

시스템 프로그래밍 실습

[2-3: FTP server using socket]

Class : [금요일 1,2 교시]

Professor : [최상호 교수님]

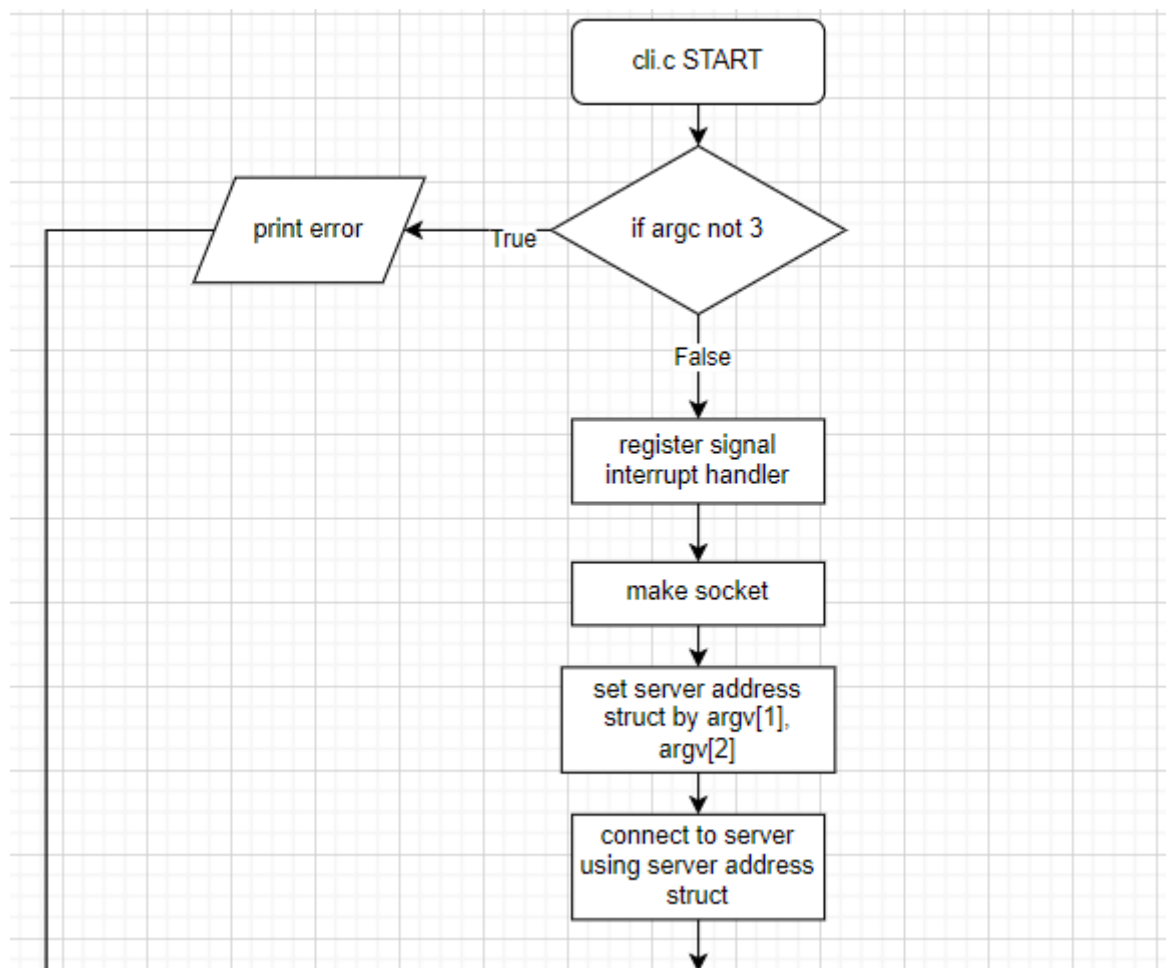
Student ID : [2020202034]

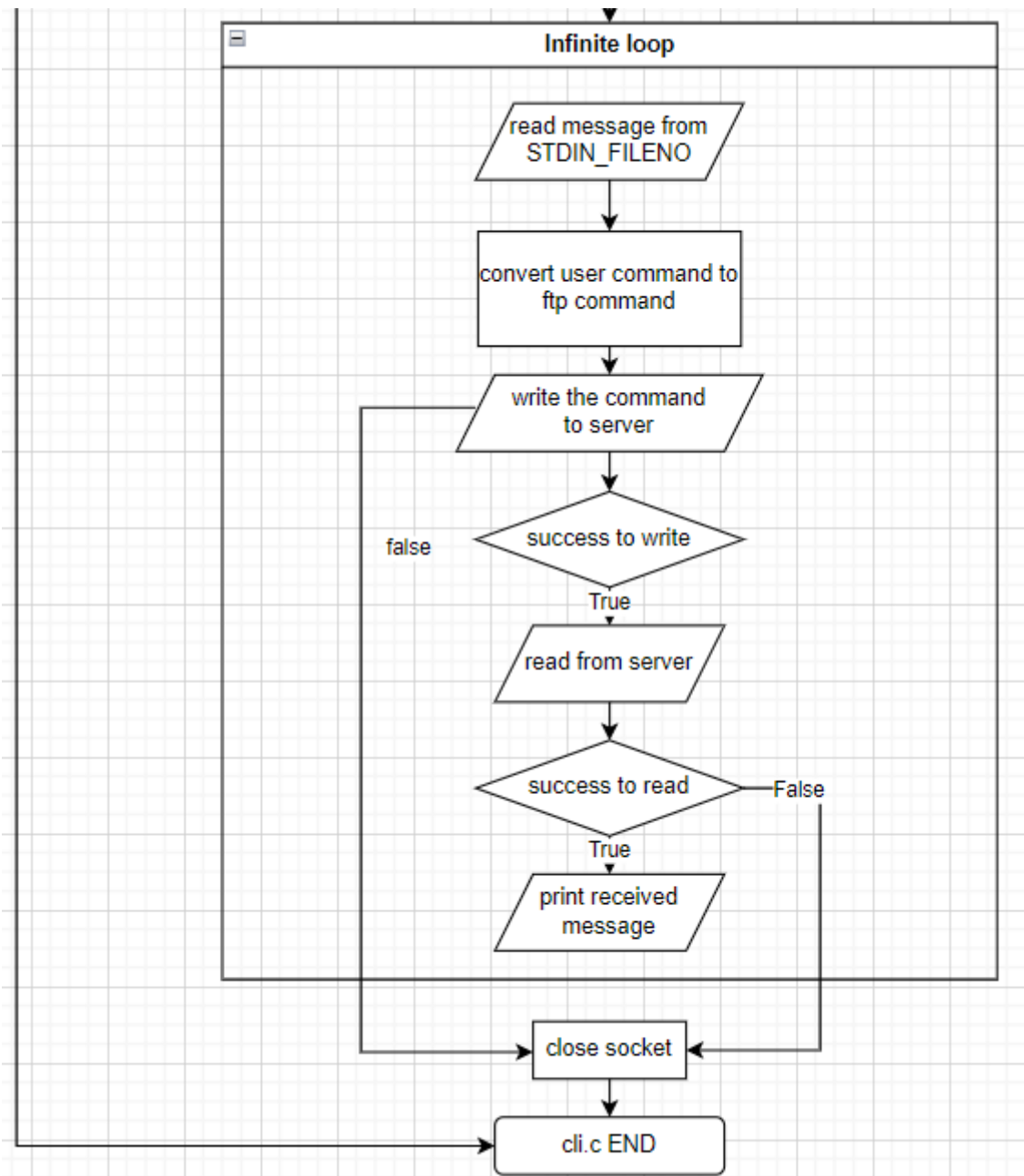
Name : [김태완]

