

시스템 프로그래밍 실습

[Assignment #1-3 FTP simulation(cli, srv)]

Class : [금요일 1, 2 교시]

Professor : [최상호 교수님]

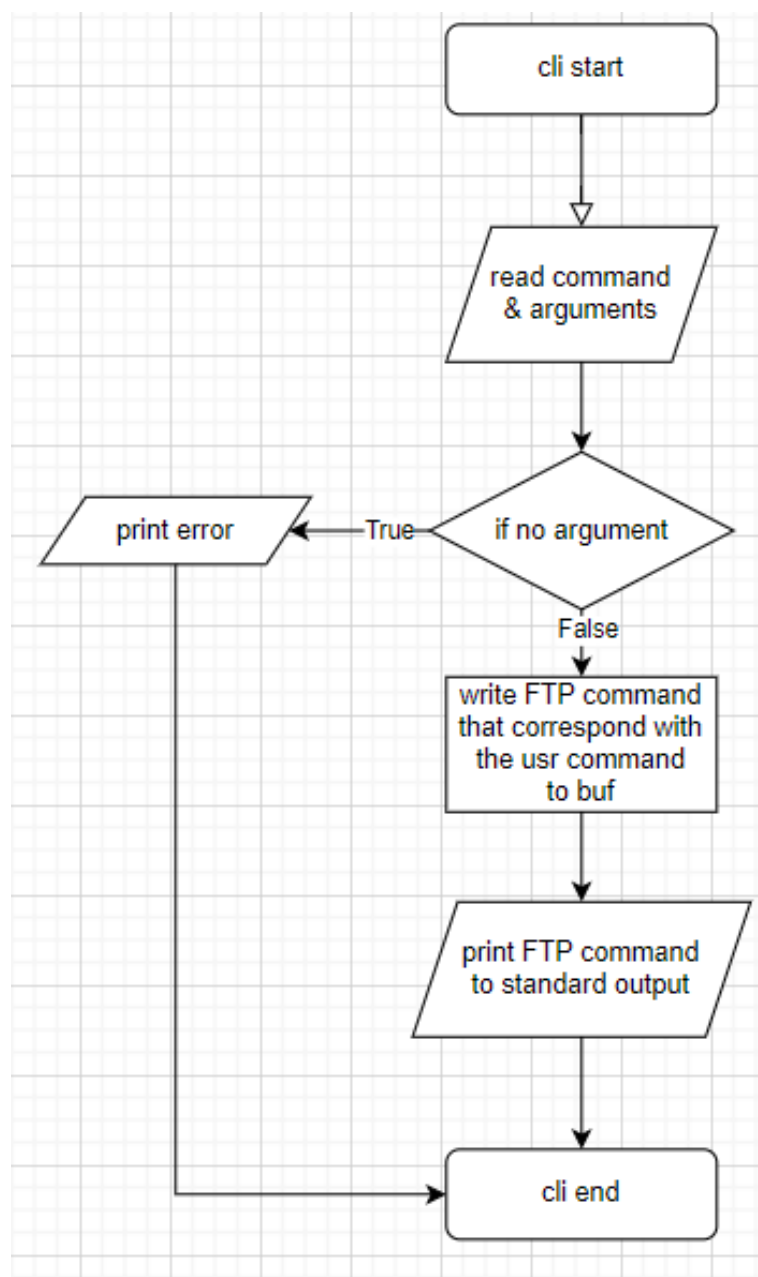
Student ID : [2020202034]

