

시스템 프로그래밍 실습

[Assignment 2-1]

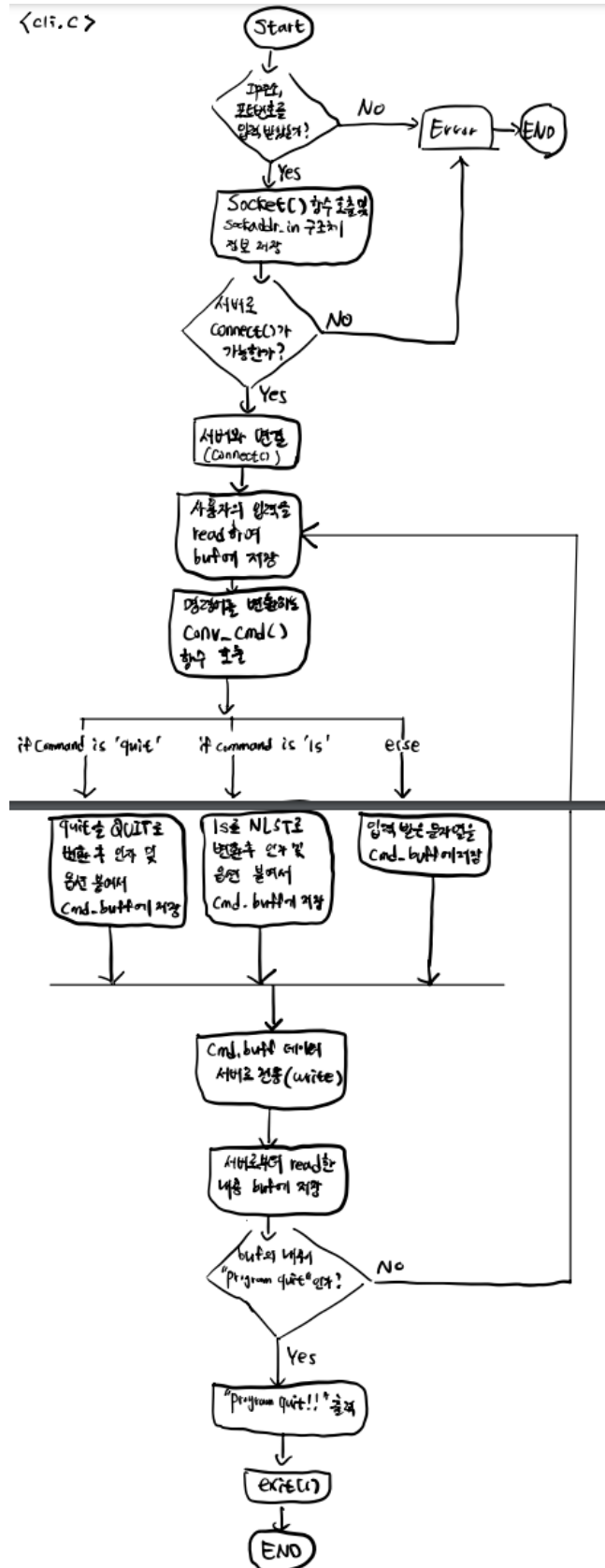
Class : D
Professor : 최상호 교수님
Student ID : 2020202092
Name : 강현민

