

# HW 3 보고서

이름 : 정주영

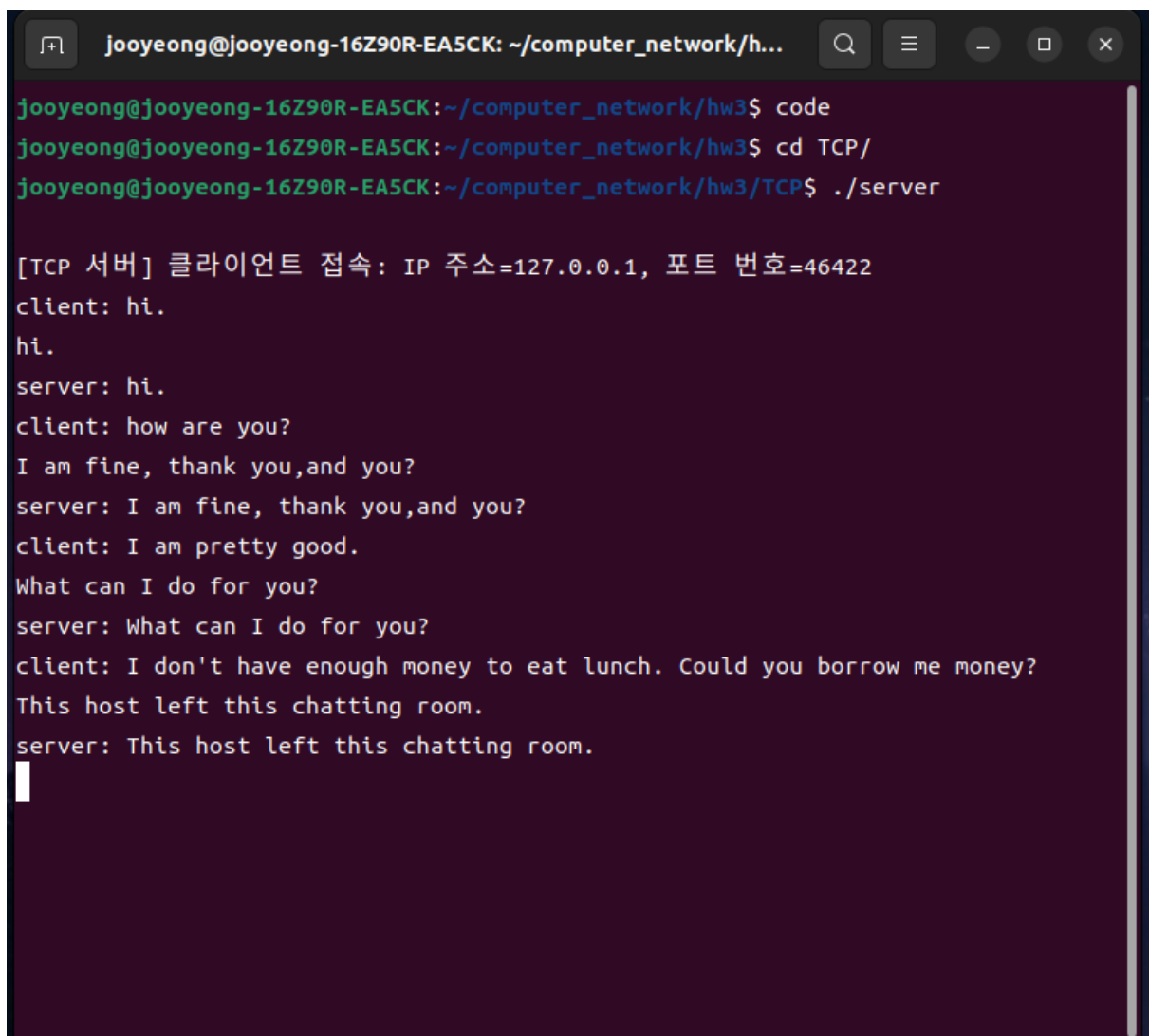
학번 : 2024317

컴퓨터네트워크 01분반

TCP/UDP를 이용한 server-client 채팅 프로그램 구현

## 1. TCP 환경에서 multi-thread를 이용한 server-client 채팅 프로그램 구현

- TCP server (TCPServer.c)



```
jooyeong@jooyeong-16Z90R-EA5CK: ~/computer_network/h...
jooyeong@jooyeong-16Z90R-EA5CK:~/computer_network/hw3$ code
jooyeong@jooyeong-16Z90R-EA5CK:~/computer_network/hw3$ cd TCP/
jooyeong@jooyeong-16Z90R-EA5CK:~/computer_network/hw3/TCP$ ./server

[TCP 서버] 클라이언트 접속: IP 주소=127.0.0.1, 포트 번호=46422
client: hi.
