

HW 2 보고서

이름 : 정주영

학번 : 2024317

컴퓨터네트워크 01분반

UDP 서버 - 클라이언트 네트워크 통신 학습

1. UDP server - client 네트워크 통신 소스코드 분석

- UDP server (UDPServer.cpp)

```
#include "../Common.h" // 각종 헤더파일 불러 오기

#define SERVERPORT 9000 // 서버포트 매크로
#define BUFSIZE 512 // 버퍼 사이즈 매크로

int main(int argc, char *argv[])
{
    int retval; // return value 를 받는 정수형 변수 선언

    // 소켓 생성
    SOCKET sock = socket(AF_INET, SOCK_DGRAM, 0);
    // SOCKET 구조체에 socket 을 생성한다. AF_INET 은 host 내부 통신을
    의미하며 SOCK_DGRAM 은 UDP 연결임을 의미한다.
    if (sock == INVALID_SOCKET) err_quit("socket()");
    // 유효하지 않은 소켓이 생성됐을 경우의 오류 처리

    // bind()
    struct sockaddr_in serveraddr; // 인터넷 주소 체계로 변수 선언
    memset(&serveraddr, 0, sizeof(serveraddr));
    // 메모리 값을 인터넷 주소 체계만큼 초기화한다.
    serveraddr.sin_family = AF_INET; // HOST 내부 통신
    serveraddr.sin_addr.s_addr = htonl(INADDR_ANY);
    // host to network long INADDR_ANY은 랜카드 IP 주소를 나타냄
    serveraddr.sin_port = htons(SERVERPORT);
```

```

// host to network short 서버포트를 넣음
retval = bind(sock, (struct sockaddr *)&serveraddr, sizeof
(serveraddr)); // 소켓에 주소를 부여한다.
if (retval == SOCKET_ERROR) err_quit("bind()");
// bind 함수 Error 시 오류 처리

// 데이터 통신에 사용할 변수
struct sockaddr_in clientaddr; // client 소켓 구조체 선언
socklen_t addrlen; // 주소길이를 선언
char buf[BUFSIZE + 1]; // 버퍼 사이즈 지정

// 클라이언트와 데이터 통신
while (1) {
    // 데이터 받기
    addrlen = sizeof(clientaddr); // client 의 주소 길이
    retval = recvfrom(sock, buf, BUFSIZE, 0,
        (struct sockaddr *)&clientaddr, &addrlen);
    // return value 에 클라이언트로 부터 받은 데이터를 버퍼에 복사
    if (retval == SOCKET_ERROR) {
        err_display("recvfrom()");
        break;
    }
    // socket error 시의 오류 처리

    // 받은 데이터 출력
    buf[retval] = '\\0'; // recvfrom의 return 값으로 받은 데이
    터의 크기 반환
    char addr[INET_ADDRSTRLEN]; // 주소 문자열 선언
    inet_ntop(AF_INET, &clientaddr.sin_addr, addr, sizeof(a
    ddr));
    // 네트워크 주소를 사람이 읽을 수 있는 텍스트 문자열로 반환
    printf("[UDP/%s:%d] %s\\n", addr, ntohs(clientaddr.sin_
    port), buf);
    // 클라이언트 네트워크 주소와 포트와 버퍼의 데이터 출력

    // 데이터 보내기
    retval = sendto(sock, buf, retval, 0,
        (struct sockaddr *)&clientaddr, sizeof(clientadd

```

```

r));
    // 클라이언트 한테 받은 데이터를 보낸다.
    if (retval == SOCKET_ERROR) {
        err_display("sendto()");
        break;
    }
}
    // 소켓 에러시의 오류 처리

// 소켓 닫기
close(sock);
return 0;

```

요약 : UDP 로 내부 통신을 진행하여 받은 client 로 받은 데이터를 출력 후 다시 클라이언트에게 전송한다. UDP 전송 특성상 connect 없이 recvfrom 과 sendto 에 주소가 들어가게 된다.

- UDP client (UDPClient.cpp)

```

#include "../Common.h" // 각종 헤더파일을 선언한다.

char *SERVERIP = (char *)"127.0.0.1"; // 서버 IP를 localhost
로 선언
#define SERVERPORT 9000 // 서버 포트 매크로 지정
#define BUFSIZE 512 // 버퍼 사이즈 매크로 지정

int main(int argc, char *argv[])
{
    int retval; // return value 선언

    // 명령행 인수가 있으면 IP 주소로 사용
    if (argc > 1) SERVERIP = argv[1];

    // 소켓 생성
    SOCKET sock = socket(AF_INET, SOCK_DGRAM, 0);