Introduction

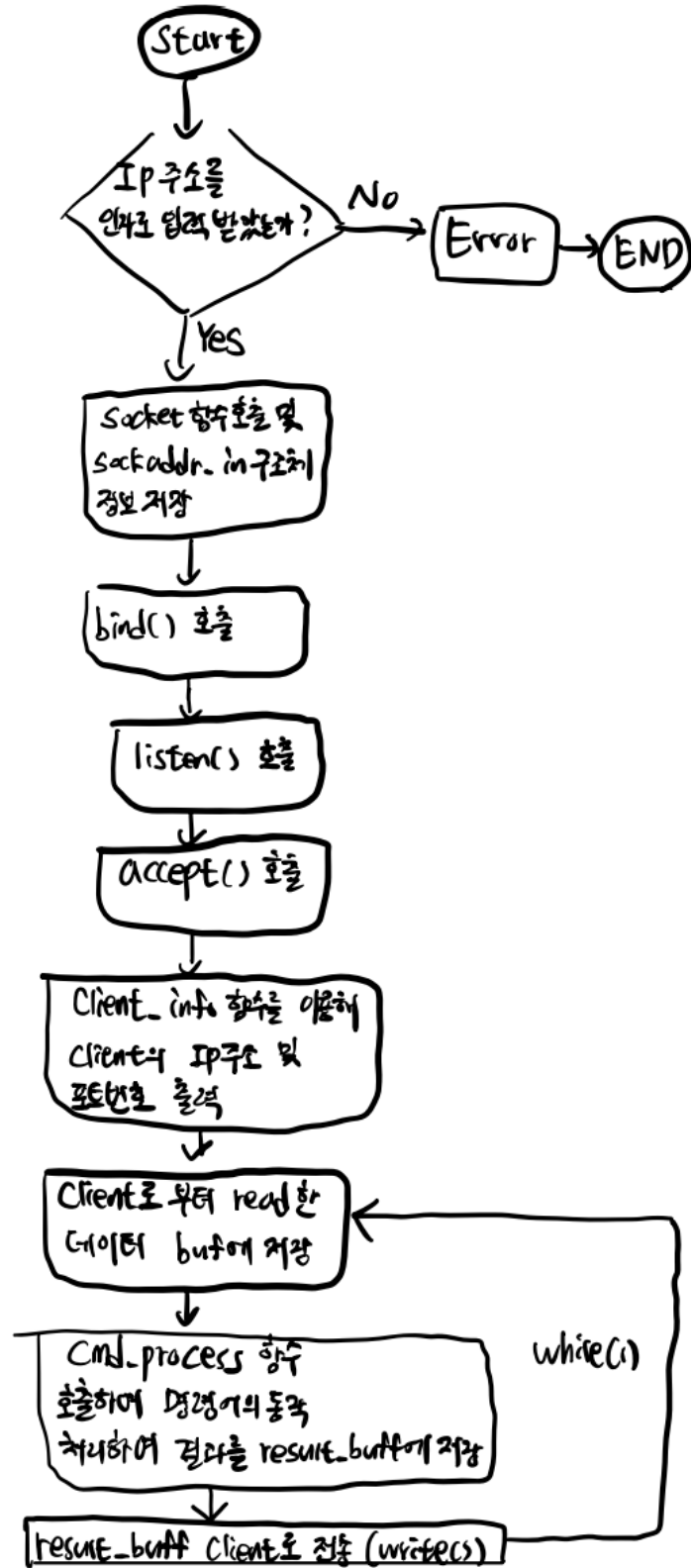
이번 [FTP 2-1](#) 과제는 최종적으로 FTP 서버를 구현하는 프로젝트의 두번째 과정인 소켓 구현 중 소켓 프로그래밍을 수행하는 단계로, 소켓 프로그래밍을 사용하여 FTP 환경을 시뮬레이션 하는 간단한 클라이언트-서버 구조를 구현하는 것을 목표로 한다. 클라이언트에서는 socket, connect, write 와 같은 소켓 작업을 하여 서버에 연결하는 기능을 구현하며, 사용자가 입력하는 명령어인 ls 와 quit 은 각각 NLST 와 QUIT FTP 명령어로 변환되어 서버로 전송한다. 서버 측에서는 클라이언트의 연결 요청을 수락하고 read 를 통해 전송받은 FTP 명령어를 처리하는 역할을 담당한다. 이를 위해 socket, bind, listen, accept 와 같은 소켓 함수를 사용한다. 연결에 성공하면 서버는 클라이언트의 IP 주소와 포트 번호를 inet_ntoa, ntohs 와 같은 함수를 사용하여 변환하여 사용한다. 즉, 서버는 클라이언트로부터 FTP 명령어를 수신하고 이를 처리한 후 결과를 다시 클라이언트로 write 하여 출력한다. 이러한 구현들을 통해 클라이언트와 서버 간의 네트워킹 및 데이터 전송의 기본 동작을 익히는 것을 목표로 한다.