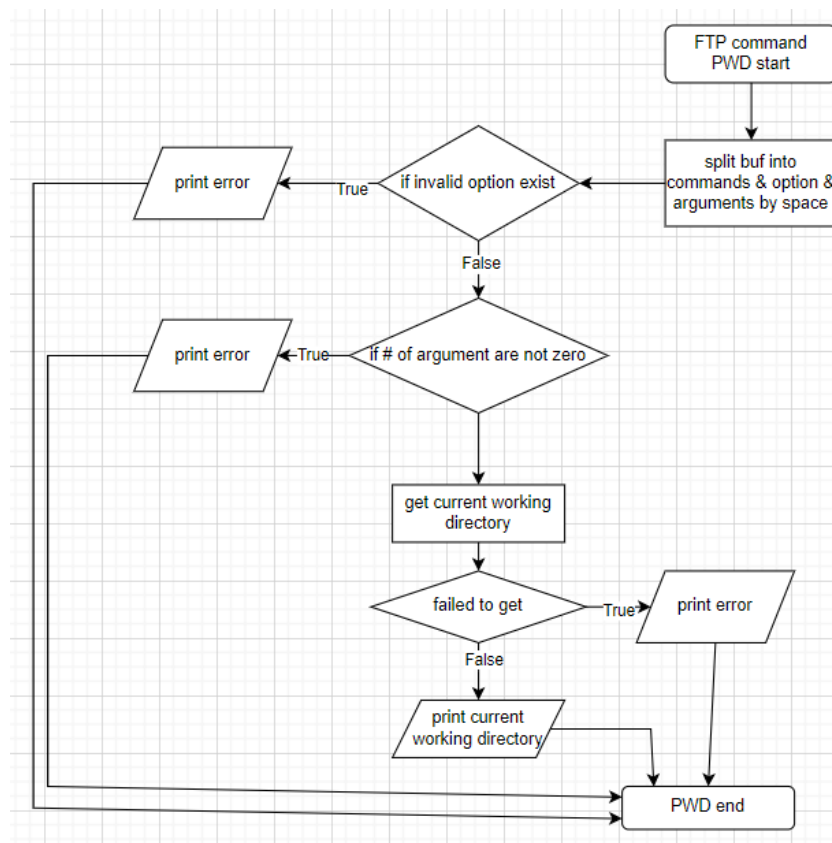
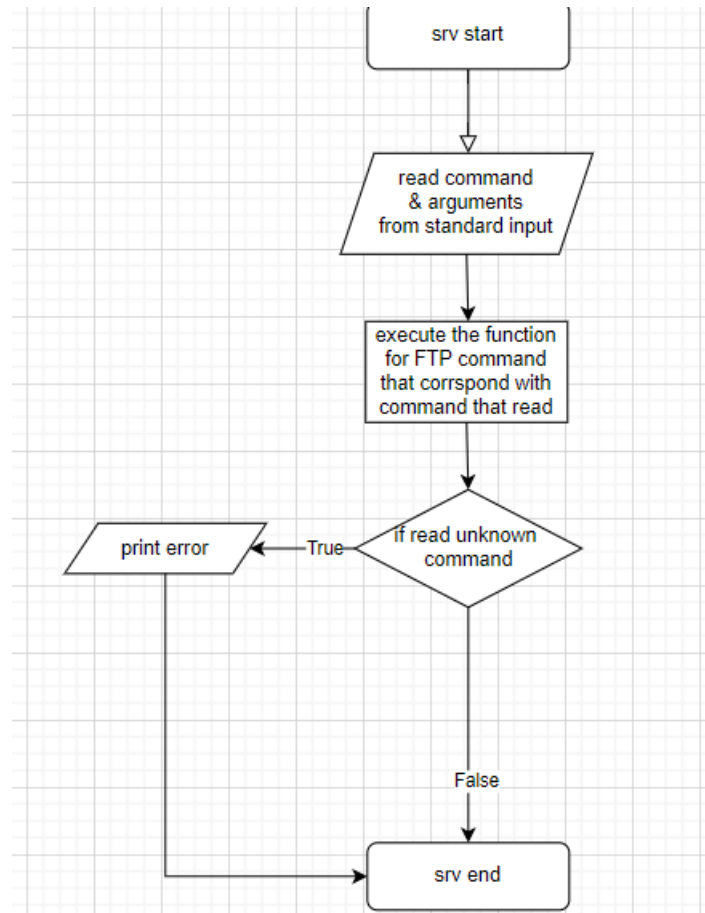
Name : [김태완]