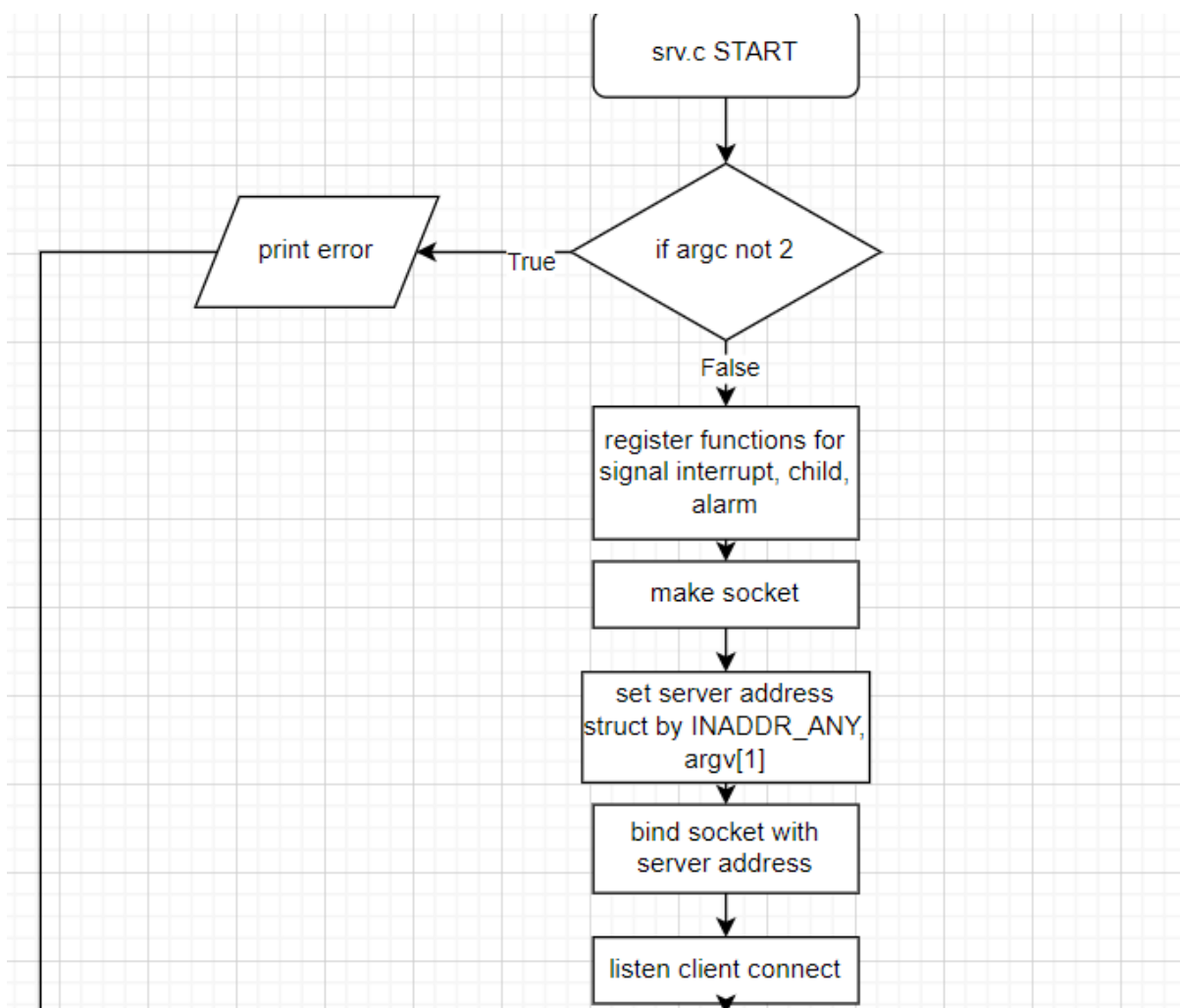
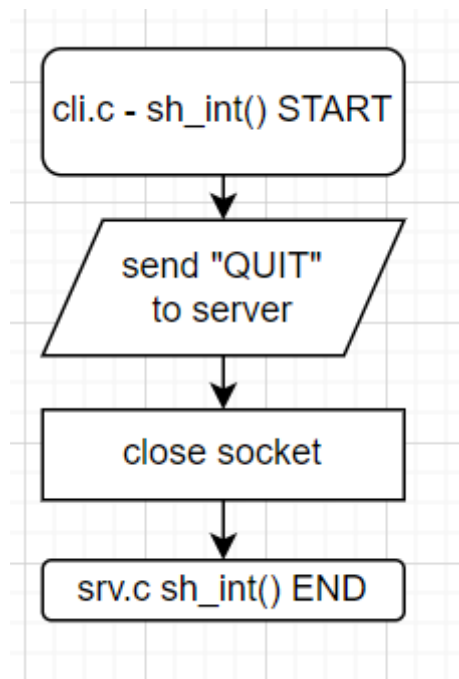
Introduction

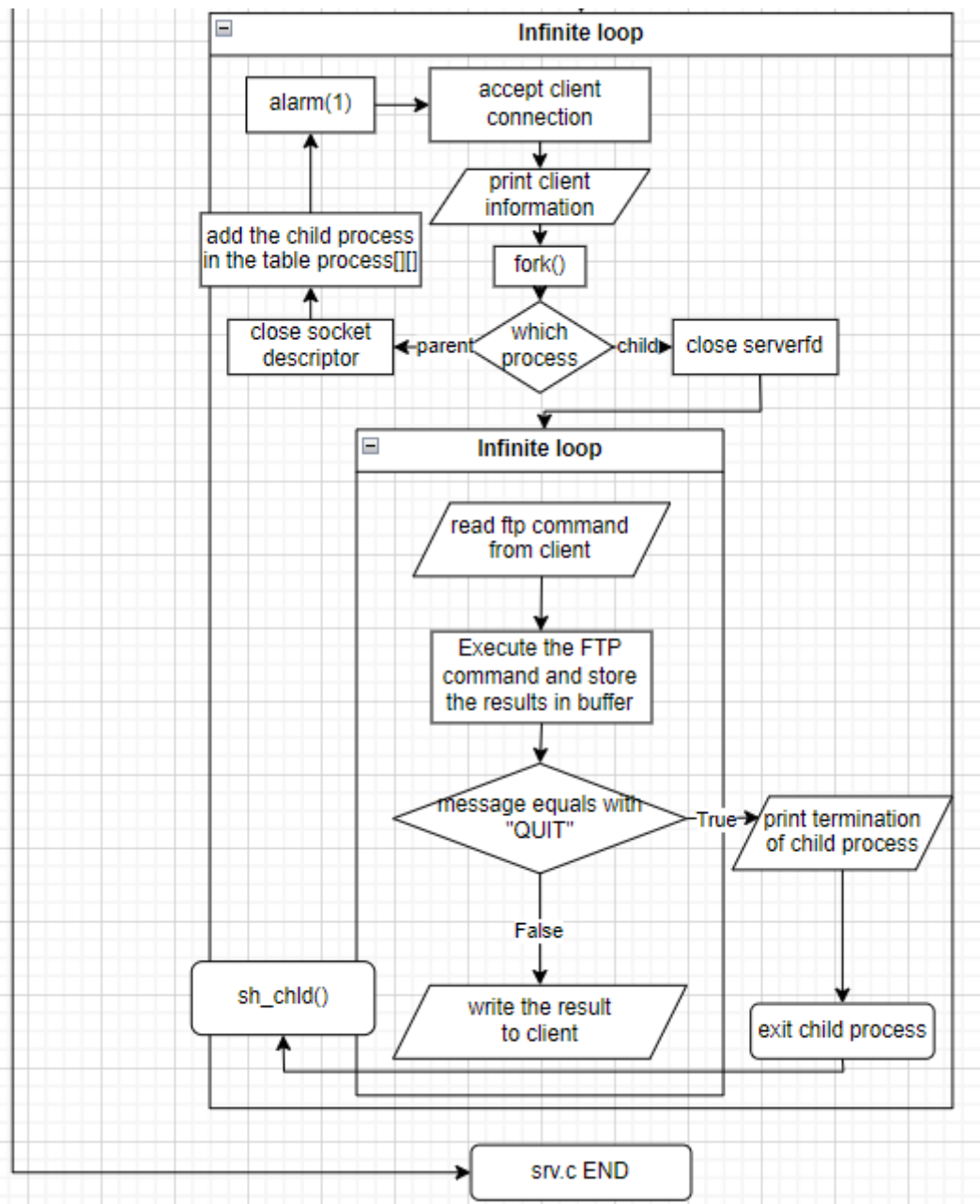
이번 과제는 1-3 에서 단순히 입력을 넣으면 출력(cli: user command -> ftp command, srv: ftp command -> print the results)을 해주는 식으로 구현한 FTP server & client를 socket을 이용한 multi-user server 로 재구현하는 과제입니다. 기존의 코드를 결합하는 것 외에 signal(SIGINT, SIGCHLD, SIGALRM)에 대한 추가적인 구현을 통해 강제 종료, 접속 중인 client 출력 등의 동작을 handling 할 수 있습니다.