Flow chart

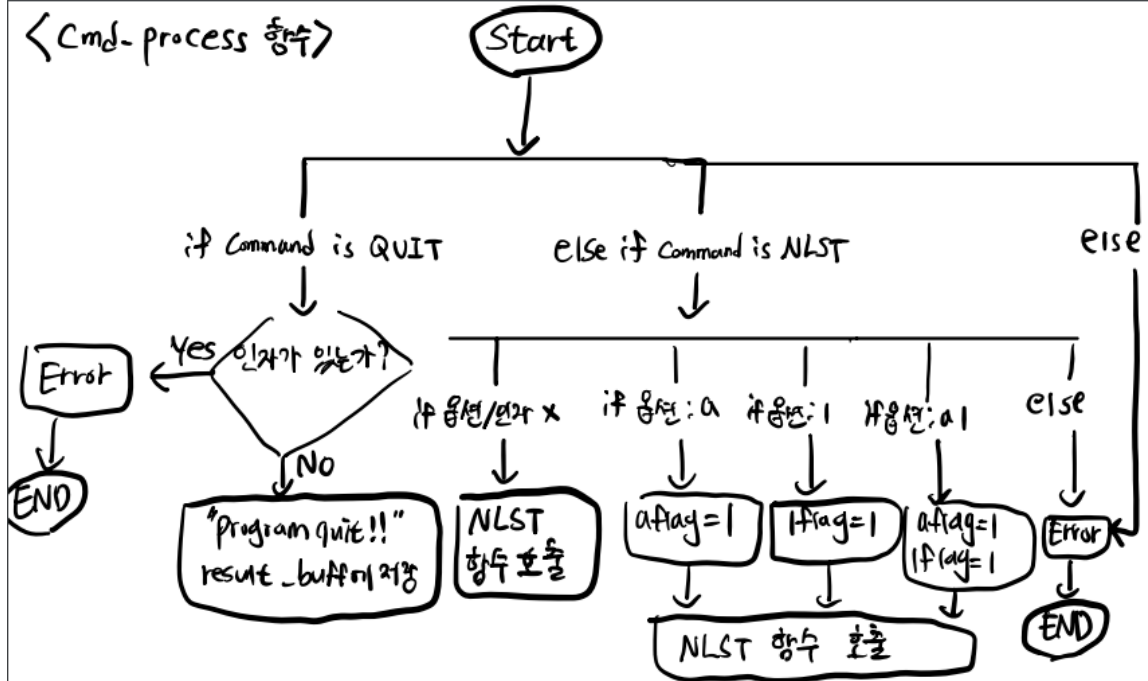
<cli.c>



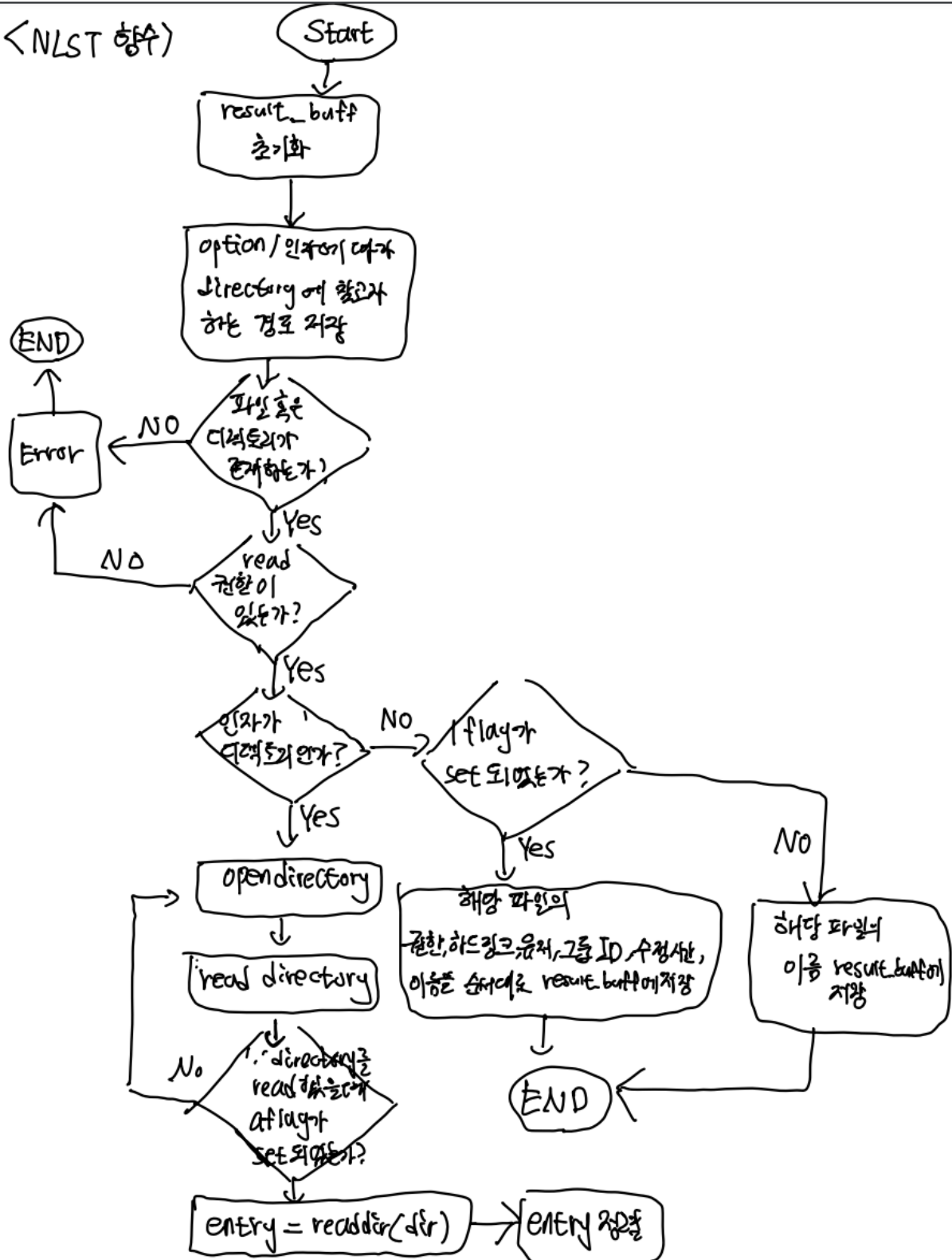
<Srv.C>



<Cmd-process 함수>



<NLST 함수>



Pseudo code

<cli.c>

```
Void conv_cmd(const char *buff, char *cmd_buff) {  
    If(buff is "quit")        store "QUIT" into cmd_buff;  
        If(there is argument or option)  
            Strcat all argument to cmd_buff;  
    Else if(buff is "ls")      store "NLST" into cmd_buff;  
        If(there is argument or option)  
            Strcat all argument to cmd_buff;  
    Else copy buff into cmd_buff;  
  
Int main(int argc, char* argv[]){  
    If(argc != 3) {  
        Print "IP address and Port number required."  
        Return -1;  
    }  
  
    Store argv[1] into char *ip_addr  
    Store argv[2] into int port  
  
    Declare int socket_fd, len;  
    Declare struct sockaddr_in server_addr;  
    Declare buf[BUFFSIZE]  
    Declare cmd_buff[BUFFSIZE]
```

```
socket_fd = socket(PF_INET, SOCK_STREAM, 0);
```

```
reset data of buf
```

```
reset data of server_addr
```

```
store AF_INET and IP address and port into server_addr
```

```
connect(socket_fd, (struct sockaddr*)&server_addr, sizeof(server_addr));
```

```
while(read user input)
```

```
    conv_cmd(buf, cmd_buff);
```

```
    write cmd_buff to client
```

```
    while(read data from server)
```

```
        if (buf is "Program quit!!")
```

```
            print buf
```

```
            exit(1);
```

```
        print buf
```


<SRV.C>

```
Void client_info(const struct sockaddr_in *cliaddr){
```

```
    Print client IP address and port number
```

```
Void cmd_process(const char* buff, char* result_buff) {
```