hi.
server: hi.
client: how are you?
I am fine, thank you,and you?
server: I am fine, thank you,and you?
client: I am pretty good.
What can I do for you?
server: What can I do for you?
client: I don't have enough money to eat lunch. Could you borrow me money?
This host left this chatting room.
server: This host left this chatting room.
```

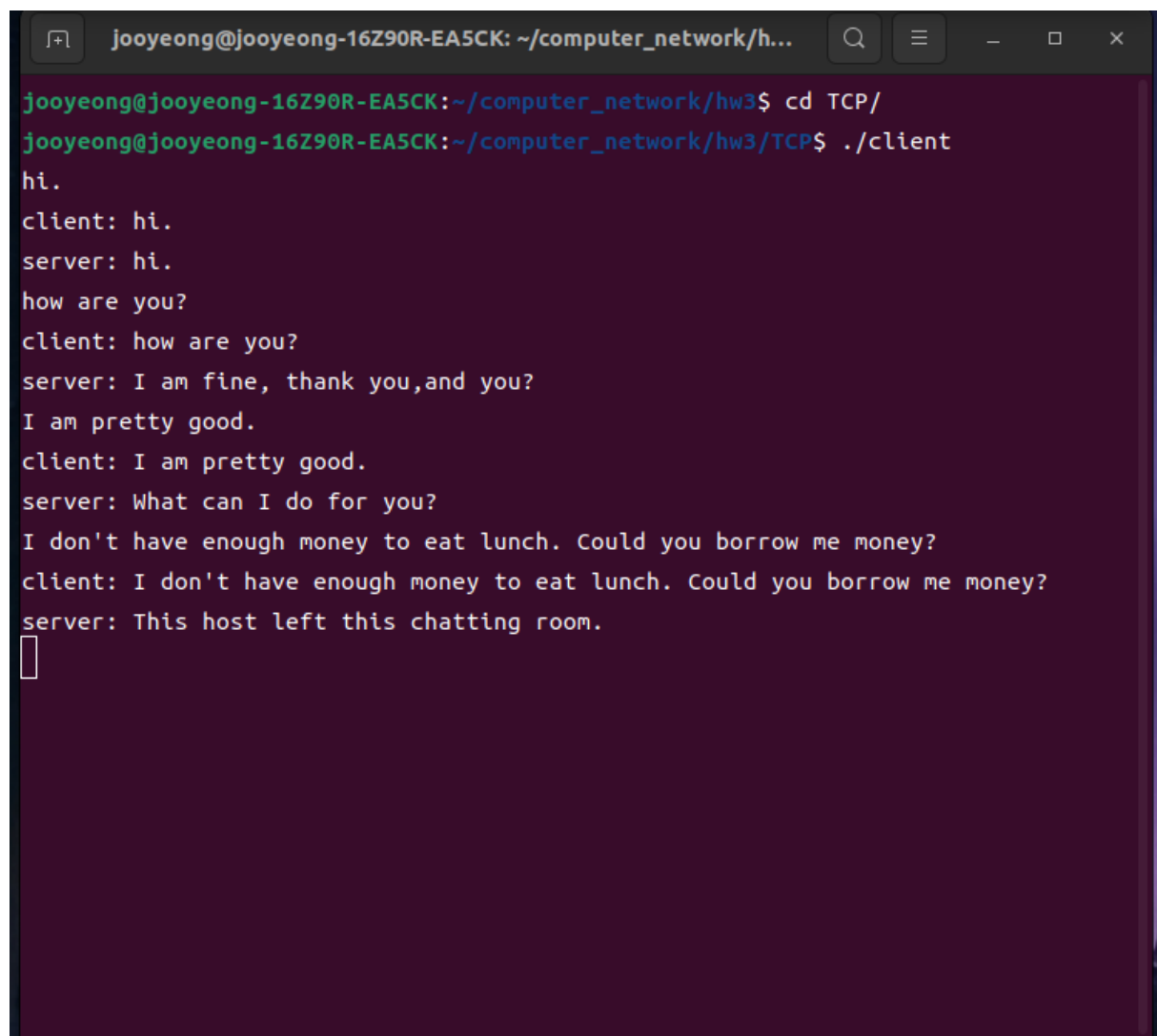
TCP 연결로 multi thread 를 이용한 server 이다. localhost 로 구현이 되어있다. 각각의 스레드가 sendFunc, recvFunc 을 실행 시켜 멀티 스레드로 하나의 스레드가 수신을 다른 하나가 송신 역할을 맡게 된다