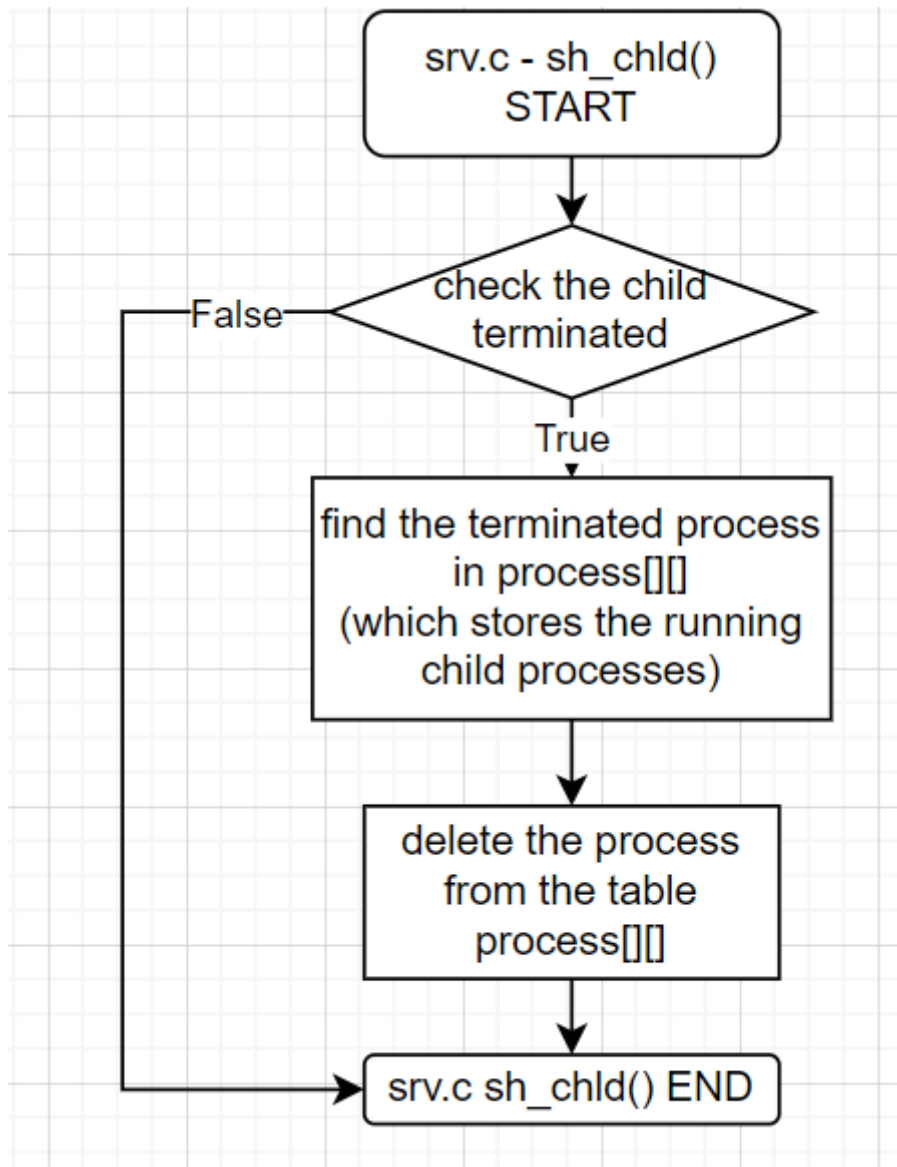
Flow chart

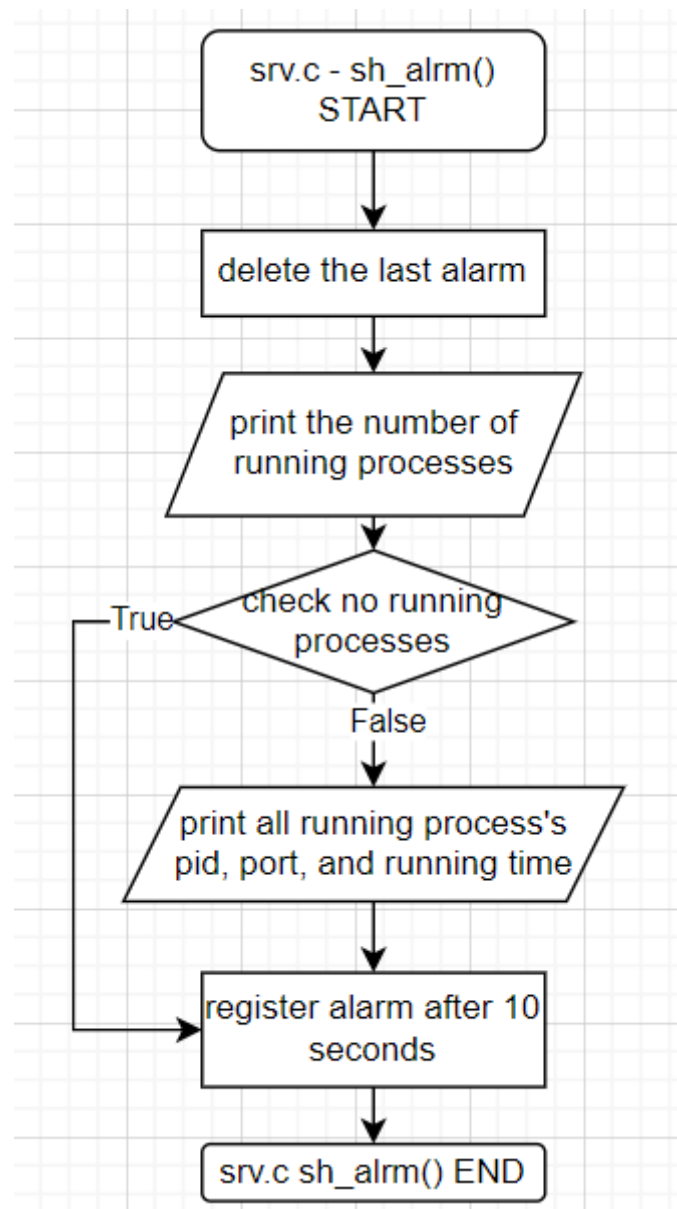


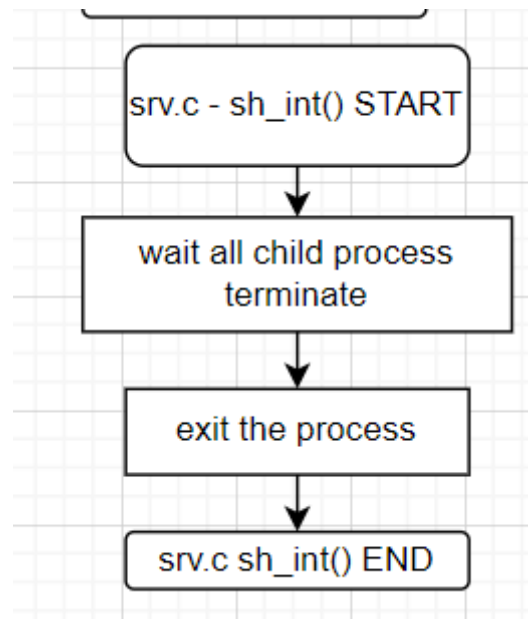


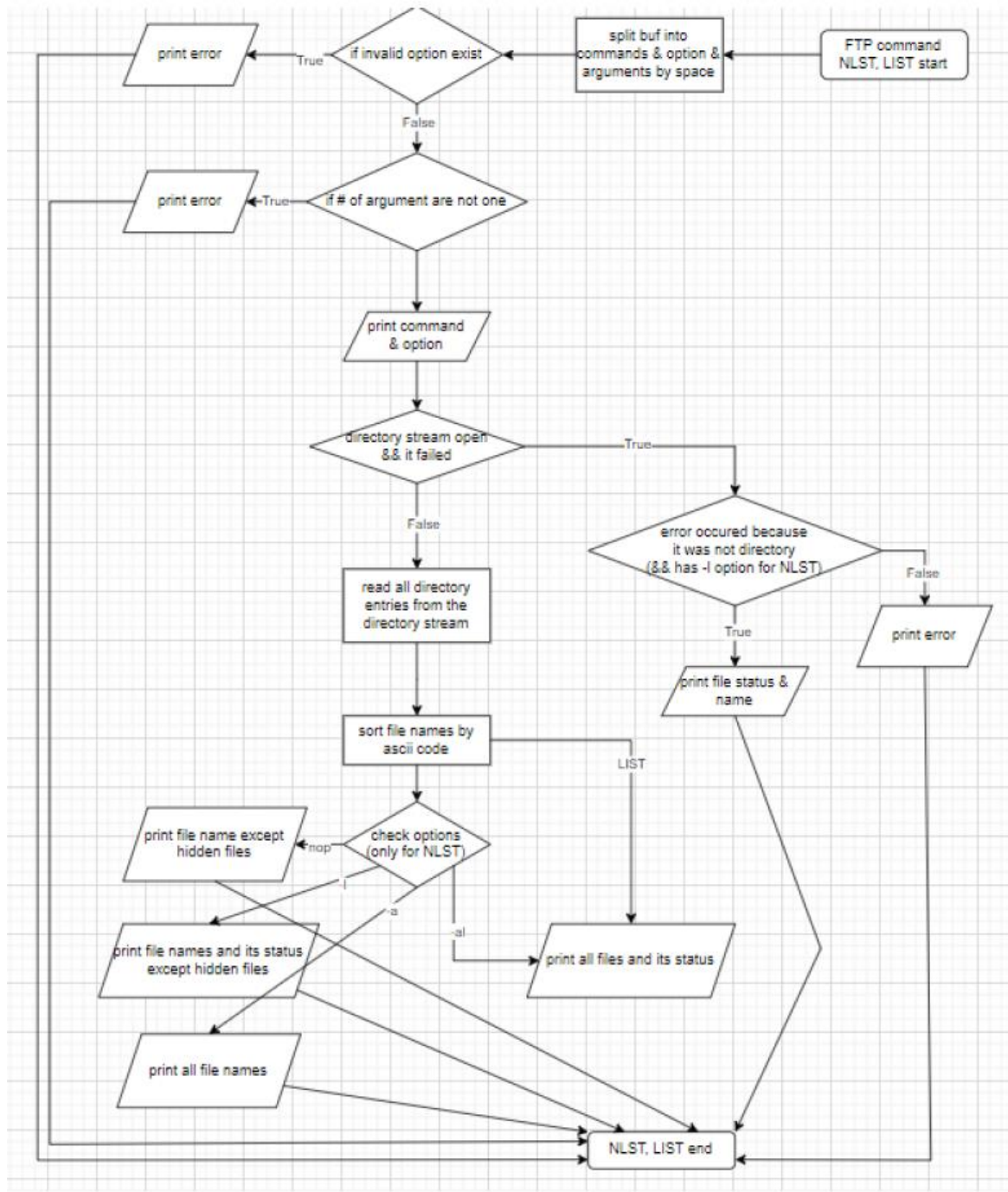


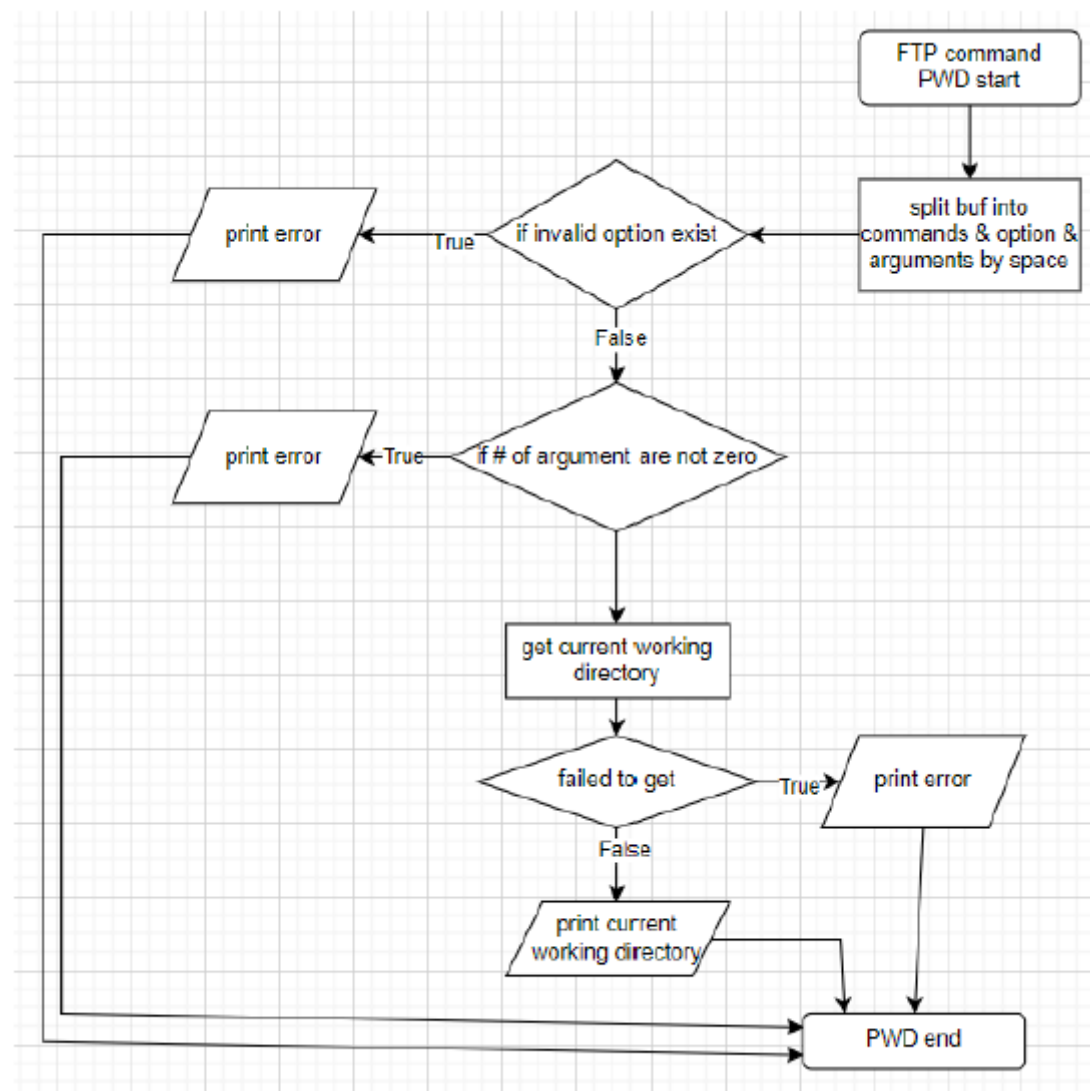


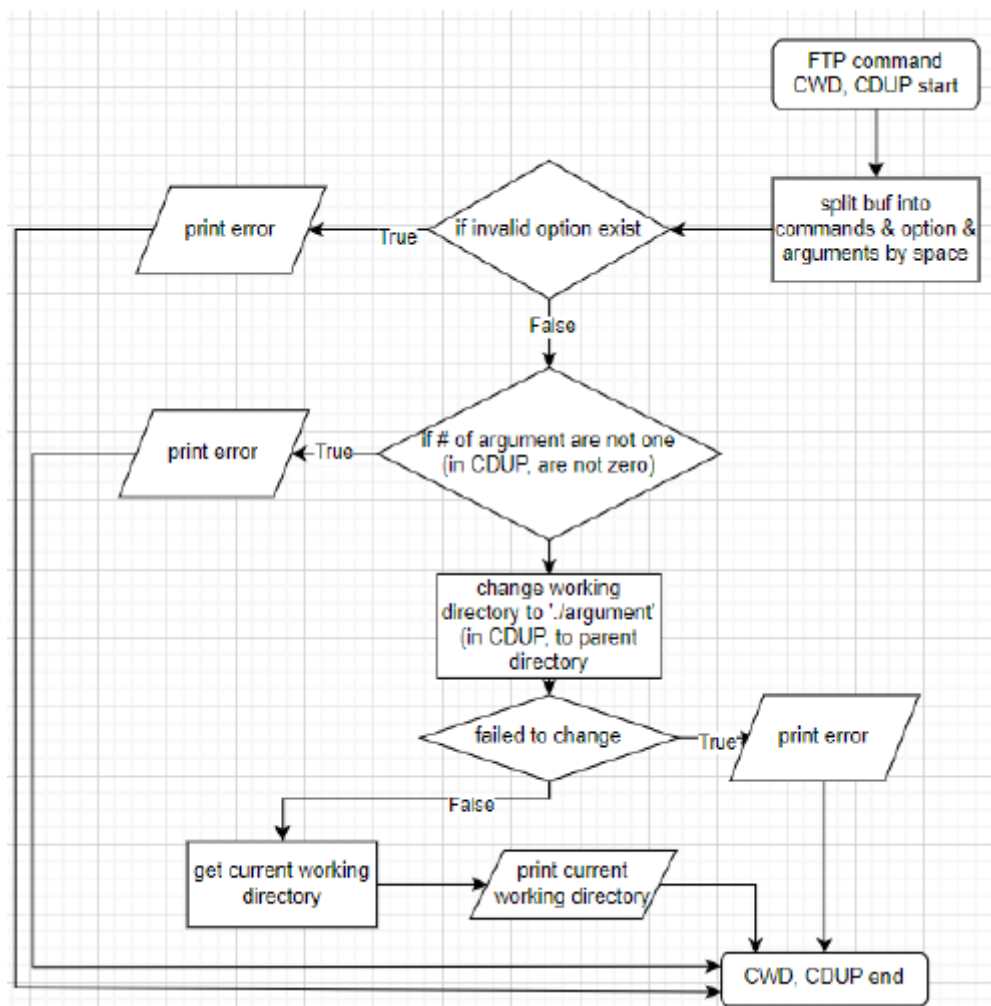


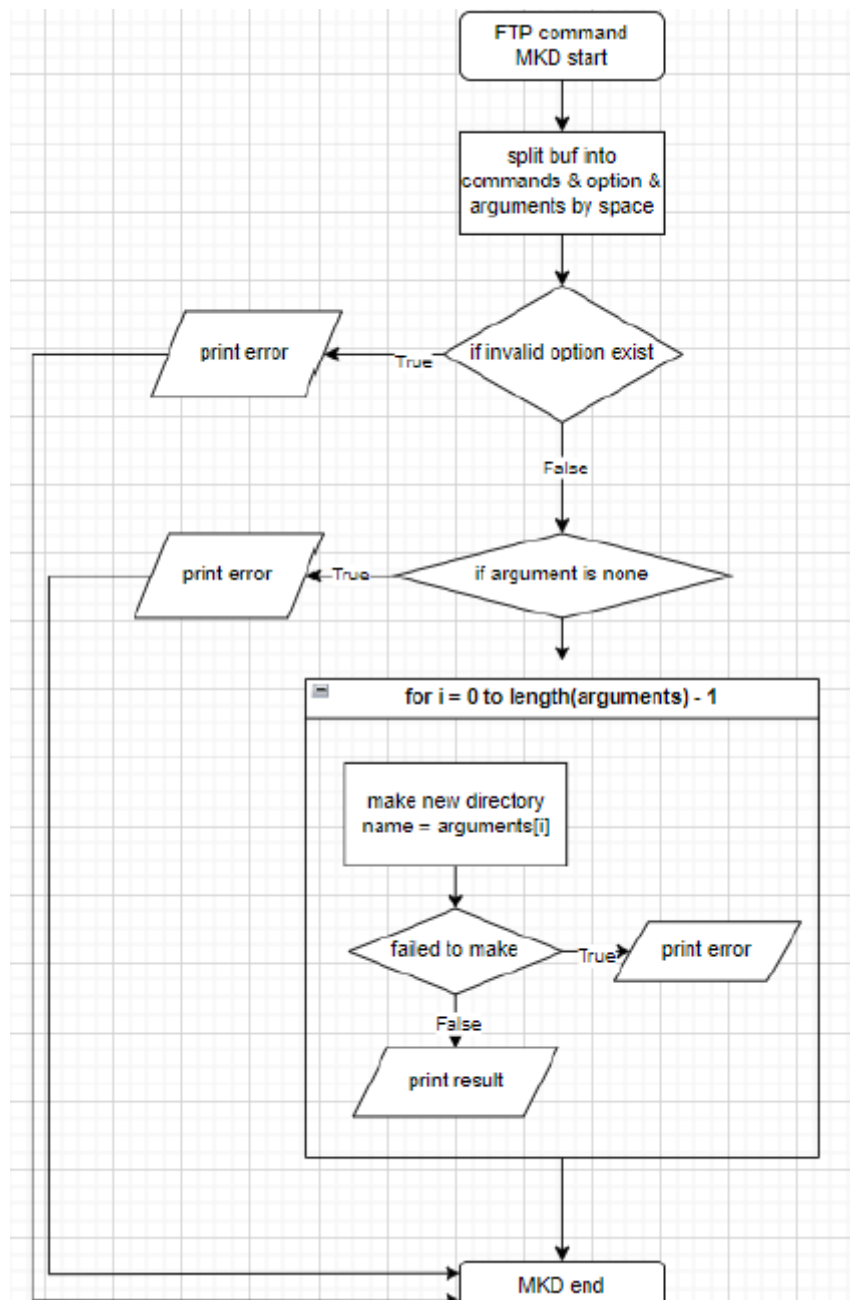


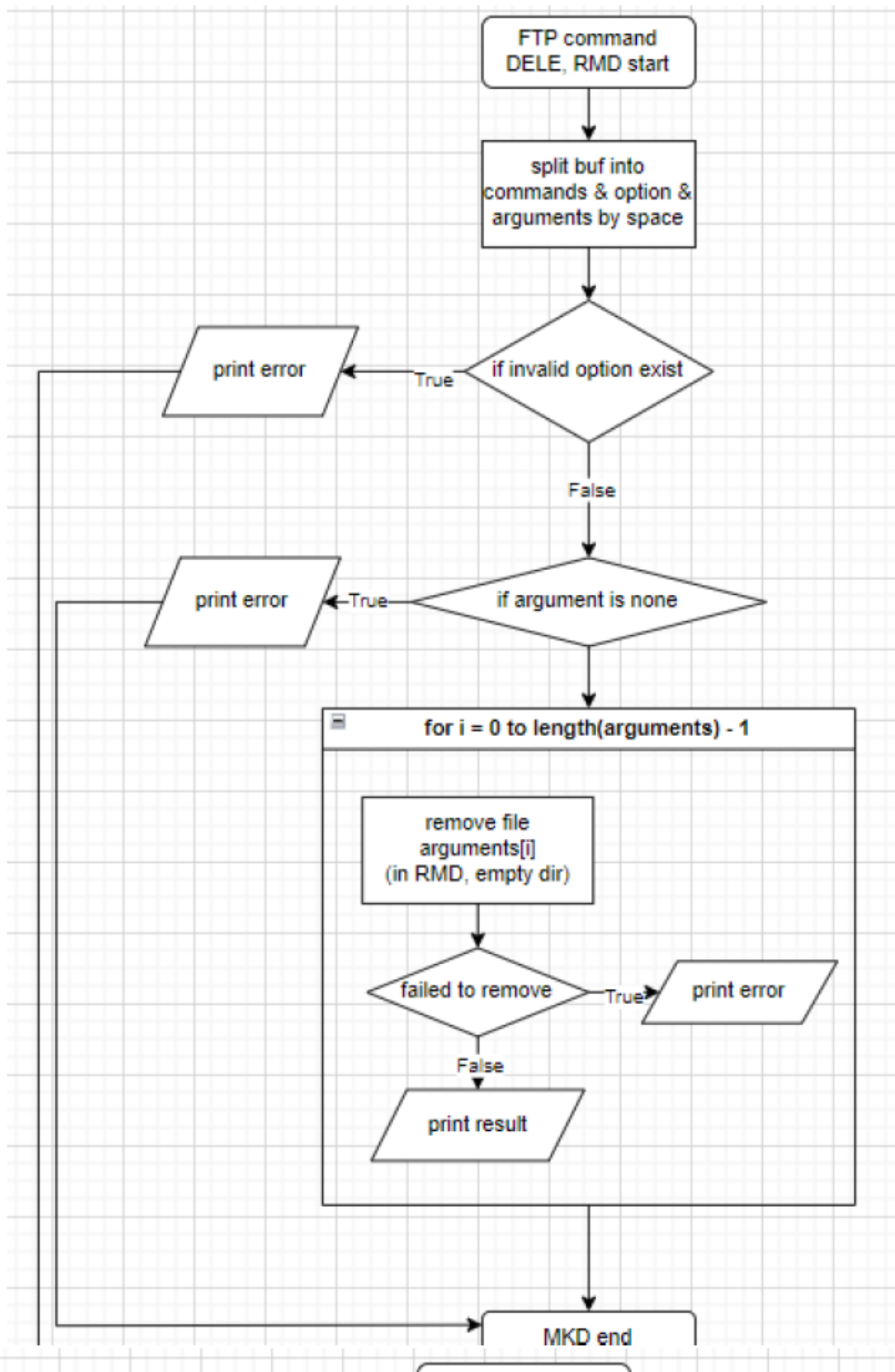


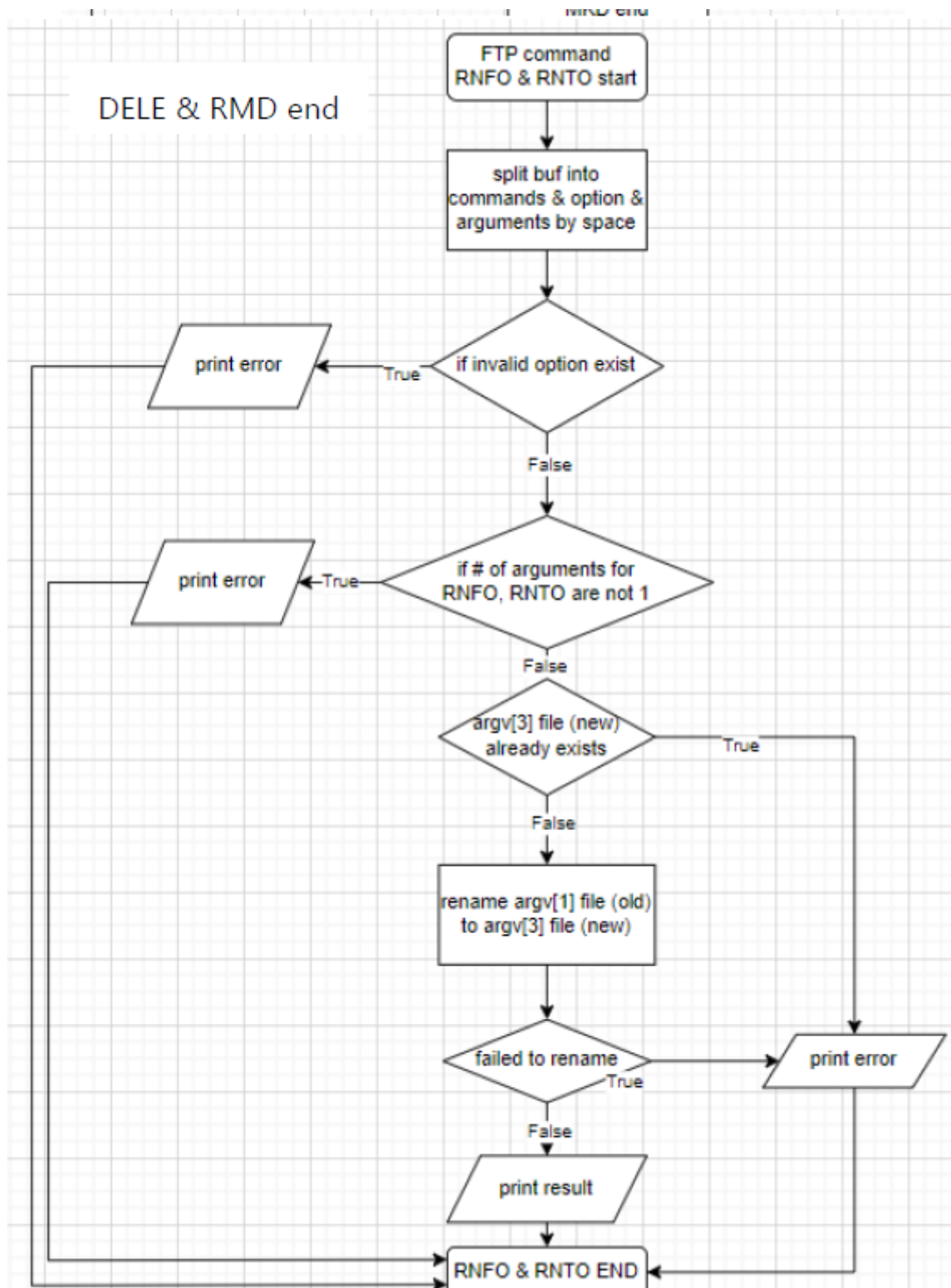


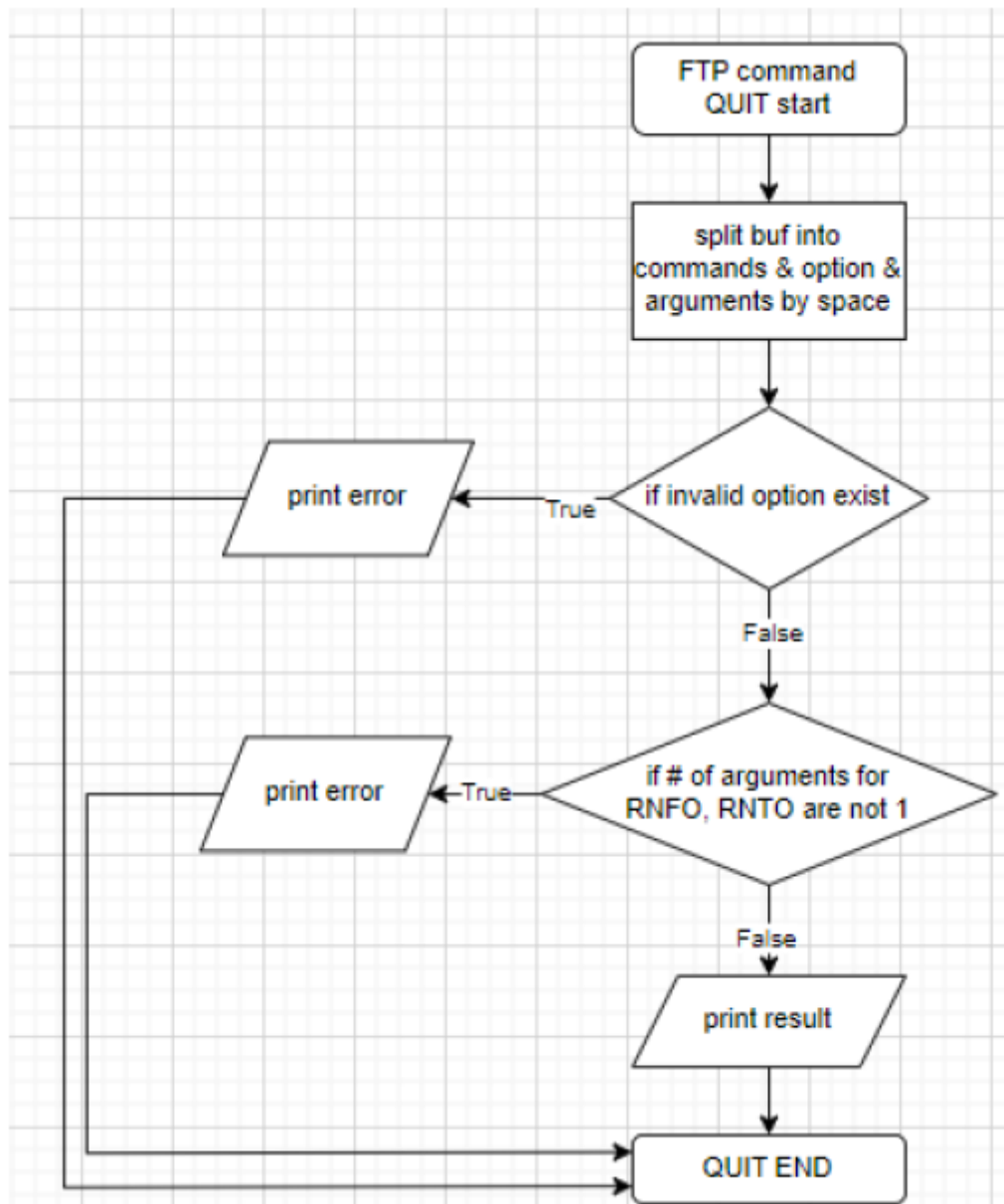












Pseudo code

cli.c

Main function:

Start

Create and initialize buffers

Create instruction conversion table

If the number of arguments is not correct

- Print error message

- Exit program

Register signal interrupt handler

Create socket

Set server address structure

Connect to server

If connection failed

- Print error

- Exit

Start loop

- Reset buffers

- Read user command input

- Split user command by whitespace

- Convert split command to FTP command using conversion table

- If rename command

 - Special handling for rename

- Else if unknown command

 - Set to "UNKNOWN"

Add options and arguments

Send command to server

Read server response

Print server response

If command is "QUIT"

Exit program

If connection failed

Exit loop

Close socket connection

End

Signal interrupt handler function:

Start

Send "QUIT" command to server

Close socket

Exit program

End

cli.c end

srv.c start

Main function:

Start