```
    Declare and reset int aflag =0, lflag = 0, char* directory = NULL;
```

```
    Tokenize buff based on space into argv
```

```
    If argv[0] == QUIT
```

```
        If(there is no argument){
```

```
            Store error message into result_buff
```

```
            Return;
```

```
        }
```

```
        Store "Program quit!!" into result_buff
```

```
    Else if argv[0] == NLST
```

```
        if(there is no argument)
```

```
            write NLST
```

```
            call NLST(arg, 0, 0)
```

```
        else if there is option
```

```
            if option is -a
```

```
                aflag++;
```

```
                write NLST -a
```

```
call NLST(arg, aflag, lflag);
```

```
else if option is -l
```

```
lflag++;
```

```
write NLST -l
```

```
call NLST(arg, aflag, lflag);
```

```
else if option is -al or -la
```

```
aflag++;
```

```
lflag++;
```

```
write NLST -al
```

```
call NLST(arg, aflag, lflag);
```

```
else write error
```

```
void NLST(char* argv[BUFFSIZE/2], int aflag, int lflag, char* result_buff) {
```

```
if there is no argument
```

```
directory = current directory
```

```
else if option is inputted
```

```
directory = arg[2];
```

```
else if option is not inputted
```

```
directory = arg[1];
```

```
if arg is directory
```

```
opendir(directory)
```

```
while readdir(dir) != NULL
```

```
readdir(dir)
```

```
store datas into namelist
```

```
if aflag == 1
    write file type / permission
    write number of hard links
    write owner name
    write group name
    write file size
    write file modification time
    write file or directory name
else write file name
```

