서버로 부터 데이터를 읽는다. */
    if (read(server_sockfd, buf, MAXLINE) <= 0)
    {
        perror("read error : ");
        return 1;
    }
    printf("read : %s", buf);
    printf("\n");
    close(server_sockfd);
    return 0;
}
```

<echo_client.c 코드 - 3>

<echo_server.c 코드>

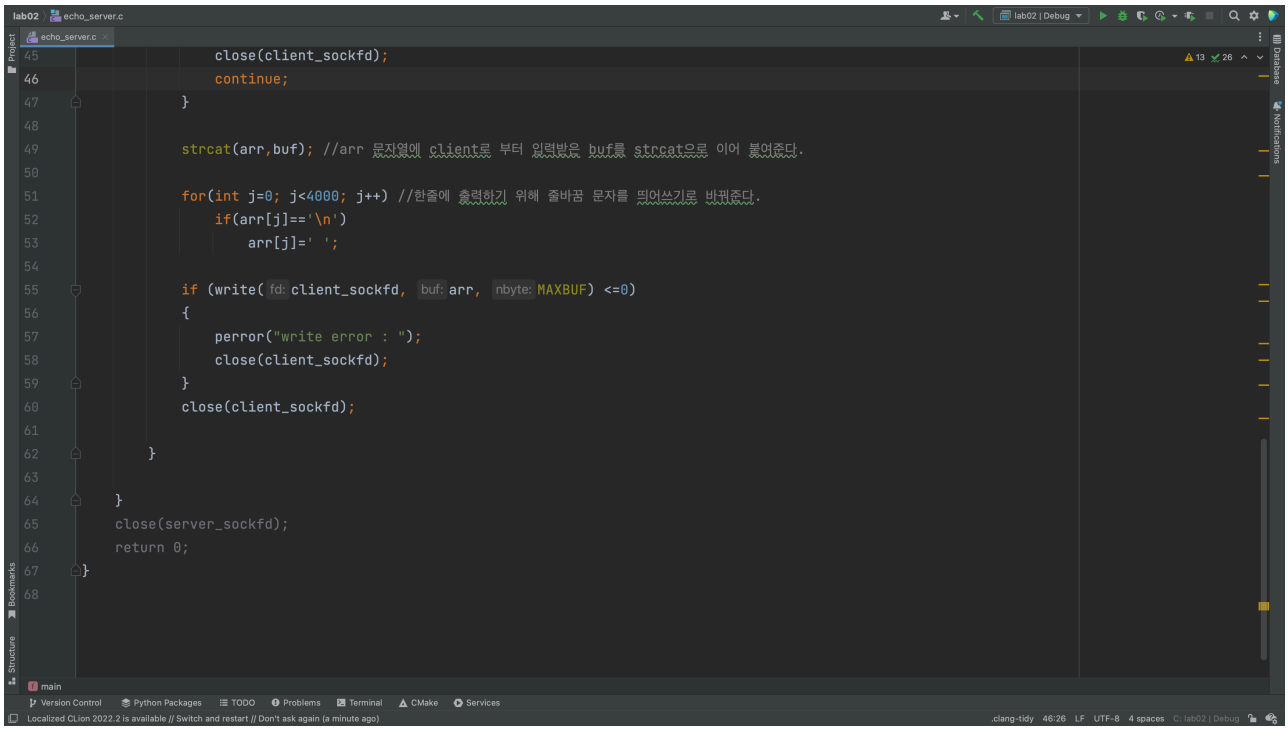
```c
#include <sys/socket.h>
#include <sys/stat.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>

#define MAXBUF 1024

int main(int argc, char **argv)
{
    int server_sockfd, client_sockfd;
    int client_len, n;
    char buf[MAXBUF];
    struct sockaddr_in clientaddr[3]; //구조체 배열로 clientaddr을 크기 3으로 선언, 각 클라이언트의 정보가 담긴다.
    struct sockaddr_in serveraddr;
    client_len = sizeof(clientaddr[0]);
    if ((server_sockfd = socket (AF_INET, SOCK_STREAM, IPPROTO_TCP )) == -1)
    {
        perror("socket error : ");
        exit(0);
    }
    memset(&serveraddr, 0x00, sizeof(serveraddr));
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_addr.s_addr = htonl(INADDR_ANY);
    serveraddr.sin_port = htons(atoi(argv[1]));
```

<echo_server.c 코드 - 1>



