

시스템 프로그래밍 실습

[FTP 3-2]

Class : D
Professor : 최상호 교수님
Student ID : 2020202092
Name : 강현민

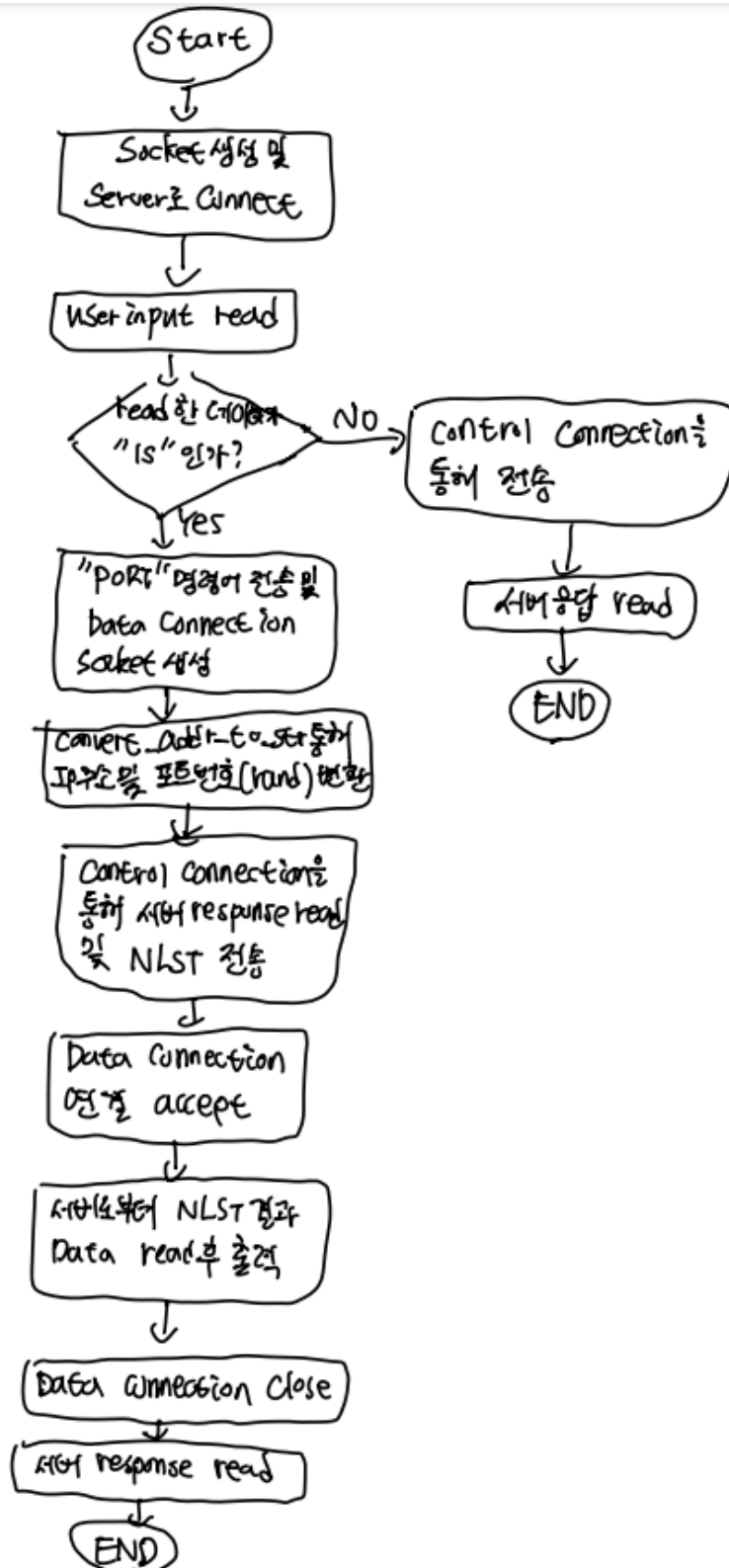
Introduction

이번 [FTP 3-2](#) 과제는 최종적으로 FTP 서버-클라이언트 구조를 구현하는 것을 목적으로 하는 프로젝트의 세번째 단계인 FTP service completion 의 두번째 단계인 Binary/ascii 와 dual connection 을 구현하는 것을 목적으로 한다. 이 때 dual connection 은 FTP 서버-클라이언트의 통신을 Control connection 과 Data connection 두가지의 연결로 통신하는 것을 뜻하며, Control connection 을 통해 클라이언트는 서버로 FTP command 를 전송하며, 1xx, 2xx 등의 numerical 응답을 서버로부터 전송받아 출력한다. 또한, Data connection 을 통해 서버는 클라이언트로 FTP 명령어에 대한 결과 데이터를 전송하여 출력한다. 이러한 FTP 서버-클라이언트 구조를 통해 ls 명령어에 대한 동작을 구현하는 것을 목적으로 한다.

