**CO SOCKET PROGRAMMING assignment #1**

학번 : 201720768 성명 : 김수영

1. 목적

Connection-oriented socket buffer 를 이용해서 client 와 server 사이에 데이터가 전해지고 받아지는 프로그램을 짠 것으로 client 와 server 사이에 connection을 맺고 나서, data를 특정 syntax 에 따라서 client 가 send(), server가 recv() 하고 다시 echo 형식으로 server 에서 send() client가 recv() 하는 과정과 부가적으로 network byte order 형식으로 데이터가 전해지는 과정을 코딩으로 제작하면서 이해한다.

1. 프로그램 설계

**Syntax**:

(1) socket 를 생성한 후에 client 가 connection request 를 server 에게 보낼 때 : sockaddr\_in(IPv4 주소를 지원하는 구조체)형식에 따라 sin\_family, sin\_port, sin\_addr, sin\_zero 로 이루어진 소켓 주소 구조체로 이루어진다.

(2) socket 과의 connect 이후, client에서 send 하는 syntax : network byte order형식인 앞에 두개의 int형 데이터와, 특정 크기의 이름 문자 배열과 1000byte 로 지정한 입력 문자 배열 로 이루어진 구조체의 포인터를 문자열 포인터로 변환, 변환된 문자열 버퍼를 입력 문자의 크기까지 합친 길이 만큼이 send() 할 때 보내지는 syntax이다.

(3) server 가 echo 형식으로 보내는 syntax : 보내는 문자열의 문자 개수의 int형 데이터와 1000byte 크기의 문자열로 이루어진 구조체를 문자열 포인터로 변환해서 생긴 버퍼를, 문자개수를 담는 2byte 와 문자열 길이를 더한 크기 만큼 자른 buffer 가 server가 보내는 syntax 이다.

**Semantics:**

1. retval = connect(sock, (SOCKADDR\*)&serveraddr, sizeof(serveraddr))

client 가 connect 할 열려 있는 server 에게 그 특정 server 주소를 입력해서 연결 syn를 보낸다. 이때, 반환되는 retval 에 들어가는 값에 따라 0의 값이면 connect 성공, -1이면 연결 오류 메시지를 command 창에 띄운다.

1. retval = send(sock, (char\*)&message, message\_len, 0)

client 에서 socket()으로 생성한 연결된 socket에 메시지와 메시지 길이를 보낸다. 이때, 반환되는 retval 값은 보낸 메시지의 byte 크기이고 -1이 반환되면 error 를 출력한다.

1. msglen = recv(client\_sock, buf, BUFSIZE, 0)

server 입장에서, accept()로 생성된 새로운 client socket을 통해 지정된 buffer 크기만큼 buffer 에 client 에 전송한 문자열을 받아오고 받은 buffer 의 크기를 반환할 때 -1이 msglen의 값으로 반환되면 error 를 출력한다.

1. retval\_send = send(client\_sock, (char\*)&sendMessage, 2+msglen-19+1, 0)

server 가 client 로 echo message 를 보내는데, 보낼 구조체를 문자열 포인터로 변환한 것과 그 문자열의 크기를 recv() 에서 연장해서 사용할 client socket를 통하여 전송한다. 이때 반환되는 값 retval\_send에 대해서는 앞서 설명한 semantics-(2)번과 마찬가지다.

1. retval = recv(sock, buf, BUFSIZE, 0)

client 입장에서 server를 통해 받은 echo 메시지를 buf라는 buffer에 bufsize 만큼 받아오고 반환되는 값에 대해서는 semantics-(3)과 동일하다.

1. 프로그램 구현
2. Client 구현

*socket(AF\_INET,SOCK\_STREAM,0);* 를 이용해서 client 에서 server 에 데이터를 전송할 때 사용될 socket 를 생성하고, 이 socket 를 통해서 *connect(sock, (SOCKADDR\*)&serveraddr, sizeof(serveraddr));* 로 server 와의 connect 를 한다. 서버와의 connect 가 완료되면 client 에서 전송할 network byte order 형식의 두개의 int 형 데이터와 이름과 특정 문자열을 가지는 구조체 형식의 data 를 만들고, *send(sock, (char\*)&message, message\_len, 0);* 를 통해서 server 에 이 data를 전송한다.

