

REPORT



과 목 :	네트워크프로그래밍
제출일자 :	2021. 11. 01
담당교수 :	황성호
학 과 :	컴퓨터공학과
학 번 :	201720970
이 름 :	권대한

1. UDPServer 소스코드

```
#pragma warning(disable : 4996)

#define _WINSOCK_DEPRECATED_NO_WARNINGS

#pragma comment(lib, "ws2_32")

#include <stdio.h>
#include <stdlib.h>
#include <WinSock2.h>

#define SERVERPORT 45159
#define BUFFERSIZE 512

// 소켓 함수 오류 출력 후 종료
//general error message output Func
void err_quit(char* msg)
{
    LPVOID lpMsgBuf;
    FormatMessage(
        FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM,
        NULL,
        WSAGetLastError(),
        MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPTSTR)&lpMsgBuf, 0, NULL);
    MessageBox(NULL, (LPTSTR)lpMsgBuf, msg, MB_ICONERROR);
    LocalFree(lpMsgBuf);
    exit(1);
}

// 소켓 함수 오류 출력
void err_display(char* msg)
{
    LPVOID lpMsgBuf;
    FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM,
        NULL,
        WSAGetLastError(),
        MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPTSTR)&lpMsgBuf, 0, NULL);
    printf("[%s] %s", msg, (char*)lpMsgBuf);
}
```

```

        LocalFree(lpMsgBuf);
    }

int main(int argc, char* argv[])
{
    int retval;

    // 윈속 초기화
    WSADATA wsa;
    if (WSAStartup(MAKEWORD(2, 2), &wsa) != 0)
    {
        return 1;
    }

    // Socket()
    SOCKET sock = socket(AF_INET, SOCK_DGRAM, 0);
    if (sock == INVALID_SOCKET)
    {
        err_quit("socket()");
    }

    // bind()
    SOCKADDR_IN serveraddr;
    ZeroMemory(&serveraddr, sizeof(serveraddr));
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_addr.s_addr = htonl(INADDR_ANY);
    serveraddr.sin_port = htons(SERVERPORT);
    retval = bind(sock, (SOCKADDR*)&serveraddr, sizeof(serveraddr));
    if (retval == SOCKET_ERROR)
    {
        err_quit("bind()");
    }

    // 데이터 통신에 사용할 변수
    SOCKADDR_IN clientaddr;
    int addrlen;
    char buf[BUFFERSIZE + 1];

    // 클라이언트와 데이터 통신
    while (1)
    {