Flow chart

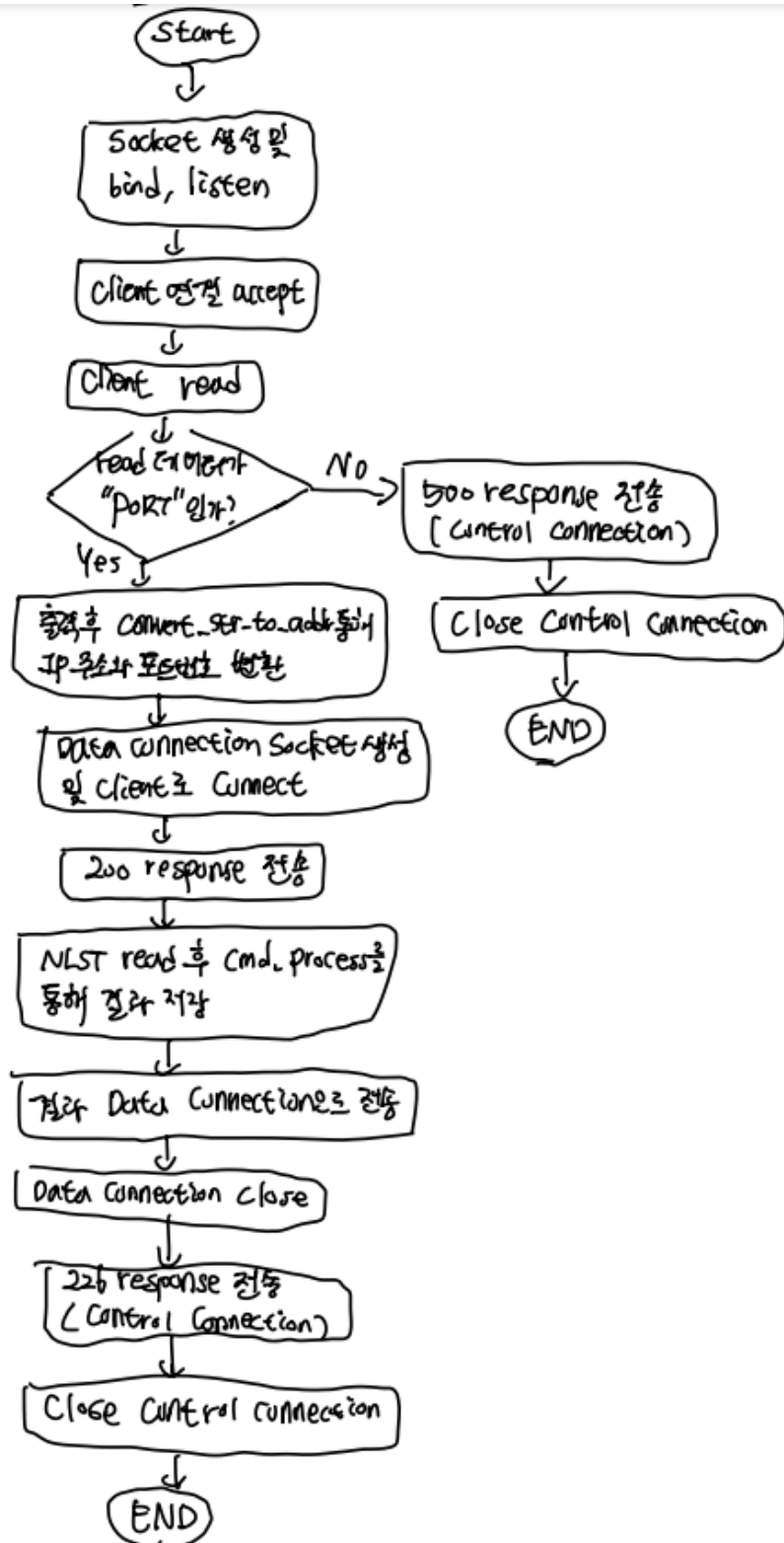
<cli.c>

<cli.c>



<srv.c>

<srv.c>



Pseudo code

<cli.c>

void main(int argc, char **argv)

 declare variables for communication with server

 Check if IP address and port number are provided

 Create socket

 set server address structure

 connect to the server

 read from user input

 if user input is equal to "ls"

 create data connection socket

 write "PORT" to the server

 read from server

 print message

 write "NLST" to the server

 read from server

 print message

 read data from server as data connection

 print data

 close data connection

 read from server

 print message

 else if user input is equal to "quit"

```
        convert quit to QUIT
        write "QUIT" to server
        read data from server
        if message is start with 221
            print message
            break;
    else
        write command to server
        close control connection
        exit(0);
close control connection
```

```
char* conver_addr_to_str(struct sockaddr_in *addr, const char *ip)
```

```
    conver IP address into four integers
    convert port number to binary
    return addr_str;
```

```
int create_data_socket(struct sockaddr_in *addr)
```

```
    create data connection socket
    set data connection address structure
    bind();
    listen();
    return sock;
```

<srv.c>

void main(int argc, char** argv)

 declare variables for connection with client

 Check if the port number is provided

 create control connection socket