전송이후에 server에서 보내는 echo message 를 *recv(sock, buf, BUFSIZE, 0);* 를 이용해서 buf에 받고 이 buffer 를 server 에서 보내는 메시지와 같은 형태의 구조체 안에 parsing 한 후에 창에 출력한다. 만약 client 에서 처음에 QUIT 라는 문자열을 입력했다면 *strncmp(recvMessage.input, "QUIT",4) == 0 && recvMessage.inputNum == 4 && a!=0)* 로 보낸 문자열을 판별하여 display 에 print 한 후 server 와의 데이터가 공유되는 loop 를 나와 *closesocket(sock);*를 통해 socket 를 close 한다.

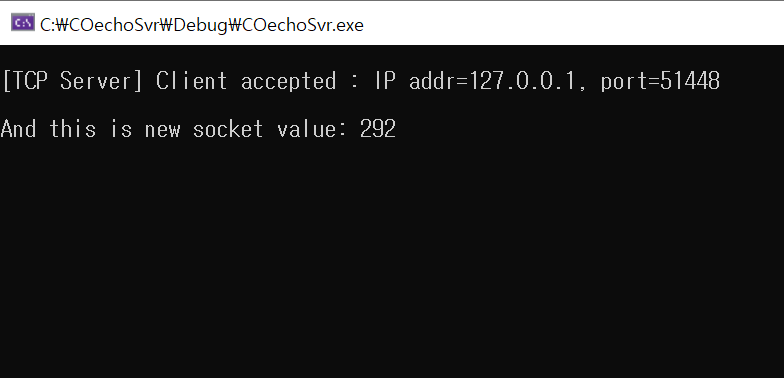
1. Server 구현

Client 를 listen 만 하는 listen socket 를 *socket(AF\_INET, SOCK\_STREAM, 0);* 를 통해서 만들어 주고, 특정 네트워크가 아닌 모든 인터페이스와의 연결을 가능하게 *bind(listen\_sock, (SOCKADDR\*)&serveraddr, sizeof(serveraddr));* 를 통해 bind시켜준후에 listen()를 실행한다. 이후에 *accept(listen\_sock, (SOCKADDR\*)&clientaddr, &addrlen);*를 통해 client 와의 data 교환을 다룰 client socket 를 생성한다. Client 가 처음에 보내는 메시지를 *recv(client\_sock, buf, BUFSIZE, 0);*를 통해서 받아 buf를 client 와 같은 구조체 형식으로 변환 후 parsing 해서 data 각각의 값을 구한다. Echo message를 보내기 위해서 client에서 온 입력 문자열과 문자열의 길이(byte)를 component 로 가지는 구조체를 생성후에 *send(client\_sock, (char\*)&sendMessage, 2+msglen-19+1, 0);* 를 통해 client 로 전송한다. 앞서 client 구현과 마찬가지로 client 에서 보내준 입력 문자열이 QUIT이라면 echo message를 send() 한 이후에 client 와의 데이터 교환 loop 를 나가고 socket 를 client 다음은 listen socket 순서로 close 한다.

1. 동작 결과
2. 동작환경

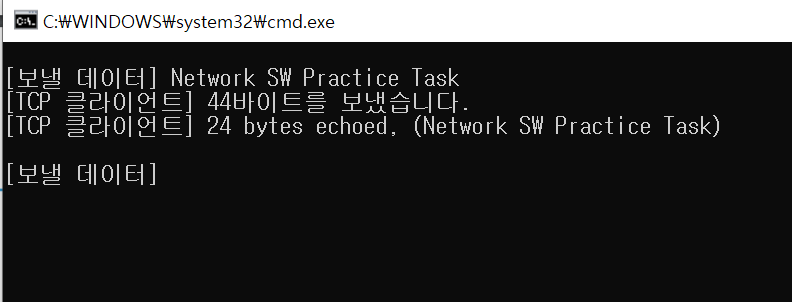
* OS환경 : windows 10
* 사용한 IDE : Visual studio 2019

1. 동작결과
2. Server 컴파일 -> Client 소스 컴파일 후에 Server 입장에서 Client socket를 생성한 후 (Server)