```

```

// SOCKET 구조체에 host 내부통신으로 UDP 연결 소켓을 생성한다.
if (sock == INVALID_SOCKET) err_quit("socket()");
// 유효하지 않은 소켓 생성시 오류 처리

// 소켓 주소 구조체 초기화
struct sockaddr_in serveraddr; // 서버주소 구조체 선언
memset(&serveraddr, 0, sizeof(serveraddr));
// 메모리 값을 인터넷 주소 체계 만큼 초기화
serveraddr.sin_family = AF_INET; // host 내부 통신 선언
inet_pton(AF_INET, SERVERIP, &serveraddr.sin_addr);
// IPv4 주소로 문자열인 서버 IP 를 binary로 반환
serveraddr.sin_port = htons(SERVERPORT);
// host to network short 서버 포트를 넣어준다.

// 데이터 통신에 사용할 변수
struct sockaddr_in peeraddr;
// peer 의 주소를 넣어준다.
socklen_t addrlen; // 주소 길이 구조체를 선언한다.
char buf[BUFSIZE + 1];
int len;

// 서버와 데이터 통신
while (1) {
    // 데이터 입력
    printf("\n[보낼 데이터] ");
    if (fgets(buf, BUFSIZE + 1, stdin) == NULL)
        break; // 문자를 입력받는다.

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

    // 데이터 보내기
    retval = sendto(sock, buf, (int)strlen(buf), 0,
        (struct sockaddr *)&serveraddr, sizeof(serverad

```

```

dr));
// 생성한 socket 과 함께 buf에 있는 데이터를 전송한다. UDP
통신으로 서버 주소를 포함한다
if (retval == SOCKET_ERROR) {
    err_display("sendto()");
    break;
}
// 소켓 에러시 오류 처리
printf("[UDP 클라이언트] %d바이트를 보냈습니다.\n", retva
1);

// 데이터 받기
addrlen = sizeof(peeraddr);
retval = recvfrom(sock, buf, BUFSIZE, 0,
    (struct sockaddr *)&peeraddr, &addrlen);
// 서버에서 보낸 데이터를 받는다. 데이터를 보낸 주소를 저장할
수 있다.
if (retval == SOCKET_ERROR) {
    err_display("recvfrom()");
    break;
}
// 소켓 에러시 오류 처리

// 송신자의 주소 체크
if (memcmp(&peeraddr, &serveraddr, sizeof(peeradd
r))) {
// 메모리 블록의 내용을 비교, 서버 주소와 전송 주소를 비교로
오류 검출
    printf("[오류] 잘못된 데이터입니다!\n");
    break;
}

// 받은 데이터 출력
buf[retval] = '\0';
printf("[UDP 클라이언트] %d바이트를 받았습니다.\n", retva
1);
printf("[받은 데이터] %s\n", buf);

```

```

    }

    // 소켓 닫기
    close(sock);
    return 0;
}

```

요약 : UDP 로 내부 통신을 하여 server 에게 데이터를 보내고 다시 받는다. 이 때 UDP 는 데이터의 신뢰를 보장하지 않기 때문에 전송 주소와 서버 주소를 비교하여 오류를 검출해 낸다.

2. UDPServer.cpp 와 UDPClient.cpp를 대로 이용하여 서버 - 클라이언트 통신을 실행하고 결과 출력

- UDP server (UDPServer.cpp)

```

jooyeong@jooyeong-16Z90R-EA5CK:~/computer_network/hw2$ gcc -o server UDPServer.c

jooyeong@jooyeong-16Z90R-EA5CK:~/computer_network/hw2$ ./server
[UDP/127.0.0.1:53692] Hello
[UDP/127.0.0.1:53692] Hi
[UDP/127.0.0.1:53692] How are you?
[UDP/127.0.0.1:53692] I am a boy. You are a girl. This is my pen.
[UDP/127.0.0.1:53692] I will do my best for computer network subjects.
^[2~

```

- UDP client (UDPClient.cpp)

```
jooyeong@jooyeong-16Z90R-EA5CK: ~/computer_network/hw2
jooyeong@jooyeong-16Z90R-EA5CK:~/computer_network/hw2$ gcc -o client UDPClient.c
jooyeong@jooyeong-16Z90R-EA5CK:~/computer_network/hw2$ ./client

[보낼 데이터] Hello
[UDP 클라이언트] 5바이트를 보냈습니다.
[UDP 클라이언트] 5바이트를 받았습니다.
[받은 데이터] Hello

[보낼 데이터] Hi
[UDP 클라이언트] 2바이트를 보냈습니다.
[UDP 클라이언트] 2바이트를 받았습니다.
[받은 데이터] Hi

[보낼 데이터] How are you?
[UDP 클라이언트] 12바이트를 보냈습니다.
[UDP 클라이언트] 12바이트를 받았습니다.
[받은 데이터] How are you?

[보낼 데이터] I am a boy. You are a girl. This is my pen.
[UDP 클라이언트] 43바이트를 보냈습니다.
[UDP 클라이언트] 43바이트를 받았습니다.
[받은 데이터] I am a boy. You are a girl. This is my pen.

[보낼 데이터] I will do my best for computer network subjects.
[UDP 클라이언트] 48바이트를 보냈습니다.
[UDP 클라이언트] 48바이트를 받았습니다.
[받은 데이터] I will do my best for computer network subjects.

[보낼 데이터] □
```