Declare buffers

Define instruction conversion table

If the number of arguments is not correct

Print error message

Exit program

Register signal interrupt handler

Create socket

Set server address structure

Bind server address to socket

Listen for client connections

Loop

Accept client connection

Get client address information

If failed to get client information

Print error message

Fork process

If fork failed

Print error message

Exit program

If child process

Close server socket

Loop

Reset buffers

Read command from client

If read successful

If command starts with "NLST"

Execute NLST function

Else if command starts with "LIST"

Execute LIST function

Else if command starts with "PWD"

Execute PWD function

Else if command starts with "CWD"

Execute CWD function

Else if command starts with "CDUP"

Execute CDUP function

Else if command starts with "MKD"

Execute MKD function

Else if command starts with "DELE"

Execute DELE function

Else if command starts with "RMD"

```
        Execute RMD function
    Else if command starts with "RNFR"
        Execute RN function
    Else if command starts with "QUIT"
        Execute QUIT function
    Else
        Set response to "unknown command"
    If response is "QUIT"
        Break loop
    Send response to client
    Close client socket
    Print termination message
    Exit child process
Else (parent process)
    Close client socket
    Increment process count
    Add process information to process table
    Set alarm
End loop
End
```

Signal interrupt handler function:

Start

Send "QUIT" command to server

Close socket

Exit program

End

Child signal handler function:

Start

If no child process terminated

Return

Decrement process count

Remove terminated process from process table

End

Alarm signal handler function:

Start

Cancel alarm

Print current number of client processes

If no client process

Set alarm for 10 seconds

Return

Print table header

Loop through process table

If process is not deleted

Print process information

Set alarm for 10 seconds

End

Client information function:

Start

Create client information string

Print client information string

If failed to print

Return -1

Return 0

End

Convert stat to string function:

Start

Get file type

Get user permissions

Get group permissions

Get other permissions

Format time string

If file is directory

Format file information string with directory indicator

Else

Format file information string without directory indicator

End

NLST function:

Start

Declare variables

Split command by space

Print command name and process ID

Parse options

If invalid option

Print error message

Return

If too many arguments

Print error message

Return

Set pathname

Open directory

If failed to open directory

Print error message

Return

Read directory entries and store in filenames array

Sort filenames by ASCII order

Find index of first non-hidden file

If -a option

Start printing from index 0

Else

Start printing from index of first non-hidden file

If -l option

Loop through filenames

Get file information

Convert file information to string

Concatenate string to print buffer

Else

Loop through filenames

Concatenate filename to print buffer

If directory, append '/' to filename

If reached 5 files per line, add newline

Close directory

End

LIST function:

Start

Declare variables

Split command by space

Print command name and process ID

Parse options

If option provided

 Print error message

 Return

If too many arguments

 Print error message

 Return

Set pathname

Open directory

If failed to open directory

 Print error message

 Return

Read directory entries and store in filenames array

Sort filenames by ASCII order

Loop through filenames

 Get file information

 Convert file information to string

 Concatenate string to print buffer

Close directory

End

PWD function:

Start

Declare variables

Split command by space

Print command name and process ID

Parse options

If option provided

Print error message

Return

If arguments provided

Print error message

Return

Get current working directory

Print current working directory

End

CWD function:

Start

Declare variables

Split command by space

Print command name and process ID

Parse options

If option provided

Print error message

Return

If no argument

Print error message

Return

If more than one argument

Print error message

Return

Expand '~' to home directory if present

Change working directory

If failed to change directory

Print error message

Return

Get current working directory

Print current working directory

End

CDUP function:

Start

Declare variables

Split command by space

Print command name and process ID

Parse options

If option provided

Print error message

Return

If arguments provided

Print error message

Return

Change directory to parent directory

If failed to change directory

Print error message

Return

Get current working directory

Print current working directory

End

MKD function:

Start

Declare variables

Split command by space

Print command name and process ID

Parse options

If option provided

Print error message

Return

If no argument

```
        Print error message
    Return
Loop through arguments
    Create directory
    If failed to create directory
        Print error message
    Else
        Print command and argument
End
```

DELE function:

```
Start
    Declare variables
    Split command by space
    Print command name and process ID
    Parse options
    If option provided
        Print error message
        Return
    If no argument
        Print error message
        Return
    Loop through arguments
```

Delete file

If failed to delete file

Print error message

Else

Print command and argument

End

RMD function:

Start

Declare variables

Split command by space

Print command name and process ID

Parse options

If option provided

Print error message

Return

If no argument

Print error message

Return

Loop through arguments

Remove directory

If failed to remove directory

Print error message

Else

Print command and argument

End

RN function:

Start

Declare variables

Split command by space

Print command name and process ID

Parse options

If option provided

Print error message

Return

If not two arguments (old and new)

Print error message

Return

If new filename already exists

Print error message

Return

Rename file

If failed to rename

Print error message

Else

```
        Print command and arguments
    End
```

QUIT function:

```
Start
    Declare variables
    Split command by space
    Parse options
    If option provided
        Print error message
        Return
    If arguments provided
        Print error message
        Return
    Set response to "QUIT"
End
```

```
srv.c end
```

결과화면


```
kw2020202034@ubuntu: ~/system_programming_1/2-3
kw2020202034@ubuntu:~/system_programming_1/2-3$ ./srv 3000
=====Client info=====
client IP: 127.0.0.1

client port: 37966
=====
> NLST [13700]
Current Number of Client : 1
  PID  PORT  TIME
13700 37966 1
=====Client info=====
client IP: 127.0.0.1

client port: 37982
=====
Current Number of Client : 2
  PID  PORT  TIME
13700 37966 6
13702 37982 1
> LIST [13702]
[ ]

kw2020202034@ubuntu:~/system_programming_1/2-3$ ./cli 127.0.0.1 3000
> ls
Makefile abc cli cli.c empty_file
srv srv.c
> [ ]

kw2020202034@ubuntu:~/system_programming_1/2-3$ ./cli 127.0.0.1 3000
> dir
drwxrwxr-x 2 kw2020202034 kw2020202034 4096 May 16 10:36 ./
drwxrwxr-x 10 kw2020202034 kw2020202034 4096 May 09 20:22 ../
-rw-rw-r-- 1 kw2020202034 kw2020202034 98 May 08 18:39 Makefile
-rw-rw-r-- 1 kw2020202034 kw2020202034 0 May 16 10:36 abc
-rwxrwxr-x 1 kw2020202034 kw2020202034 17656 May 16 09:51 cli
```

서버를 키고 클라이언트 1(우측 상단)을 켜니 서버에 접속 정보가 뜨는 것을 볼 수 있습니다. 그 후 ls와 ls -l 명령어를 실행하니 서버에는 명령어가, 클라이언트에는 결과가 뜨는 것을 볼 수 있습니다. 클라이언트 2(우측 하단)도 접속하니 접속 정보가 뜨고, 접속 중인 클라이언트가 2 명으로 나타남을 볼 수 있습니다. 또, dir(== ls -al) 명령어를 실행하니 결과가 클라이언트에 잘 나오고, 서버에 명령어가 나옴을 볼 수 있습니다.

그리고 10 초마다 접속중인 클라이언트의 정보를 출력하고 있음을 볼 수 있습니다.

```
kw2020202034@ubuntu:~/system_programming_1/2-3$ ./srv 3000
client port: 37982
=====
Current Number of Client : 2
  PID  PORT  TIME
13700 37966 6
13702 37982 1
> LIST [13702]
Current Number of Client : 2
  PID  PORT  TIME
13700 37966 16
13702 37982 11
Current Number of Client : 2
  PID  PORT  TIME
13700 37966 26
13702 37982 21
> NLST [13700]
Current Number of Client : 2
  PID  PORT  TIME
13700 37966 36
13702 37982 31
Current Number of Client : 2
  PID  PORT  TIME
13700 37966 46
13702 37982 41
> PWD [13702]
Current Number of Client : 2
  PID  PORT  TIME
13700 37966 56
13702 37982 51

kw2020202034@ubuntu:~/system_programming_1/2-3$ ./cli 127.0.0.1 3000
> ls
Makefile abc cli cli.c empty_file
srv srv.c
> ls -al ~/system_programming_1
drwxrwxr-x 10 kw2020202034 kw2020202034 4096 May 09 20:22 ./
drwxr-xr-x 28 kw2020202034 kw2020202034 4096 May 16 10:32 ../
drwxrwxr-x 8 kw2020202034 kw2020202034 4096 May 16 10:32 .git/
drwxrwxr-x 2 kw2020202034 kw2020202034 4096 Apr 01 17:40 .vscode/
drwxrwxr-x 2 kw2020202034 kw2020202034 4096 Apr 12 09:51 cli
1 cli
-rw-rw-r-- 1 kw2020202034 kw2020202034 4684 May 16 09:51 cli.c
-rw-rw-r-- 1 kw2020202034 kw2020202034 0 May 16 10:35 empty_file
-rwxrwxr-x 1 kw2020202034 kw2020202034 35832 May 16 10:32 srv
-rw-rw-r-- 1 kw2020202034 kw2020202034 39355 May 16 10:32 srv.c
> pwd
"/home/kw2020202034/system_programming_1/2-3" is current directory
>
```

client1 에는 ls 명령어에 -al 옵션을 주고, client2 는 pwd 명령어를 실행시키게 하였습니다. 결과가 잘 나옴을 확인할 수 있습니다.

```
kw2020202034@ubuntu: ~/system_programming_1/2-3
Current Number of Client : 2
PID PORT TIME
13700 37966 26
13702 37982 21
> NLST [13700]
Current Number of Client : 2
PID PORT TIME
13700 37966 36
13702 37982 31
Current Number of Client : 2
PID PORT TIME
13700 37966 46
13702 37982 41
> PWD [13702]
Current Number of Client : 2
PID PORT TIME
13700 37966 56
13702 37982 51
Current Number of Client : 2
PID PORT TIME
13700 37966 66
13702 37982 61
> CWD [13700]
Error: directory not found
Current Number of Client : 2
PID PORT TIME
13700 37966 76
13702 37982 71
> CDUP [13702]

kw2020202034@ubuntu: ~/system_programming_1/2-3
drwxrwxr-x 2 kw2020202034 kw2020202034 4096 Apr 09 11:
36 1-2/
drwxrwxr-x 3 kw2020202034 kw2020202034 4096 May 02 14:
42 1-3/
drwxrwxr-x 2 kw2020202034 kw2020202034 4096 May 02 14:
44 2-1/
drwxrwxr-x 2 kw2020202034 kw2020202034 4096 May 09 14:
54 2-2/
drwxrwxr-x 2 kw2020202034 kw2020202034 4096 May 16 10:
36 2-3/
-rw-rw-r-- 1 kw2020202034 kw2020202034 2222 Apr 08 20:
06 header.sh
> cd ~/not_exist_path
Error: directory not found
>

kw2020202034@ubuntu: ~/system_programming_1/2-3
-rw-rw-r-- 1 kw2020202034 kw2020202034 0 May 16 10:3
5 empty_file
-rwxrwxr-x 1 kw2020202034 kw2020202034 35832 May 16 10:3
2 srv
-rw-rw-r-- 1 kw2020202034 kw2020202034 39355 May 16 10:3
2 srv.c
> pwd
"/home/kw2020202034/system_programming_1/2-3" is current d
irectory
> cd ..
"/home/kw2020202034/system_programming_1" is current dire
tory
>
```