```
        serveraddr.sin_addr.s_addr = htonl(INADDR_ANY);
        serveraddr.sin_port = htons(atoi(argv[1]));

        bind (server_sockfd, (struct sockaddr *)&serveraddr, sizeof(serveraddr));
        listen(server_sockfd, 5);

        char arr[4000] = {}; //buf로 입력받은 값을 strcat으로 합쳐서 클라이언트에 돌려줄 문자열 선언

        for(int i = 0; i<3; i++) //for문으로 3명의 클라이언트를 받도록 설정, 한번의 for문에 한명의 클라이언트
        {
            while(1) //클라이언트 접속까지 대기
            {
                client_sockfd = accept(server_sockfd, (struct sockaddr *)&clientaddr[0],
                                       &client_len);
                printf("New Client Connect: %s\n", inet_ntoa(clientaddr[0].sin_addr));

                memset(buf, 0x00, MAXBUF);
                if ((n = read(client_sockfd, buf, MAXBUF)) <= 0)
                {
                    close(client_sockfd);
                    continue;
                }

                strcat(arr,buf); //arr 문자열에 client로 부터 입력받은 buf를 strcat으로 이어 붙여준다.

                for(int j=0; j<4000; j++) //한줄에 출력하기 위해 줄바꿈 문자를 띄어쓰기로 바꿔준다.
                    if(arr[j]=='\n')
```

<echo_server.c 코드 - 2>



```
                    close(client_sockfd);
                    continue;
                }

                strcat(arr,buf); //arr 문자열에 client로 부터 입력받은 buf를 strcat으로 이어 붙여준다.

                for(int j=0; j<4000; j++) //한줄에 출력하기 위해 줄바꿈 문자를 띄어쓰기로 바꿔준다.
                    if(arr[j]=='\n')
                        arr[j]=' ';

                if (write( fd: client_sockfd,  buf: arr,  nbyte: MAXBUF) <=0)
                {
                    perror("write error : ");
                    close(client_sockfd);
                }
                close(client_sockfd);

            }

        }
        close(server_sockfd);
        return 0;
    }
```

<echo_server.c 코드 - 3>

<결과화면>


minjunjung — ce4074@osnw-stud: ~/lab02 — ssh ce4074@133.186.134.199 — 84×18
ce4074@osnw-stud:~/lab02$ ./echo_server 3600
New Client Connect: 127.0.0.1
New Client Connect: 127.0.0.1
New Client Connect: 127.0.0.1
minjunjung — ce4074@osnw-stud: ~/lab02 — ssh ce4074@133.186.134.199 — 84×22
ce4074@osnw-stud:~/lab02$ ./echo_client
osnw
read : osnw
ce4074@osnw-stud:~/lab02$
minjunjung — ce4074@osnw-stud: ~/lab02 — ssh ce4074@133.186.134.199 — 84×18
ce4074@osnw-stud:~/lab02$ ./echo_client
32184074
read : osnw 32184074
ce4074@osnw-stud:~/lab02$
minjunjung — ce4074@osnw-stud: ~/lab02 — ssh ce4074@133.186.134.199 — 84×15
ce4074@osnw-stud:~/lab02$ ./echo_client
hello
read : osnw 32184074 hello
ce4074@osnw-stud:~/lab02$

1. 3명의 클라이언트를 받아야 하기 때문에 sockaddr_in 의 구조체 배열을 크기3, 이름을 clientaddr 로 선언합니다.
2. 반복문을 이용해서 accept 부터 read, write 과정을 3번 반복합니다.
3. arr 문자열에 client로 부터 입력받은 문자열 buf를 strcat으로 이어 붙여 줍니다.
4. 한줄에 출력하기 위해서 개행문자를 띄어쓰기로 바꿔주는 과정도 반복문으로 처리합니다.
5. 서버에서 클라이언트로 arr 문자열을 보내주고 클라이언트에서는 받은 문자열을 읽고 출력합니다.
6. 위의 사진처럼 정상적으로 결과가 나오는 것을 확인 할 수 있습니다.