else

```
readdir(dir)
store datas into namelist
```

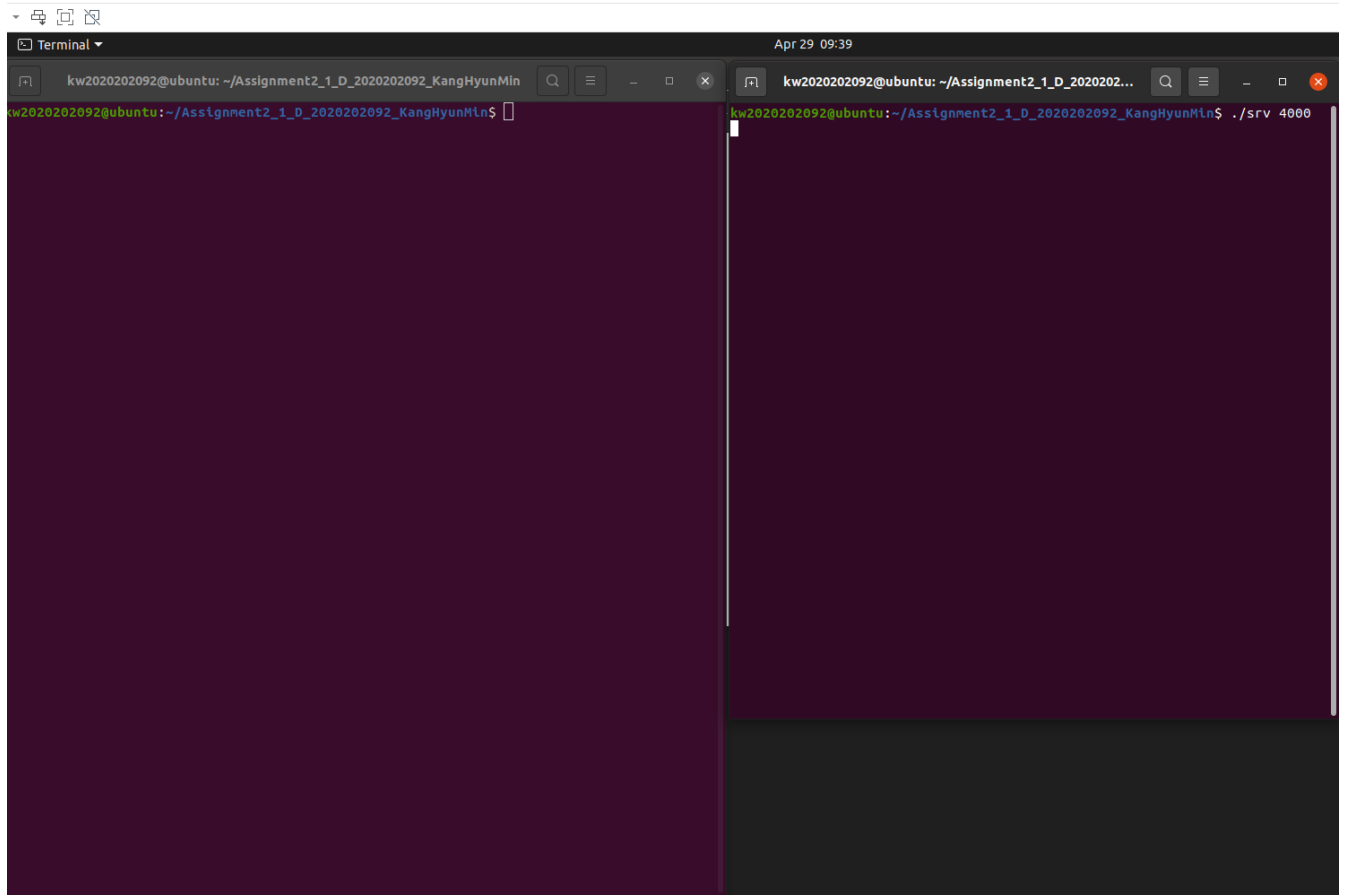
```
if aflag == 1
    write file type / permission
    write number of hard links
    write owner name
    write group name
    write file size
    write file modification time
    write file or directory name
else write file name
```

```
int main (int argc, char* argv[]) {  
    if(there is no two arguments)  
        print error  
        return -1;  
  
    declare int port and store argv[1] into port  
    declare int socket_fd, client_fd, len, len_out;  
  
    socket_fd = socket(PF_INET, SOCK_STREAM, 0);  
  
    store PF_INET and INADDR_ANY and port into server_addr  
  
    bind(socket_fd, (struct sockaddr*) &server_addr, sizeof(server_addr);  
  
    listen(socket_fd, 5);  
  
    while(1) {  
        client_fd = accept(socket_fd, (struct sockaddr*)&client_addr, &len);  
        call client_info to print client IP address and port number  
        while(read data from client) {  
            call cmd_process;  
            print result_buff  
        }  
    }  
}
```