cli1, cli2 에 cd 명령어를 주었는데, cli1 은 존재하지 않는 경로를 인자로 받아 서버, 클라이언트에 모두 에러를 출력하였고, cli2 는 옳은 결과가 잘 출력되었음을 확인할 수 있습니다.

```
kw2020202034@ubuntu: ~/system_programming_1/2-3
Current Number of Client : 2
PID PORT TIME
13700 37966 86
13702 37982 81
> MKD [13700]
Current Number of Client : 2
PID PORT TIME
13700 37966 96
13702 37982 91
> DELE [13702]
Error: failed to delete 'empty_file'
Current Number of Client : 2
PID PORT TIME
13700 37966 106
13702 37982 101
Current Number of Client : 2
PID PORT TIME
13700 37966 116
13702 37982 111
Current Number of Client : 2
PID PORT TIME
13700 37966 126
13702 37982 121
> CWD [13702]
Current Number of Client : 2
PID PORT TIME
13700 37966 136
13702 37982 131
> DELE [13702]

kw2020202034@ubuntu: ~/system_programming_1/2-3
42 1-3/
drwxrwxr-x 2 kw2020202034 kw2020202034 4096 May 02 14:
44 2-1/
drwxrwxr-x 2 kw2020202034 kw2020202034 4096 May 09 14:
54 2-2/
drwxrwxr-x 2 kw2020202034 kw2020202034 4096 May 16 10:
36 2-3/
-rw-rw-r-- 1 kw2020202034 kw2020202034 2222 Apr 08 20:
06 header.sh
> cd ~/not_exist_path
Error: directory not found
> mkdir d1 d2
MKD d1
MKD d2
>

kw2020202034@ubuntu: ~/system_programming_1/2-3
"/home/kw2020202034/system_programming_1/2-3" is current d
irectory
> cd ..
"/home/kw2020202034/system_programming_1" is current dire
tory
> delete empty_file
Error: failed to delete 'empty_file'
> cd 2-3
"/home/kw2020202034/system_programming_1/2-3" is current d
irectory
> delete empty_file
DELE empty_file
>
```

cli1 에는 mkdir 로 디렉토리를 생성하게 하고, cli2 는 delete 로 파일을 지우게 하였습니다. mkdir 은 성공적으로 되었지만, cli2 에서는 현재 경로가 상위 경로로 바뀌어있어 에러가 났으므로 다시 현재 경로로 이동 후 delete 를 실행하니 오류 없이 잘 실행됨을 볼 수 있습니다.

```
kw2020202034@ubuntu: ~/system_programming_1/2-3
13700 37966 116
13702 37982 111
Current Number of Client : 2
PID PORT TIME
13700 37966 126
13702 37982 121
> CWD [13702]
Current Number of Client : 2
PID PORT TIME
13700 37966 136
13702 37982 131
> DELE [13702]
Current Number of Client : 2
PID PORT TIME
13700 37966 146
13702 37982 141
> RMD [13700]
RMD d1
RMD d2
Error: failed to remove 'd3'
Current Number of Client : 2
PID PORT TIME
13700 37966 156
13702 37982 151
> (null) [13702]
Current Number of Client : 2
PID PORT TIME
13700 37966 166
13702 37982 161

kw2020202034@ubuntu: ~/system_programming_1/2-3
54 2-2/
drwxrwxr-x 2 kw2020202034 kw2020202034 4096 May 16 10:
36 2-3/
-rw-rw-r-- 1 kw2020202034 kw2020202034 2222 Apr 08 20:
06 header.sh
> cd ~/not_exist_path
Error: directory not found
> mkdir d1 d2
MKD d1
MKD d2
> rmdir d1 d2 d3
RMD d1
RMD d2
Error: failed to remove 'd3'
>

kw2020202034@ubuntu: ~/system_programming_1/2-3
"/home/kw2020202034/system_programming_1" is current dire
tory
> delete empty_file
Error: failed to delete 'empty_file'
> cd 2-3
"/home/kw2020202034/system_programming_1/2-3" is current d
irectory
> delete empty_file
DELE empty_file
> rename abc def
RNFR abc
RNTD def
>
```

cli1 에서는 아까 mkdir 로 만든 d1, d2 를 지우는데, 존재하지 않는 d3 도 지우게 해 오류 출력을 검증하였습니다. cli2 에서는 존재하는 abc 파일을 def 로 바꾸도록 rename 을 사용하였고, 잘 되었음을 확인할 수 있습니다.

```
kw2020202034@ubuntu: ~/system_programming_1/2-3$ ./srv 3000
=====Client info=====
client IP: 127.0.0.1

client port: 57878
=====
Current Number of Client : 1
PID PORT TIME
13979 57878 1
> RNFR & RNTD [13979]
Error: name to change already exists
Current Number of Client : 1
PID PORT TIME
13979 57878 11
> RNFR & RNTD [13979]

kw2020202034@ubuntu: ~/system_programming_1/2-3$ ./cli 127.0.0.1 3000
> rename def abc
Error: name to change already exists
> rename abc def
RNFR abc
RNTD def
>
```

이전 사진에서 rename 을 입력 시 서버에 RNFR & RNTD 대신 (null)이 나오는 것을 테스트 중에 확인해 이를 수정하고 검증한 것입니다. 명령어 변환 및 결과가 잘 수행되었음을 알 수 있습니다.

```
kw2020202034@ubuntu: ~/system_programming_1/2-3
Current Number of Client : 2
  PID  PORT  TIME
13700 37966 146
13702 37982 141
> RMD
RMD d1
RMD d2
Error: failed to remove 'd3'
Current Number of Client : 2
  PID  PORT  TIME
13700 37966 156
13702 37982 151
> (null) [13702]
Current Number of Client : 2
  PID  PORT  TIME
13700 37966 166
13702 37982 161
Current Number of Client : 2
  PID  PORT  TIME
13700 37966 176
13702 37982 171
Client (13700)'s Release

Current Number of Client : 1
  PID  PORT  TIME
13702 37982 181
Client (13702)'s Release

Current Number of Client : 0

kw2020202034@ubuntu: ~/system_programming_1/2-3
drwxrwxr-x 2 kw2020202034 kw2020202034 4096 May 16 10:
36 2-3/
-rw-rw-r-- 1 kw2020202034 kw2020202034 2222 Apr 08 20:
06 header.sh
> cd ~/not_exist_path
Error: directory not found
> mkdir d1 d2
MKD d1
MKD d2
> rmdir d1 d2 d3
RMD d1
RMD d2
Error: failed to remove 'd3'
> quit
kw2020202034@ubuntu: ~/system_programming_1/2-3$

kw2020202034@ubuntu: ~/system_programming_1/2-3
"/home/kw2020202034/system_programming_1" is current dire
tory
> delete empty_file
Error: failed to delete 'empty_file'
> cd 2-3
"/home/kw2020202034/system_programming_1/2-3" is current d
irectory
> delete empty_file
DELE empty_file
> rename abc def
RNFR abc
RNT0 def
^Ckw2020202034@ubuntu: ~/system_programming_1/2-3$
```

cli1 은 quit 으로, cli2 는 ctrl+c(sigint)로 종료하였는데, 이것이 잘 수행되어 각자 종료될 때 메시지가 바로 나오고, 추가적인 명령어 입력이 없으면 10 초마다 울리는 알람에서 client 와의 연결이 끊겼음을 잘 보이는 것을 볼 수 있습니다.

```
kw2020202034@ubuntu: ~/system_programming_1/2-3
  PID  PORT  TIME
13700 37966 156
13702 37982 151
> (null) [13702]
Current Number of Client : 2
  PID  PORT  TIME
13700 37966 166
13702 37982 161
Current Number of Client : 2
  PID  PORT  TIME
13700 37966 176
13702 37982 171
Client (13700)'s Release

Current Number of Client : 1
  PID  PORT  TIME
13702 37982 181
Client (13702)'s Release

Current Number of Client : 0
Current Number of Client : 0
=====Client info=====
client IP: 127.0.0.1

client port: 42548
=====
Current Number of Client : 1
  PID  PORT  TIME
13722 42548 1
^Ckw2020202034@ubuntu: ~/system_programming_1/2-3$

kw2020202034@ubuntu: ~/system_programming_1/2-3
06 header.sh
> cd ~/not_exist_path
Error: directory not found
> mkdir d1 d2
MKD d1
MKD d2
> rmdir d1 d2 d3
RMD d1
RMD d2
Error: failed to remove 'd3'
> quit
kw2020202034@ubuntu: ~/system_programming_1/2-3$ ./cli 127
.0.0.1 3000
>
kw2020202034@ubuntu: ~/system_programming_1/2-3$

kw2020202034@ubuntu: ~/system_programming_1/2-3
"/home/kw2020202034/system_programming_1" is current dire
tory
> delete empty_file
Error: failed to delete 'empty_file'
> cd 2-3
"/home/kw2020202034/system_programming_1/2-3" is current d
irectory
> delete empty_file
DELE empty_file
> rename abc def
RNFR abc
RNT0 def
^Ckw2020202034@ubuntu: ~/system_programming_1/2-3$
```

모든 client 종료 후 다시 client 하나를 연결 후, 이번엔 서버에서 ctrl+c(sigint)를 눌러 강제 종료시켰습니다. 그러니 client 도 종료되었음을 알 수 있습니다.

고찰

1-3에서는 client와 server의 데이터 전송이 표준 입력, 표준 출력을 통해 이루어졌었는데, 이번 과제에서는 소켓 통신을 통해 이루어져야 하므로 이를 2-2에서 만들었던 코드를 1-3에 추가해 구현하는데 사소한 오류들이 있어 이를 고치는데 시간을 좀 썼습니다. 또, 기존에는 server side에서 수행한 명령어만 출력하였는데, 에러가 났을 때 이를 출력하는 추가 수정사항도 적용시켰습니다.