 Initialize control connection address

 bind()

 listen()

 accept incoming connection from client

 read from client

 if read data is "PORT"

 convert string to address

 print port command

 create socket for data connection

 write 200 response to client

 read from client

 cmd_process(buff, result_buff);

 if read data is "NLST"

 write 150 response to client

 print message

 write result of NLST to the client

 write 226 response to client

```
        print message
    else
        write buffer to the client
else if command is "QUIT"
    if there are arguments
        print error message
        write error message
        continue;
    print 221 ack
    write message to client
else
    write 500 response to client
    print message
    close control connection
    exit(1);
```

```
close control connection
exit(1);
```

```
void cmd_process(char *buff, char* result_buff)
```

```
    if command is "NLST"
        if argc > 1
            write 501 response into result_buff
        else NLST(argv, 0, 0, result_buff);
```


else

write 500 response into result_buff;

void NLST(char *argv[128], int aflag, int lflag, char *result_buff)

char *directory = current directory

if there is no such directory

write 550 response into result_buff

if the directory cannot be accessed

write 550 response into result_buff

while readdir(directory) != NULL

namelist[i++] = entry;

bubblesort(namelist);

for (j=0; j<l ;j++)

write namelist[j]->d_name into the result_buff;

char *convert_str_to_addr (char* str, int *port)