```

```

// 데이터 받기
addrlen = sizeof(clientaddr);
retval = recvfrom(sock, buf, BUFFERSIZE, 0, (SOCKADDR*)&clientaddr, &addrlen);
if (retval == SOCKET_ERROR)
{
    err_display("recvfrom()");
    continue;
}

// 받은 데이터 출력
buf[retval] = '\0';
printf("[UDP /%s:%d] %s\n", inet_ntoa(clientaddr.sin_addr), ntohs(clientaddr.sin_port),
buf);

// 데이터 보내기
retval = sendto(sock, buf, retval, 0, (SOCKADDR*)&clientaddr, sizeof(clientaddr));
if (retval == SOCKET_ERROR)
{
    err_display("sendto()");
    continue;
}
}

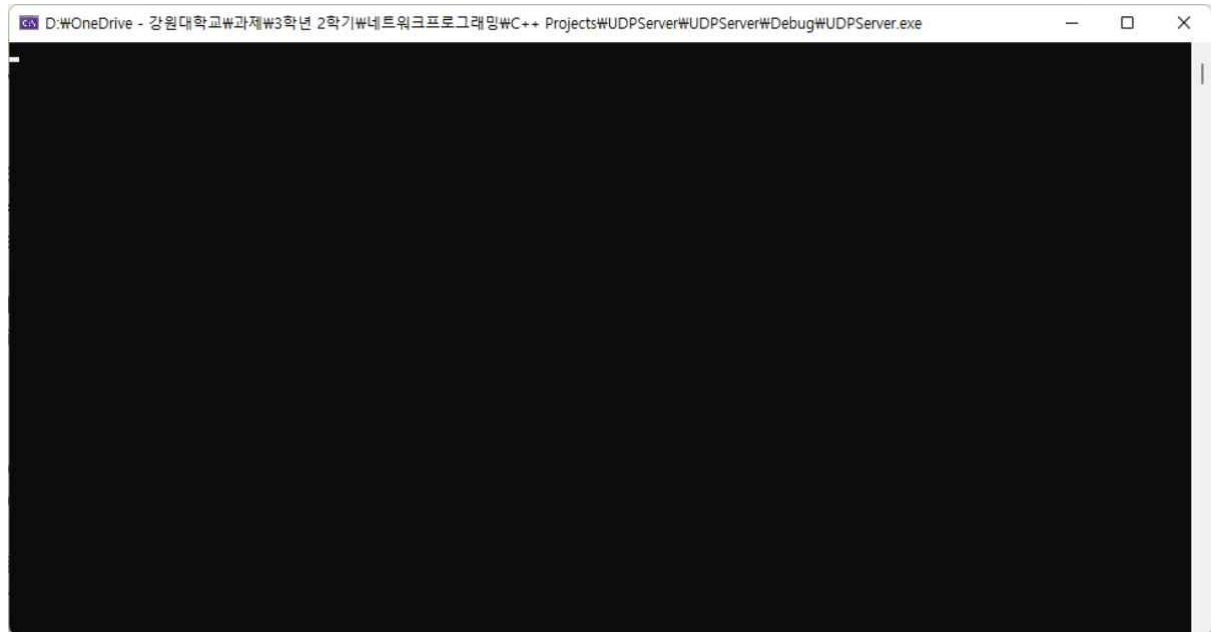
// Closesocket()
closesocket(sock);

// 윈속 종료
WSACleanup();
return 0;
}

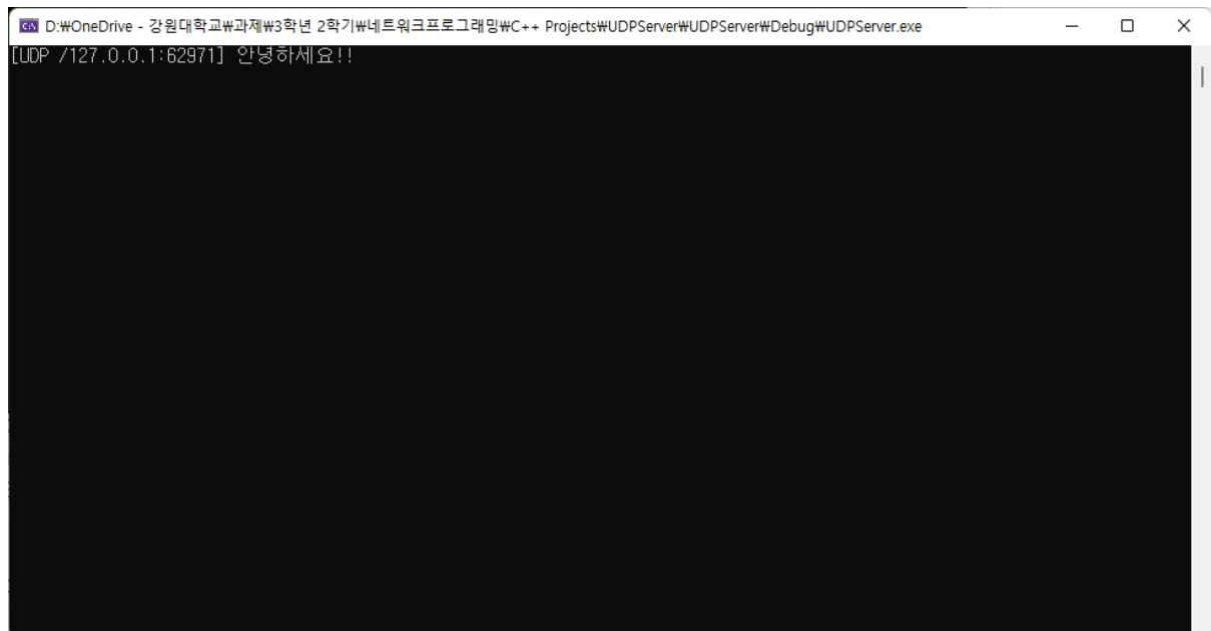
```