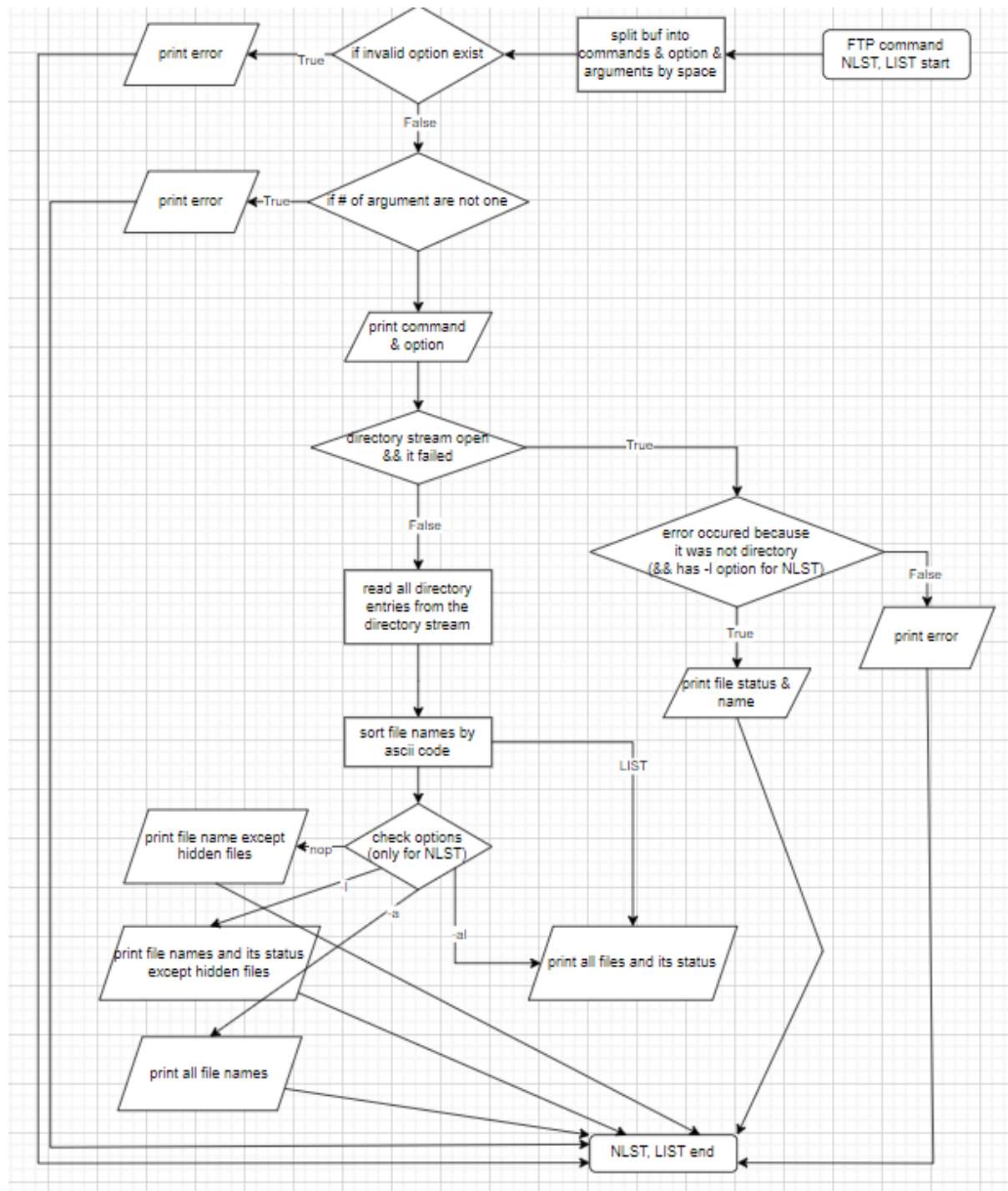
Introduction

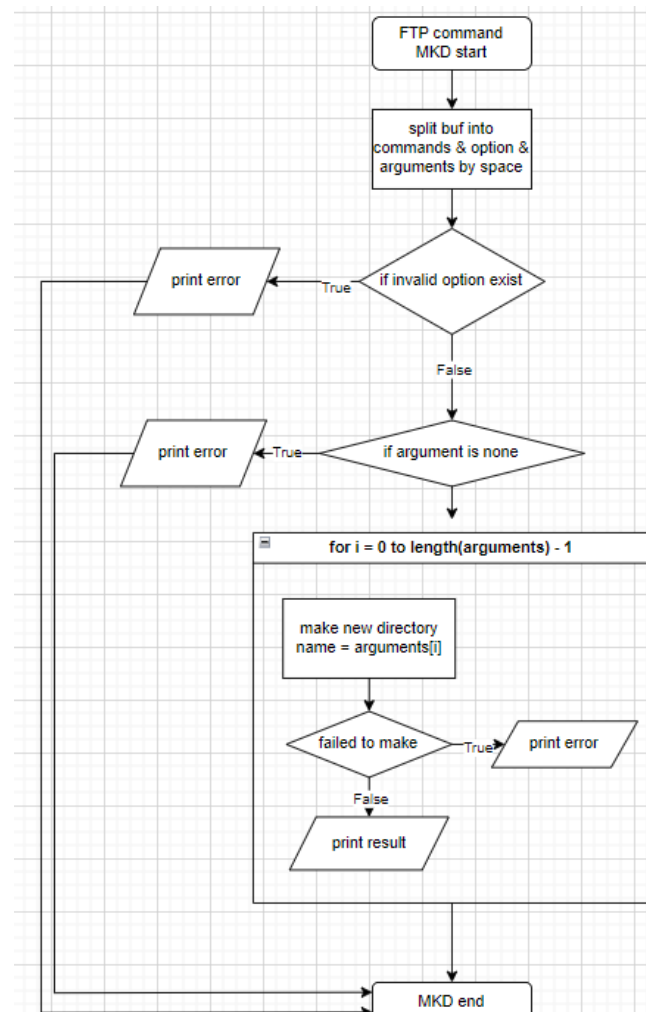
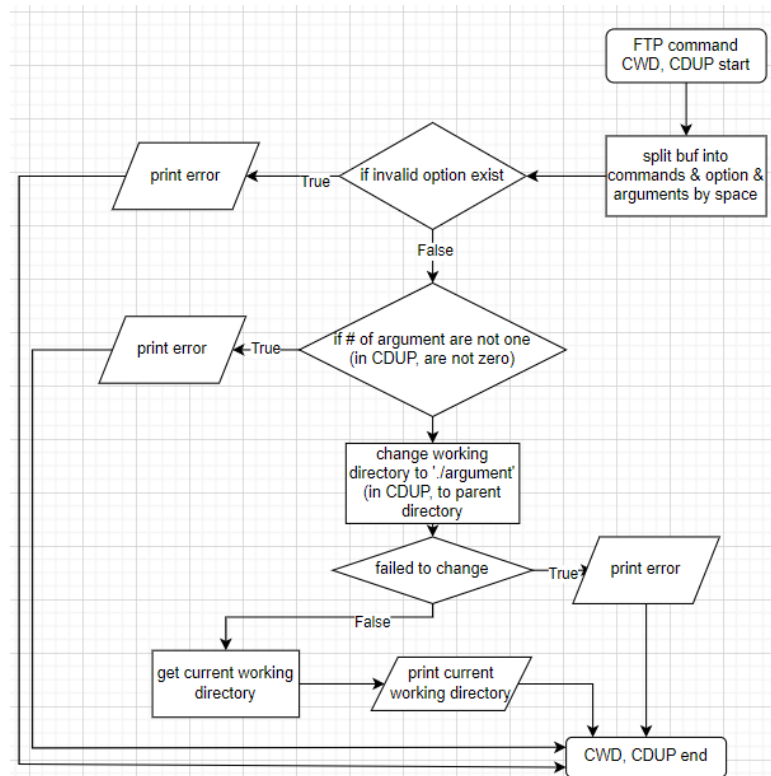
이번 과제는 FTP server(srv)와 client(cli)의 간소화된 버전을 만드는 내용입니다. cli 프로그램은 유저 명령어를 인자로 받아 이를 FTP 명령어로 반환해 출력하여 주고, srv 프로그램은 FTP server 로써 FTP 명령어를 받아 이를 실행시키는 기능을 수행합니다. srv 프로그램에서 구현해야 하는 FTP 명령어는 NLST(ls), LIST(dir), PWD(pwd), CWD(cd), CDUP(cd ..), MKD(mkdir), RMD(rmdir), DELE(delete), RNFR & RNTD(rename), EXIT(quit) 총 11 개 입니다.