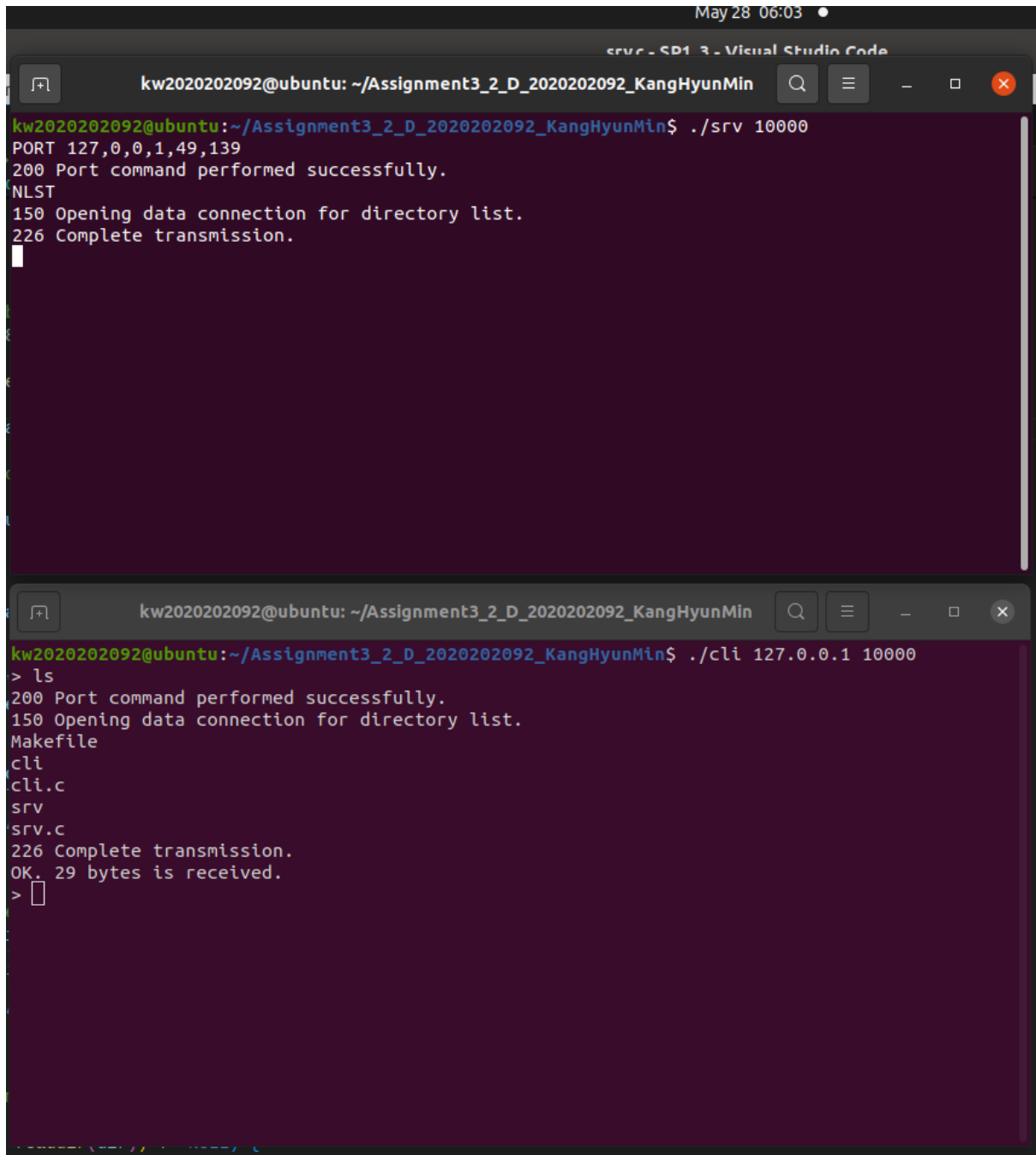
convert four integer to IP address

convert two 8-bit binary to port number (decimal)

return addr_str;