sendFunc(tid1)에서 메세지 전송을 하게 되는데 TCP 연결 특성상 connect 가 되어있어 socket 만 있으면 send 함수를 쉽게 사용해서 메세지를 보낼 수 있다.

recvFunc(tid2)에서 메세지 수신을 하게 된다. 마찬가지로 connect 가 되어 있기 때문에 socket 만 있으면 쉽게 값을 받을 수 있다.

main 함수에서 서버 주소를 socket에 bind 함수를 사용해 지정한다. 그 후 listen 을 통해 connect 요청을 받으면 accept 로 연결을 수락하여 client의 주소를 받을 수 있다. 주소가 들어간 socket 은 스레드 실행 함수에 인자로 전달 된다. 이때 전달 인자는 포인터로 값을 전달해 주어야 한다.

- TCP client (TCPClient.c)



```
jooyeong@jooyeong-16Z90R-EA5CK: ~/computer_network/h...
jooyeong@jooyeong-16Z90R-EA5CK:~/computer_network/hw3$ cd TCP/
jooyeong@jooyeong-16Z90R-EA5CK:~/computer_network/hw3/TCP$ ./client
hi.
client: hi.
server: hi.
how are you?
client: how are you?
server: I am fine, thank you,and you?
I am pretty good.
client: I am pretty good.
server: What can I do for you?
I don't have enough money to eat lunch. Could you borrow me money?
client: I don't have enough money to eat lunch. Could you borrow me money?
server: This host left this chatting room.
█
```

TCP 연결로 multi thread 를 이용한 Client 이다. 마찬가지로 내부 host 로 구현이 되어있다. 각각의 쓰레드가 sendFunc, recvFunc 을 실행 시켜 멀티 쓰레드로 하나의 쓰레드가 수신을 다른 하나가 송신 역할을 맡게 된다