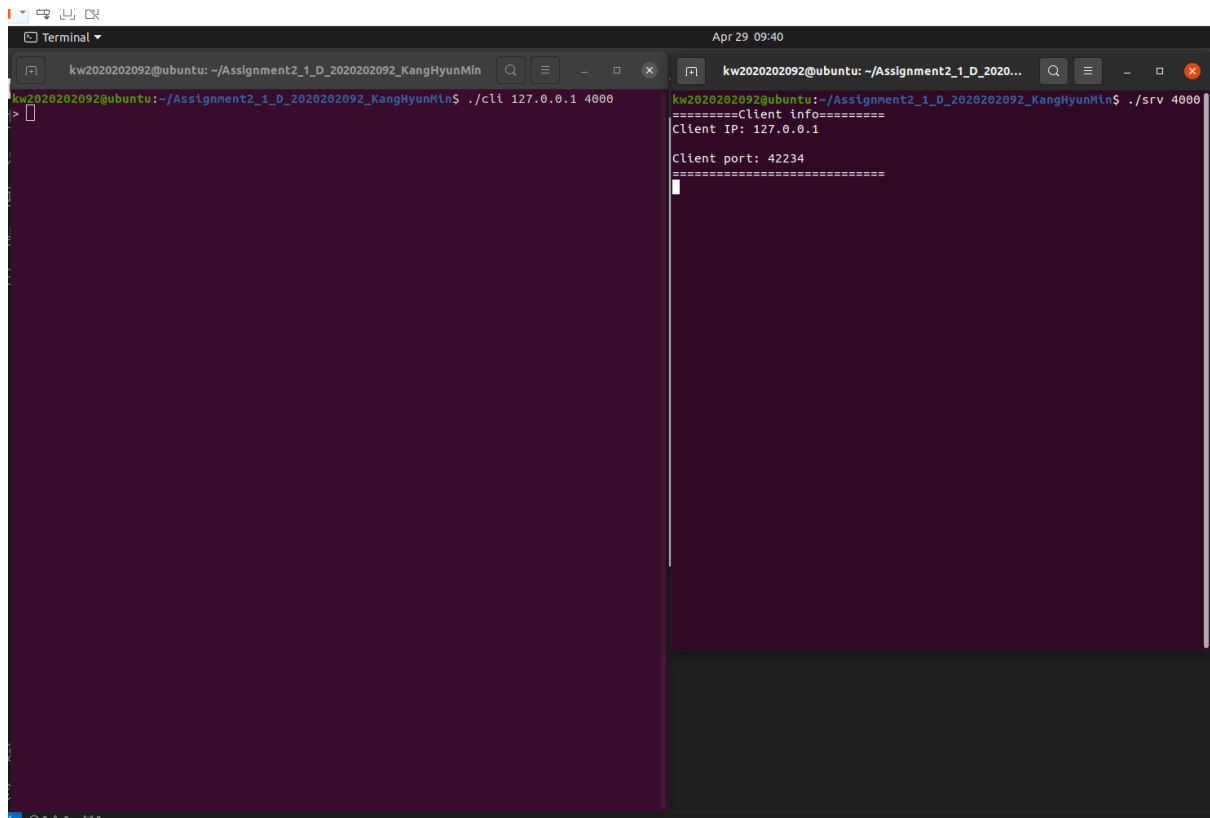
Close client_fd

Return 0;

결과화면



먼저 위와 같이 포트 번호 4000 을 인자로 하여 서버를 실행시킨다



```
kw2020202092@ubuntu: ~/Assignment2_1_D_2020202092_KangHyunMin
kw2020202092@ubuntu:~/Assignment2_1_D_2020202092_KangHyunMin$ ./cli 127.0.0.1 4000
>

kw2020202092@ubuntu:~/Assignment2_1_D_2020202092_KangHyunMin$ ./srv 4000
=====Client info=====
Client IP: 127.0.0.1
Client port: 42234
=====
```

그 후 위와 같이 IP 주소를 127.0.0.1, 그리고 포트 번호는 서버와 동일하게 4000 을 인자로 하여 클라이언트를 실행시킨 결과 클라이언트가 서버와 연결되고, 서버에 연결된 클라이언트의 정보(IP 주소, 포트 번호)가 서버에 출력되는 것을 확인할 수 있다.

```
kw2020202092@ubuntu: ~/Assignment2_1_D_2020202092_KangHyunMin
> ls
Makefile      cli      output/  srv
srv.c         test_dir/ testdir/
> ls -a
./      ../      .swp     Makefile  cli
cli.c   output/  srv      srv.c     test_dir/
testdir/
> ls -l
-rw-rw-r-- 1 kw2020202092 kw2020202092 55 Apr 27 06:40 Makefile
-rwxrwxr-x 1 kw2020202092 kw2020202092 17344 Apr 29 09:24 cli
-rwxrwxr-x 1 kw2020202092 kw2020202092 2508 Apr 29 09:17 cli.c
-rwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 28 05:32 output/
-rwxrwxr-x 1 kw2020202092 kw2020202092 22760 Apr 29 09:24 srv
-rwxrwxr-x 1 kw2020202092 kw2020202092 13976 Apr 29 09:24 srv.c
-rwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 29 09:25 test_dir/
-rwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 27 23:38 testdir/

> ls -al
drwxrwxr-x 5 kw2020202092 kw2020202092 4096 Apr 29 09:25 ./
drwxr-xr-x 21 kw2020202092 kw2020202092 4096 Apr 29 09:38 ../
-rw-r--r-- 1 kw2020202092 kw2020202092 12288 Apr 28 05:57 .swp
-rw-rw-r-- 1 kw2020202092 kw2020202092 55 Apr 27 06:40 Makefile
-rwxrwxr-x 1 kw2020202092 kw2020202092 17344 Apr 29 09:24 cli
-rwxrwxr-x 1 kw2020202092 kw2020202092 2508 Apr 29 09:17 cli.c
drwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 28 05:32 output/
-rwxrwxr-x 1 kw2020202092 kw2020202092 22760 Apr 29 09:24 srv
-rwxrwxr-x 1 kw2020202092 kw2020202092 13976 Apr 29 09:24 srv.c
drwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 29 09:25 test_dir/
drwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 27 23:38 testdir/

> ls -al test_dir
drwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 29 09:25 ./
drwxrwxr-x 5 kw2020202092 kw2020202092 4096 Apr 29 09:25 ../
-rw-rw-r-- 1 kw2020202092 kw2020202092 0 Apr 29 09:25 fileA
-rw-rw-r-- 1 kw2020202092 kw2020202092 0 Apr 29 09:25 fileB

> ls -e
Error: invalid option
> ls ./not_exist_dir
Error: No such file or directory
>
```

```
kw2020202092@ubuntu: ~/Assignment2_1_D_2020202092_KangHyunMin
=====Client info=====
Client IP: 127.0.0.1

Client port: 42234
=====
NLST
NLST -a
NLST -l
NLST -al
NLST -al test_dir
NLST -e
NLST ./not_exist_dir
[]
```