결과화면

<ls 를 입력한 경우>



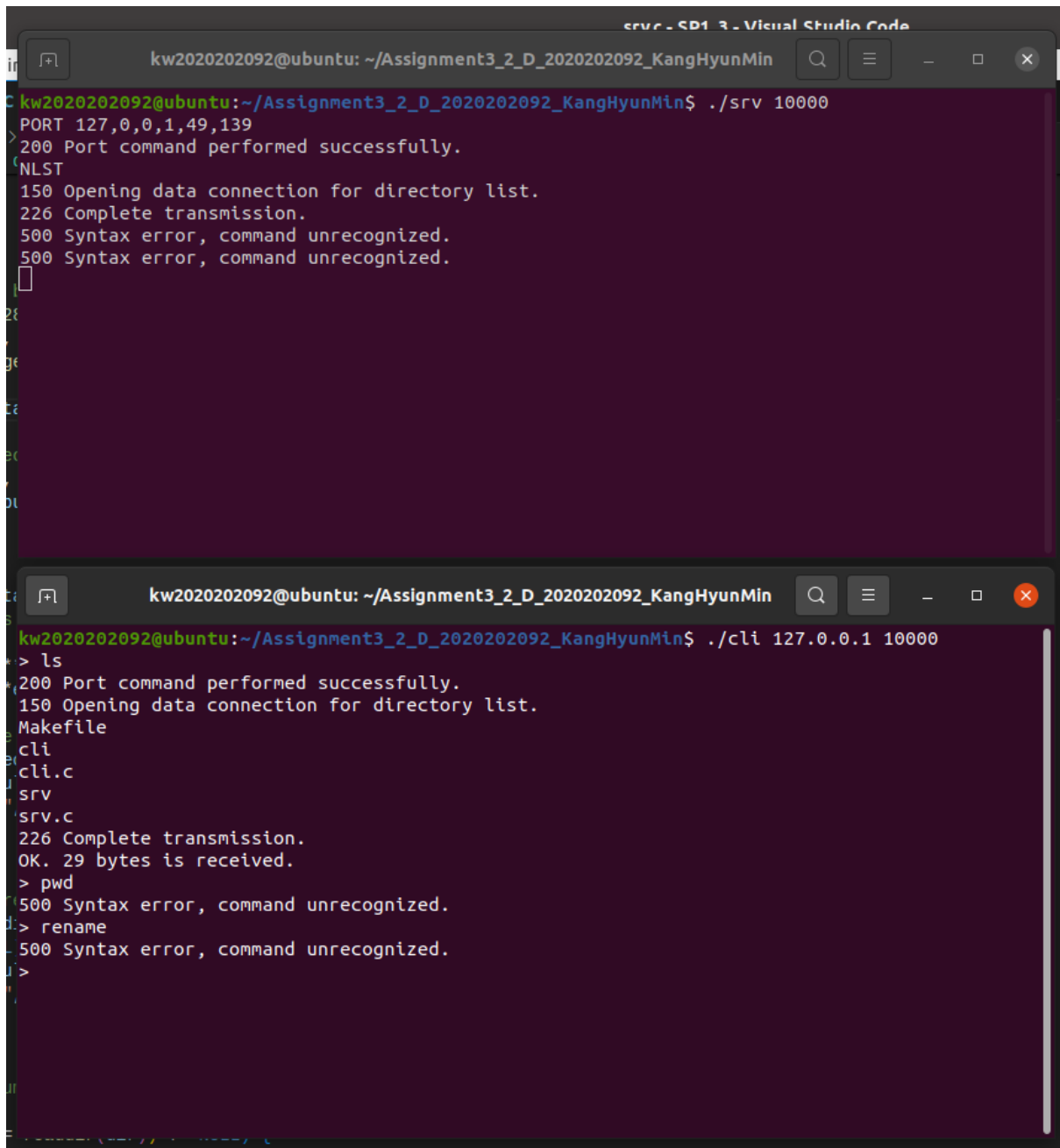
```
May 28 06:03 •
srcv - SD1 3 - Visual Studio Code
kw2020202092@ubuntu: ~/Assignment3_2_D_2020202092_KangHyunMin
kw2020202092@ubuntu:~/Assignment3_2_D_2020202092_KangHyunMin$ ./srv 10000
PORT 127,0,0,1,49,139
200 Port command performed successfully.
NLST
150 Opening data connection for directory list.
226 Complete transmission.

kw2020202092@ubuntu:~/Assignment3_2_D_2020202092_KangHyunMin$ ./cli 127.0.0.1 10000
> ls
200 Port command performed successfully.
150 Opening data connection for directory list.
Makefile
cli
cli.c
srv
srv.c
226 Complete transmission.
OK. 29 bytes is received.
>
```