2. UDPServer 출력결과

Case 1. 단독으로 실행했을 경우



Case 2. UDPClient로부터 데이터를 받았을 경우



1. UDPCliet 소스코드

```
#pragma warning(disable : 4996)

#define _WINSOCK_DEPRECATED_NO_WARNINGS

#pragma comment(lib, "ws2_32")

#include <stdio.h>
#include <stdlib.h>
#include <WinSock2.h>

#define SERVERIP "127.0.0.1"
#define SERVERPORT 45159
#define BUFFERSIZE 512

// 소켓 함수 오류 출력 후 종료
//general error message output Func
void err_quit(char* msg)
{
    LPVOID lpMsgBuf;
    FormatMessage(
        FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM,
        NULL,
        WSAGetLastError(),
        MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPTSTR)&lpMsgBuf, 0, NULL);
    MessageBox(NULL, (LPCTSTR)lpMsgBuf, msg, MB_ICONERROR);
    LocalFree(lpMsgBuf);
    exit(1);
}

// 소켓 함수 오류 출력
void err_display(char* msg)
{
    LPVOID lpMsgBuf;
    FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM,
        NULL,
        WSAGetLastError(),
        MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPTSTR)&lpMsgBuf, 0, NULL);
```

```

        printf("[%s] %s", msg, (char*)lpMsgBuf);
        LocalFree(lpMsgBuf);
    }

int main(int argc, char* argv[])
{
    int retval;

    // 윈속 초기화
    WSADATA wsa;
    if (WSAStartup(MAKEWORD(2, 2), &wsa) != 0)
    {
        return 1;
    }

    // Socket()
    SOCKET sock = socket(AF_INET, SOCK_DGRAM, 0);
    if (sock == INVALID_SOCKET)
    {
        err_quit("socket()");
    }

    // 소켓 주소 구조체 초기화
    SOCKADDR_IN serveraddr;
    ZeroMemory(&serveraddr, sizeof(serveraddr));
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_addr.s_addr = inet_addr(SERVERIP);
    serveraddr.sin_port = htons(SERVERPORT);

    // 데이터 통신에 사용할 변수
    SOCKADDR_IN peeraddr;
    int addrlen;
    char buf[BUFFERSIZE + 1];
    int len;

    // 서버와 데이터 통신
    while (1)
    {
        // 데이터 입력
        printf("\n[보낼 데이터] ");
    }

```

```

if (fgets(buf, BUFFERSIZE + 1, stdin) == NULL)
{
    break;
}

// '\n' 문자 제거
len = strlen(buf);
if (buf[len - 1] == '\n')
{
    buf[len - 1] = '\0';
}
if (strlen(buf) == 0)
{
    break;
}

// 데이터 보내기
retval = sendto(sock, buf, strlen(buf), 0, (SOCKADDR*)&serveraddr, sizeof(serveraddr));
if (retval == SOCKET_ERROR)
{
    err_display("sendto()");
    continue;
}

printf("[UDP 클라이언트] %d바이트를 보냈습니다.\n", retval);

// 데이터 받기
addrlen = sizeof(peeraddr);
retval = recvfrom(sock, buf, BUFFERSIZE, 0, (SOCKADDR*)&peeraddr, &addrlen);
if (retval == SOCKET_ERROR)
{
    err_display("recvfrom()");
    continue;
}

// 송신자의 IP 주소 체크
if (memcmp(&peeraddr, &serveraddr, sizeof(peeraddr)))
{
    printf("[오류] 잘못된 데이터입니다!\n");
    continue;
}