그 후 위와 같이 ls 명령어를 입력하는데, 먼저 ls를 클라이언트에서 입력한 결과 서버에서 해당 입력의 FTP 명령어인 NLST가 출력되고, 이에 대한 결과가 클라이언트에 출력되는 것을 확인할 수 있다. 그리고 이와 마찬가지로 ls -a, ls -l, ls -al을 입력했을 때 동일하게 서버에 해당 명령어에 대한 FTP 명령어가 서버에 출력되고, 해당 명령어에 대한 결과가 클라이언트로 출력되는 것을 확인할 수 있고, ls -al test_dir의 결과 또한 마찬가지로 의도대로 출력되는 것을 확인할 수 있다. 그리고 ls의 옵션으로 정의되지 않은 옵션을 입력하거나 존재하지 않는 경로를 인자로 입력하는 경우에 에러 메시지가 클라이언트로 정상적으로 출력되는 것을 확인할 수 있다.


```
kw2020202092@ubuntu: ~/Assignment2_1_D_2020202092_KangHyunMin
> ls
Makefile      cli      output/  srv
srv.c         test_dir/ testdir/
> ls -a
./      ../      .swp      Makefile  cli
cli.c   output/  srv      srv.c     test_dir/
testdir/
> ls -l
-rw-rw-r-- 1 kw2020202092 kw2020202092 55 Apr 27 06:40 Makefile
-rwxrwxr-x 1 kw2020202092 kw2020202092 17344 Apr 29 09:24 cli
-rwxrwxr-x 1 kw2020202092 kw2020202092 2508 Apr 29 09:17 cli.c
-rwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 28 05:32 output/
-rwxrwxr-x 1 kw2020202092 kw2020202092 22760 Apr 29 09:24 srv
-rwxrwxr-x 1 kw2020202092 kw2020202092 13976 Apr 29 09:24 srv.c
-rwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 29 09:25 test_dir/
-rwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 27 23:38 testdir/

> ls -al
drwxrwxr-x 5 kw2020202092 kw2020202092 4096 Apr 29 09:25 ./
drwxr-xr-x 21 kw2020202092 kw2020202092 4096 Apr 29 09:38 ../
-rw-r--r-- 1 kw2020202092 kw2020202092 12288 Apr 28 05:57 .swp
-rw-rw-r-- 1 kw2020202092 kw2020202092 55 Apr 27 06:40 Makefile
-rwxrwxr-x 1 kw2020202092 kw2020202092 17344 Apr 29 09:24 cli
-rwxrwxr-x 1 kw2020202092 kw2020202092 2508 Apr 29 09:17 cli.c
-rwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 28 05:32 output/
-rwxrwxr-x 1 kw2020202092 kw2020202092 22760 Apr 29 09:24 srv
-rwxrwxr-x 1 kw2020202092 kw2020202092 13976 Apr 29 09:24 srv.c
drwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 29 09:25 test_dir/
drwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 27 23:38 testdir/

> ls -al test_dir
drwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 29 09:25 ./
drwxrwxr-x 5 kw2020202092 kw2020202092 4096 Apr 29 09:25 ../
-rw-rw-r-- 1 kw2020202092 kw2020202092 0 Apr 29 09:25 fileA
-rw-rw-r-- 1 kw2020202092 kw2020202092 0 Apr 29 09:25 fileB

> ls -e
Error: invalid option
> ls ./not_exist_dir
Error: No such file or directory
>
```

```
kw2020202092@ubuntu: ~/Assignment2_1_D_2020202092_KangHyunMin
./cli 127.0.0.1 4000
=====Client info=====
Client IP: 127.0.0.1
Client port: 42234
=====
NLST
NLST -a
NLST -l
NLST -al
NLST -al test_dir
NLST -e
NLST ./not_exist_dir
[]
```