클라이언트에서 사용자 입력으로 ls 를 입력한 경우 클라이언트에서 PORT command 를 서버로 전송하고, ip 주소인 127.0.0.1 과 랜덤 포트번호가 ,로 구분되어 서버에 출력되는 것을 확인할 수 있다. 이 때, 포트번호는 10001 ~ 30000 사이의 임의의 수로 이 수를 2 진수로 나타낸 후, 상위 8 비트와 하위 8 비트로 잘라 10 진수로 나타낸 값이다. 즉, 61, 175 는 포트번호 15791 를 나타낸다. 따라서 PORT 명령어에 대한 ack 를 전송하고,

서버와 클라이언트에 출력되는 것을 확인할 수 있다. 그 후 data connection 을 open 한 뒤 해당 ack 가 서버와 클라이언트에 출력되고, 클라이언트에 NLST 에 대한 결과가 출력되는 것을 확인할 수 있다. 이후 이에 대한 ack 가 서버와 클라이언트에 출력되고 클라이언트에 전송받은 데이터의 바이트 수가 출력된다.

<정의되지 않은 명령어>

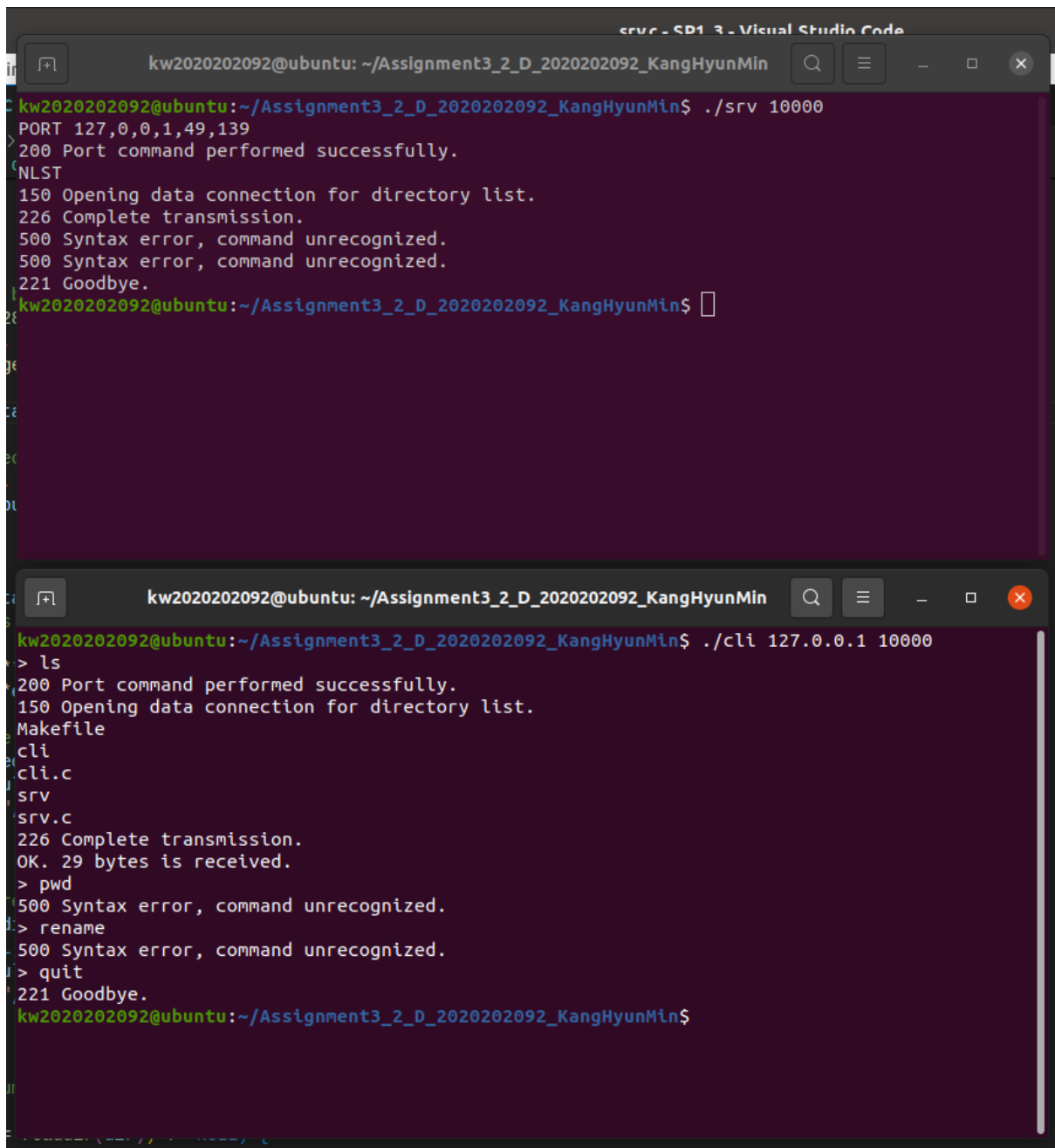


```
kw2020202092@ubuntu: ~/Assignment3_2_D_2020202092_KangHyunMin
kw2020202092@ubuntu:~/Assignment3_2_D_2020202092_KangHyunMin$ ./srv 10000
PORT 127,0,0,1,49,139
> 200 Port command performed successfully.
(NLST
150 Opening data connection for directory list.
226 Complete transmission.
500 Syntax error, command unrecognized.
500 Syntax error, command unrecognized.
[]

kw2020202092@ubuntu: ~/Assignment3_2_D_2020202092_KangHyunMin
kw2020202092@ubuntu:~/Assignment3_2_D_2020202092_KangHyunMin$ ./cli 127.0.0.1 10000
> ls
200 Port command performed successfully.
150 Opening data connection for directory list.
Makefile
cli
cli.c
srv
srv.c
226 Complete transmission.
OK. 29 bytes is received.
> pwd
500 Syntax error, command unrecognized.
> rename
500 Syntax error, command unrecognized.
>
```