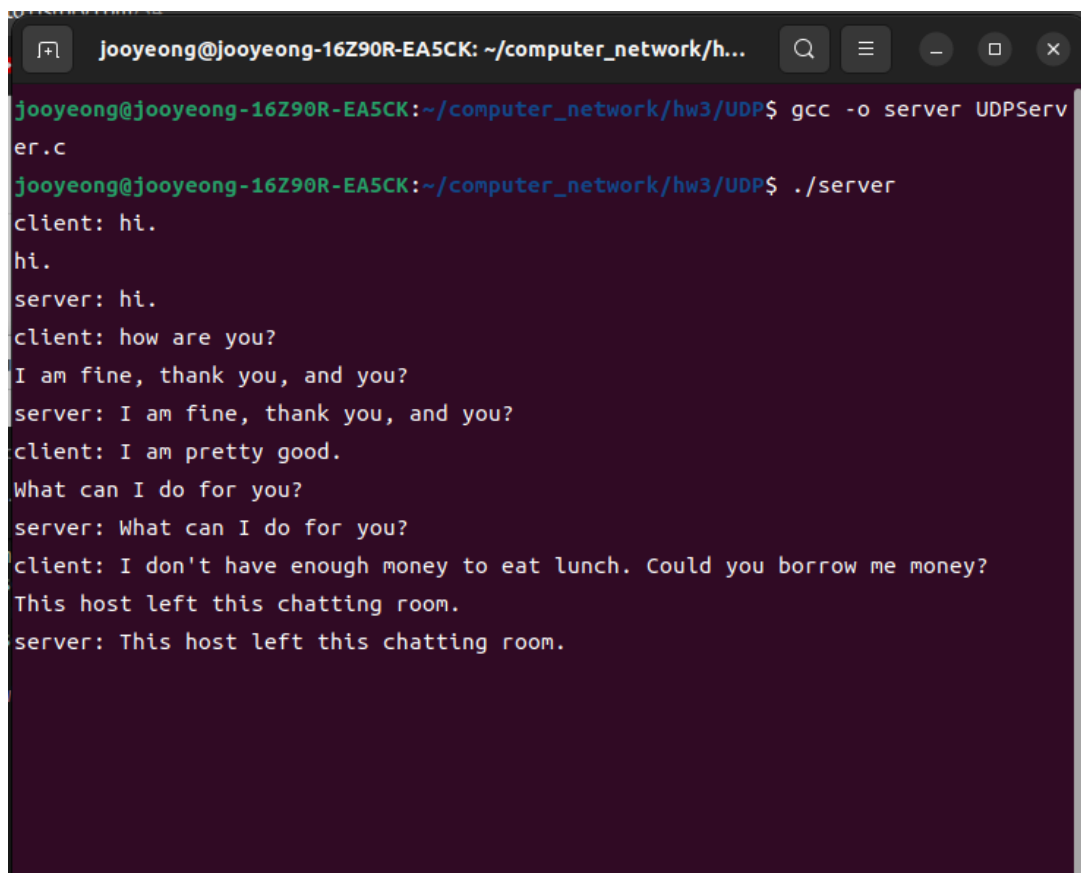
sendFunc(tid1)에서 메세지 전송을 하게 되는데 TCP 연결 특성상 connect 가 되어있어 socket 만 있으면 send 함수를 쉽게 사용해서 메세지를 보낼 수 있다.

recvFunc(tid2)에서 메세지 수신을 하게 된다. 마찬가지로 connect 가 되어 있기 때문에 socket 만 있으면 쉽게 값을 받을 수 있다.

main 함수에서 server 주소와 포트로 connect 요청을 보낸다. connect 가 되면 쓰레드를 생성한다. 주소가 들어간 socket 은 쓰레드 실행 함수에 인자로 전달 된다. 이때 전달 인자는 포인터로 값을 전달해 주어야 한다.

## 2. UDP 환경에서 multi-thread를 이용한 server-client 채팅 프로그램 구현

- UDP server (UDPServer.c)



```
jooyeong@jooyeong-16Z90R-EA5CK: ~/computer_network/h...
jooyeong@jooyeong-16Z90R-EA5CK:~/computer_network/hw3/UDP$ gcc -o server UDPServ
er.c
jooyeong@jooyeong-16Z90R-EA5CK:~/computer_network/hw3/UDP$ ./server
client: hi.
hi.
server: hi.
client: how are you?
I am fine, thank you, and you?
server: I am fine, thank you, and you?
client: I am pretty good.
What can I do for you?
server: What can I do for you?
client: I don't have enough money to eat lunch. Could you borrow me money?
This host left this chatting room.
server: This host left this chatting room.
```