그 후 not_exist_command 와 hello 를 입력한 결과 해당 커맨드가 서버에 출력되며, 정의되지 않은 명령이기 때문에 클라이언트로 Unknown command 에러 메시지가 출력되는 것을 확인할 수 있다.

```
kw2020202092@ubuntu: ~/Assignment2_1_D_2020202092_KangHyunMin
> ls
Makefile      cli      output/  srv
srv.c         test_dir/ testdir/
> ls -la
total 12
-rw-rw-r-- 1 kw2020202092 kw2020202092 55 Apr 27 06:40 Makefile
-rwxrwxr-x 1 kw2020202092 kw2020202092 17344 Apr 29 09:24 cli
-rwxrwxr-x 1 kw2020202092 kw2020202092 2508 Apr 29 09:17 cli.c
-rwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 28 05:32 output/
-rwxrwxr-x 1 kw2020202092 kw2020202092 22760 Apr 29 09:24 srv
-rwxrwxr-x 1 kw2020202092 kw2020202092 13976 Apr 29 09:24 srv.c
-rwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 29 09:25 test_dir/
-rwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 27 23:38 testdir/

> ls -al
total 12
-rw-rw-r-- 1 kw2020202092 kw2020202092 55 Apr 27 06:40 Makefile
-rwxrwxr-x 1 kw2020202092 kw2020202092 17344 Apr 29 09:24 cli
-rwxrwxr-x 1 kw2020202092 kw2020202092 2508 Apr 29 09:17 cli.c
-rwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 28 05:32 output/
-rwxrwxr-x 1 kw2020202092 kw2020202092 22760 Apr 29 09:24 srv
-rwxrwxr-x 1 kw2020202092 kw2020202092 13976 Apr 29 09:24 srv.c
-rwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 29 09:25 test_dir/
-rwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 27 23:38 testdir/

> ls -al test_dir
total 4
-rwxrwxr-x 2 kw2020202092 kw2020202092 4096 Apr 29 09:25 ./
-rwxrwxr-x 5 kw2020202092 kw2020202092 4096 Apr 29 09:25 ../
-rw-rw-r-- 1 kw2020202092 kw2020202092 0 Apr 29 09:25 fileA
-rw-rw-r-- 1 kw2020202092 kw2020202092 0 Apr 29 09:25 fileB

> ls -e
Error: invalid option
> ls ./not_exist_dir
Error: No such file or directory
> not_exist_command
Unknown command
> hello
Unknown command
> quit -e
Error: invalid option
> quit arg
Error: invalid option
> quit
Program quit!!
kw2020202092@ubuntu: ~/Assignment2_1_D_2020202092_KangHyunMin$

kw2020202092@ubuntu: ~/Assignment2_1_D_2020202092_KangHyunMin$ ./srv 4096
=====Client info=====
Client IP: 127.0.0.1

Client port: 42234
=====
NLST
NLST -a
NLST -l
NLST -al
NLST -al test_dir
NLST -e
NLST ./not_exist_dir
not_exist_command
hello
QUIT -e
QUIT arg
QUIT
kw2020202092@ubuntu: ~/Assignment2_1_D_2020202092_KangHyunMin$
```

그 후 quit 명령어를 입력하는데, 먼저 quit -e 와 quit arg 를 입력하여 quit 명령어에 대한 옵션과 인자가 입력되는 경우에 quit 의 FTP 명령어인 QUIT 으로 변환되어 서버에 출력되고, QUIT 명령어는 옵션과 인자를 사용하지 않기 때문에 클라이언트로 에러 메시지가 출력된다. 그리고 그 후 인자와 옵션 없이 quit 을 입력한 결과 서버에 QUIT 이 출력되고, 클라이언트에 Program quit!! 성공메세지가 출력되면서 클라이언트 프로그램이 종료되는 것을 확인할 수 있다.

고찰

이번 [FTP 2-1](#) 과제에서는 소켓 프로그래밍을 하였는데, 이 과제를 하면서 아쉬웠던 점은 먼저 서버에서 ls 명령어에 대한 옵션 파싱 과정이었다. getopt 함수를 사용하여 aflag 혹은 lflag 를 set 하여 옵션을 파싱하고, 이 변수를 통해 NLST 함수를 호출하여 동작을 구현하고자 하였으나 ls -al 명령어에 대한 플래그 set 이 의도대로 잘 동작하지 않았고, 결국 getopt 를 사용하지 않고 옵션을 파싱하는 방향으로 과제를 완성하게 되었다. 이 과제들 제출한 후 getopt 함수 사용 방법에 대해서 자세하게 학습해봐야 할 것 같다. 또한, 이 과제를 통해 깨닫게 된 점은 소켓 프로그래밍이 생각보다 복잡하다는 점이었다. 간단한 구조라도 클라이언트와 서버 사이에 socket, bind, connect, accept 등의 여러 함수가 사용되었고, 클라이언트에서 명령어를 서버로 전송하면 서버에서 명령어를 처리하고 해당 결과를 다시 클라이언트로 전송하여 출력하는 과정이 직접 코드를 작성하여 구현할 때 더욱 헷갈리게 느껴졌던 것 같다. 하지만 이번 과제를 직접 실습해 보면서 이론 시간에 학습한 클라이언트-서버 구조의 데이터 통신에서 더 나아가 이러한 데이터 통신의 구현 방법 및 자세한 동작에 대해 학습할 수 있는 계기가 되었다.

Reference