정의되지 않은 명령어가 입력된 경우 이에 해당하는 ack 가 서버와 클라이언트에 출력되는 것을 확인할 수 있다.

<quit 을 입력한 경우>



```
kw2020202092@ubuntu: ~/Assignment3_2_D_2020202092_KangHyunMin
kw2020202092@ubuntu:~/Assignment3_2_D_2020202092_KangHyunMin$ ./srv 10000
PORT 127,0,0,1,49,139
> 200 Port command performed successfully.
(NLST
150 Opening data connection for directory list.
226 Complete transmission.
500 Syntax error, command unrecognized.
500 Syntax error, command unrecognized.
221 Goodbye.
kw2020202092@ubuntu:~/Assignment3_2_D_2020202092_KangHyunMin$

kw2020202092@ubuntu: ~/Assignment3_2_D_2020202092_KangHyunMin
kw2020202092@ubuntu:~/Assignment3_2_D_2020202092_KangHyunMin$ ./cli 127.0.0.1 10000
> ls
200 Port command performed successfully.
150 Opening data connection for directory list.
Makefile
cli
cli.c
srv
srv.c
226 Complete transmission.
OK. 29 bytes is received.
> pwd
500 Syntax error, command unrecognized.
> rename
500 Syntax error, command unrecognized.
> quit
221 Goodbye.
kw2020202092@ubuntu:~/Assignment3_2_D_2020202092_KangHyunMin$
```

클라이언트의 사용자 입력으로 quit 을 입력한 경우 서버와 클라이언트에 이에 해당하는 ack 가 출력된 후 클라이언트가 종료되는 것을 확인할 수 있다.

고찰

이번 [FTP 3-2](#) 과제를 통해 FTP 서버-클라이언트에서 Control connection 과 Data connection 으로 분리하여 통신하는 구조를 구현하는 방법에 대해 학습할 수 있었다. 또한 이렇게 dual connection 으로 분리한 구조의 이점에 대해 학습할 수 있었으며, FTP 의 numerical reply 를 사용함으로써 숫자에 따른 응답의 의미를 학습할 수 있었다. 또한 이번 과제 제출 후 지금까지 구현했던 로그인 및 다른 명령어들을 전부 통합하여 최종적인 FTP 서버-클라이언트 구조를 구현하는 방법에 대해 고민해봐야겠다는 생각이 들었다.

Reference