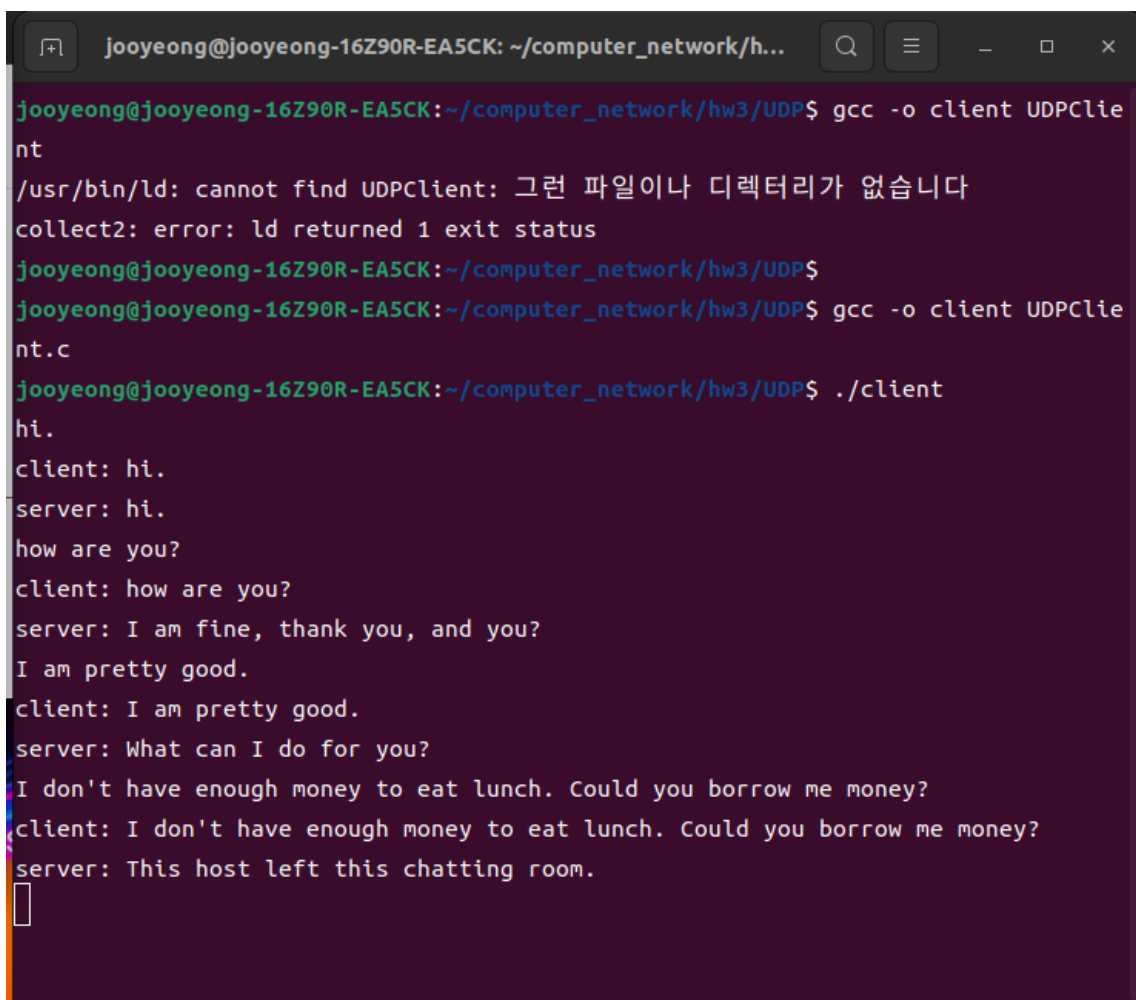
UDP 연결로 multi thread 를 이용한 Client 이다. 내부 host 로 구현이 되어있다. 각각의 스레드가 sendFunc, recvFunc 을 실행 시켜 멀티 스레드로 하나의 스레드가 수신을 다른 하나가 송신 역할을 맡게 된다. UDP 통신으로 멀티 스레드로 송, 수신을 구현 하였더니 포트가 충돌이 발생하여 Client 포트와 Server 포트를 나누어 구성했다.