3. UDP를 이용하여 파일 송수신 후에 파일 내용 출력

- UDP Server (UDPServerFile.c)

```
jooyeong@jooyeong-16Z90R-EA5CK: ~/computer_network/h...
4
❖❖ 고통을 주었다. 그 무릎 위에는 늘 내게 지어 입힐 고운 형겔 조각이나 털실 같은
것이 얹혀 있었지만, 그리고 그 입에서는 늘 나에 관한 이야기가 흘러나왔지만 나는
그것이 불만하고 불안하기조차 하였다.
그런 걸 만들어 주지 않아도 좋으니 다른 애들 엄마처럼 집안 살림에 북이어서 때로는
악도 쓰고 나더러 야단도 치고 어린애도 돌려 업고 다니고- 말하자면 그녀 자신의 생
활을

5
하고 있으면 나도 흐뭇할 것 같았다. 할아버지도 나에게와 마찬가지로 엄마에게도 그
저 유하고 부드럽기만 하였다.
엄마의 그림자 같은 생활은 언제부터 시작되었는지 기억할 수 없다. 사변과 함께 우리
가 시골 할아버지 댁으로 내려가던 때 그러니까 지금부터 십년쯤 전에도 이미 그랬었
고 또 그보다 전 서울서 국민 학교에 입학하던 즈음에도 역시 그런 느낌이던 것을 잊
지 않고 있다

6
.하고 있으면 나도 흐뭇할 것 같았다. 할아버지도 나에게와 마찬가지로 엄마에게도 그
저 유하고 부드럽기만 하였다.
엄마의 그림자 같은 생활은 언제부터 시작되었는지 기억할 수 없다. 사변과 함께 우리
가 시골 할아버지 댁으로 내려가던 때 그러니까 지금부터 십년쯤 전에도 이미 그랬었
고 또 그보다 전 서울서 국민 학교에 입학하던 즈음에도 역시 그런 느낌이던 것을 잊
지 않고 있다
█
```

- UDP Client (UDPClientFile.c)


```
jooyeong@jooyeong-16Z90R-EA5CK: ~/computer_network/h...  
악도 쓰고 나더러 야단도 치고 어린애도 돌려 업고 다니고- 말하자면 그녀 자신의 생  
활을  
하고 있으면 나도 흐뭇할 것 같았다. 할아버지도 나에게와 마찬가지로 엄마에게도 그  
저 유하고 부드럽기만 하였다.  
엄마의 그림자 같은 생활은 언제부터 시작되었는지 기억할 수 없다. 사변과 함께 우리  
가 시골 할아버지 댁으로 내려가던 때 그러니까 지금부터 십년쯤 전에도 이미 그랬었  
고 또 그보다 전 서울서 국민 학교에 입학하던 즈음에도 역시 그런 느낌이던 것을 잊  
지 않고 있다  
.하고 있으면 나도 흐뭇할 것 같았다. 할아버지도 나에게와 마찬가지로 엄마에게도 그  
저 유하고 부드럽기만 하였다.  
엄마의 그림자 같은 생활은 언제부터 시작되었는지 기억할 수 없다. 사변과 함께 우리  
가 시골 할아버지 댁으로 내려가던 때 그러니까 지금부터 십년쯤 전에도 이미 그랬었  
고 또 그보다 전 서울서 국민 학교에 입학하던 즈음에도 역시 그런 느낌이던 것을 잊  
지 않고 있다  
1하고 있으면 나도 흐뭇할 것 같았다. 할아버지도 나에게와 마찬가지로 엄마에게도 그  
저 유하고 부드럽기만 하였다.  
엄마의 그림자 같은 생활은 언제부터 시작되었는지 기억할 수 없다. 사변과 함께 우리  
가 시골 할아버지 댁으로 내려가던 때 그러니까 지금부터 십년쯤 전에도 이미 그랬었  
고 또 그보다 전 서울서 국민 학교에 입학하던 즈음에도 역시 그런 느낌이던 것을 잊  
지 않고 있다  
파일전송을 완료하였습니다.  
  
Shut down  
jooyeong@jooyeong-16Z90R-EA5CK:~/computer_network/hw2/client$ s
```