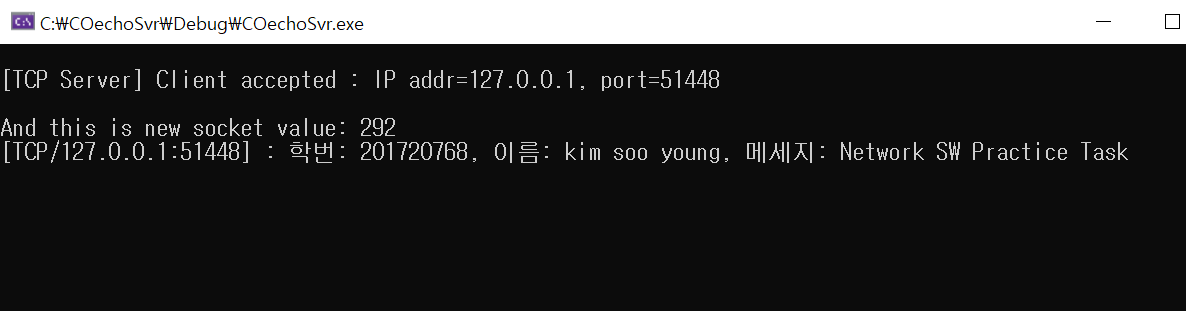


1. Client 컴파일 후에 [보낼 데이터]에 ‘Network SW Practice Task’ 입력 후에 Client 에선 보낸 문자열의 전체 byte 수를 출력, Server 에서의 echo 메시지를 받은 후에 메시지를 출력하고, Server에선 Client 로부터 받은 메시지를 parsing 해서 출력

(Client)

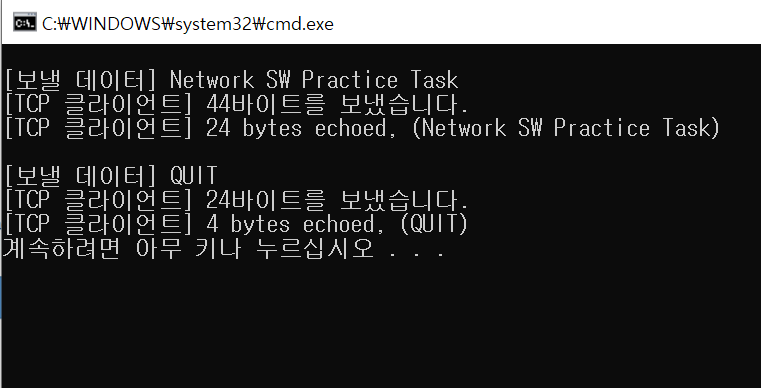


(Server)

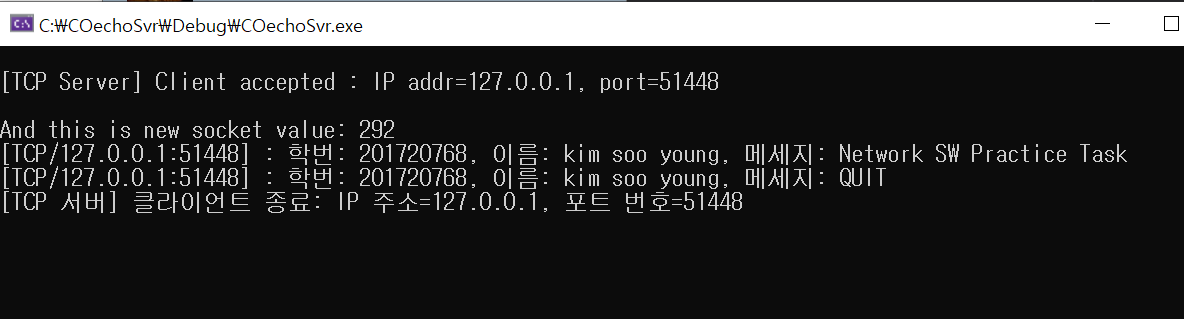


1. Client [보낼 데이터]에 ‘QUIT’입력 후 Server 가 echo message 보낸 이후에 client socket 종료, Client 입장에선 Server 에서의 echo message를 받고 socket close, 전체 종료.

(Client)



(Server)



1. 프로그램 내역

R01\_201720768\_COprogramming.zip 압축파일 안에

1. R01\_201720768\_COechoClient\_passing.cpp

Connection-oriented programming 에서의 client 부분 소스파일

1. R01\_201720768\_COechoSvr.cpp

Connection-oriented programming 에서의 server 부분 소스파일

* 소스파일만 압축
* 링커: ws2\_32.lib, 메니페스트 포함: 아니요, 특정 경고 사용 안함: 4996