```



```
    }


    // 받은 데이터 출력
    buf[retval] = '\0';
    printf("[UDP 클라이언트] %d바이트를 받았습니다.\n", retval);
    printf("[받은 데이터] %s\n", buf);
}

// Closesocket()
closesocket(sock);

// 윈속 종료
WSACleanup();
return 0;
}
```

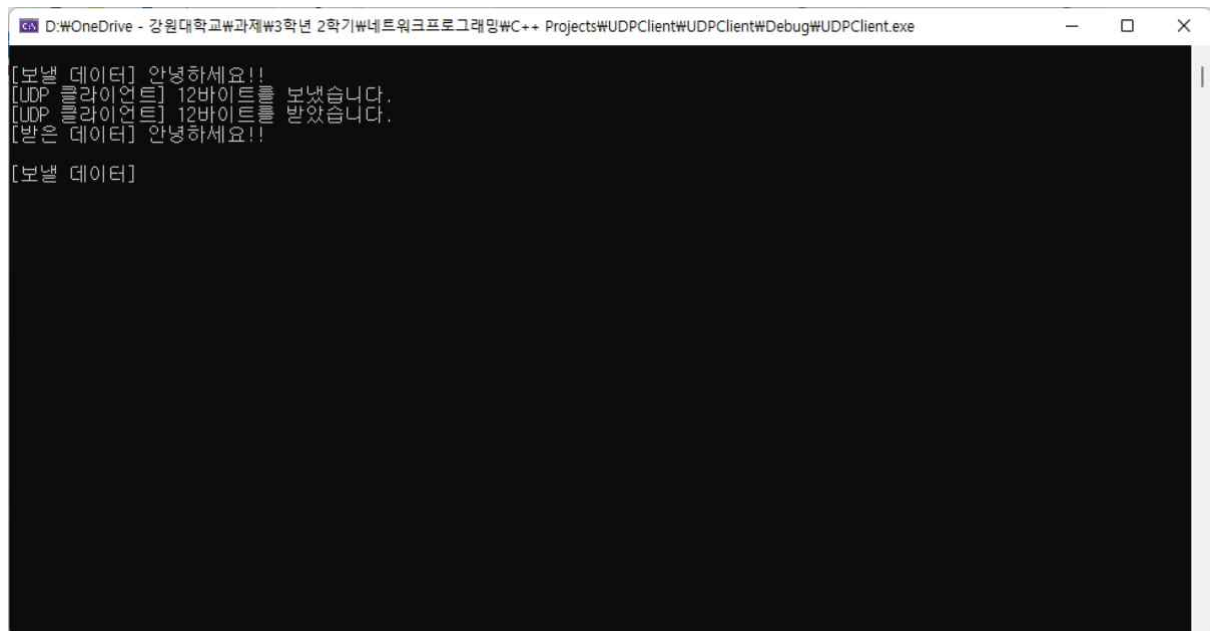
2. UDPClient 출력 결과

Case 1. 단독으로 실행했을 경우



```
D:\OneDrive - 강원대학교\과제\3학년 2학기\네트워크프로그래밍\C++ Projects\UDPClient\UDPClient\Debug\UDPClient.exe
[보낼 데이터] _
```

Case 2. UDPServer에 요청을 정상적으로 보냈을 경우



```
D:\OneDrive - 강원대학교\과제\3학년 2학기\네트워크프로그래밍\C++ Projects\UDPClient\UDPClient\Debug\UDPClient.exe
[보낼 데이터] 안녕하세요!!
[UDP 클라이언트] 12바이트를 보냈습니다.
[UDP 클라이언트] 12바이트를 받았습니다.
[받은 데이터] 안녕하세요!!
[보낼 데이터]
```