sendFunc(tid1)에서 메세지 전송을 한다. UDP 통신을 사용하기 때문에 serveraddr 은 socket 에 있고 clientaddr 을 sendto()에 포함시켜야 한다.

recvFunc(tid2)에서 메세지 수신을 하게 된다. 마찬가지로 connect 가 되어 있기 때문에 socket 만 있으면 쉽게 값을 받을 수 있다. 마찬가지로 UDP 통신을 사용하기 때문에 recvfrom() 함수를 사용하여 client의 주소를 함께 넣어주었다.

main 함수에서 socket을 생성 해주고 serveraddr 로 bind 해주었다. 그 후 스레드를 생성하여 스레드 생성 실행 함수에 인자로 socket 을 넣어 보내준다.

- UDP client (UDPClient.)



```
jooyeong@jooyeong-16Z90R-EA5CK: ~/computer_network/h...
jooyeong@jooyeong-16Z90R-EA5CK:~/computer_network/hw3/UDP$ gcc -o client UDPClient.c
/usr/bin/ld: cannot find UDPClient: 그런 파일이나 디렉터리가 없습니다
collect2: error: ld returned 1 exit status
jooyeong@jooyeong-16Z90R-EA5CK:~/computer_network/hw3/UDP$
jooyeong@jooyeong-16Z90R-EA5CK:~/computer_network/hw3/UDP$ gcc -o client UDPClient.c
jooyeong@jooyeong-16Z90R-EA5CK:~/computer_network/hw3/UDP$ ./client
hi.
client: hi.
server: hi.
how are you?
client: how are you?
server: I am fine, thank you, and you?
I am pretty good.
client: I am pretty good.
server: What can I do for you?
I don't have enough money to eat lunch. Could you borrow me money?
client: I don't have enough money to eat lunch. Could you borrow me money?
server: This host left this chatting room.
█
```

UDP 연결로 multi thread 를 이용한 Client 이다. 내부 host 로 구현이 되어있다. 각각의 쓰레드가 sendFunc, recvFunc 을 실행 시켜 멀티 쓰레드로 하나의 쓰레드가 수신을 다른 하나가 송신 역할을 맡게 된다. UDP 통신으로 멀티 쓰레드로 송, 수신을 구현 하였더니 포트가 충돌이 발생하여 Client 포트와 Server 포트를 나누어 구성했다.

sendFunc(tid1)에서 메세지 전송을 한다. UDP 통신을 사용하기 때문에 serveraddr 은 socket 에 있고 clientaddr 을 sendto()에 포함시켜야 한다. 따라서 serveraddr 에 서버 주소와 port 번호를 포함하여 준다.

recvFunc(tid2)에서 메세지 수신을 하게 된다. 마찬가지로 connect 가 되어 있기 때문에 socket 만 있으면 쉽게 값을 받을 수 있다. 마찬가지로 UDP 통신을 사용하기 때문에 recvfrom() 함수를 사용하여 client의 주소를 함께 넣어주었다.

main 함수에서 socket을 생성 해주고 serveraddr 로 bind 해주었다. 그 후 쓰레드를 생성하여 쓰레드 생성 실행 함수에 인자로 socket 을 넣어 보내준다.

채팅 프로그램 구현 특성상 속도의 차이를 거의 체감할 수 없기 때문에 UDP 보다는 TCP로 통신을 하는 것이 연결만 하면 송수신이 간단해지는 점과 안정성의 측면에서 더 효율적일 것 같다고 생각한다.