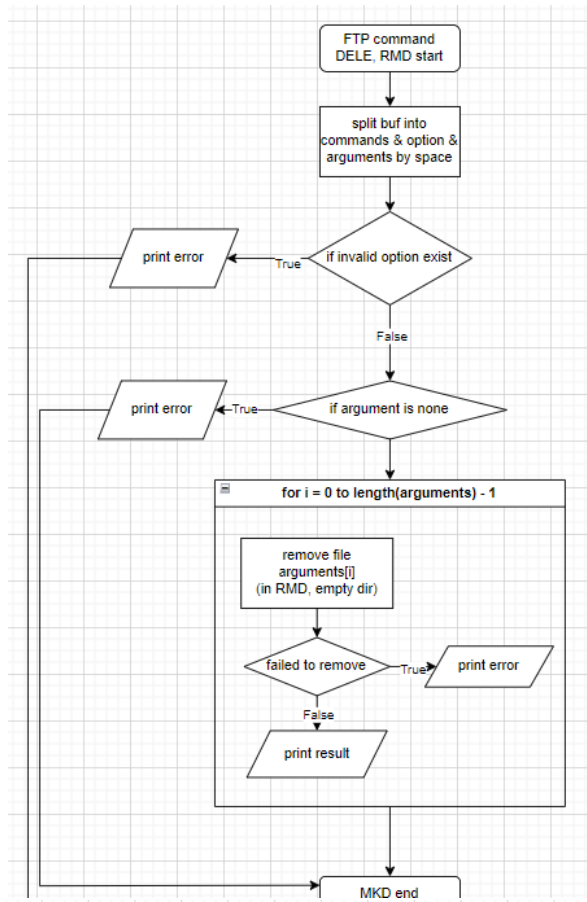
Flow chart



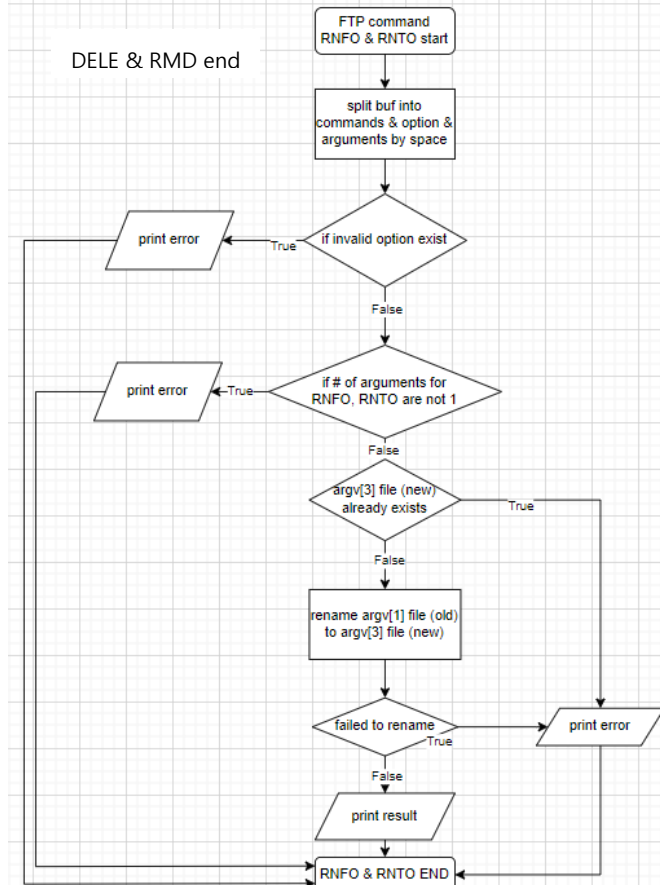


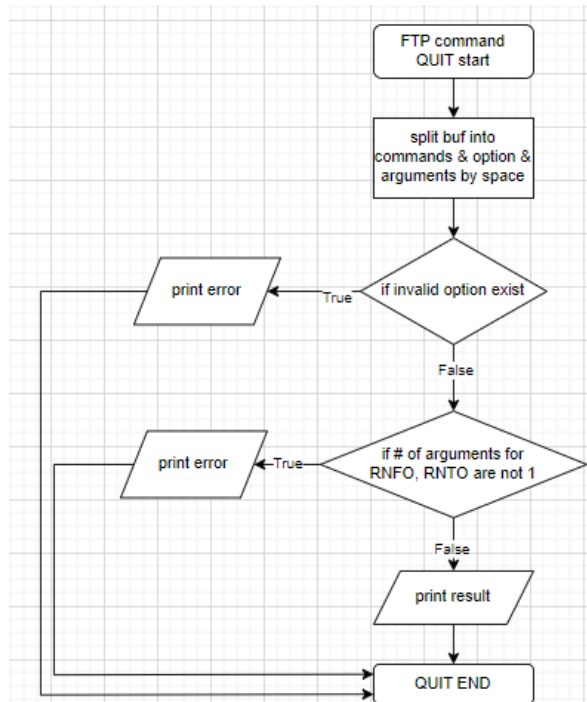






DELE & RMD end





Pseudo code

cli.c

```

Main(argc, argv) {
    char *buf[MAX_BUF]
    initialize char *table[][2] with "usr command: NLST command"
    If (argc == 1) {
        Write error message to stderr
        stop
    }
    For i = 0 to 7
        If (argv[1] equals table[i][0]) Then
            Concatenate table[i][1] to buf
            Break from loop
        }
    }
    If (argv[1] equals "rename") {
        Concatenate "RNFR" to buf
        For i = 2 to argc - 2
            Concatenate " " and argv[i] to buf
        Concatenate " RNTD " and argv[argc - 1] to buf
    }
}
  
```

```

        Write buf to stdout
        stop
    }
    If (buf[0] is empty)
        Concatenate "UNKNOWN" to buf
    For i = 2 to argc - 1
        If (argv[1] equals "cd" and argv[i] equals "..") {
            Move buf one position to the right
            Copy "CDUP" to the beginning of buf
            Continue to the next iteration
        }
        Concatenate " " and argv[i] to buf
    }
    Write buf to stdout
    stop
}

```

srv.c

```

main() {

    char buf[];

    set table[10] for {FTP command, cmd's Function}

    read standard input and stores it in buf;

    for i 0 to 10 {

        if (buf is about the FTP command in the table)

            execute the function in the table[]

        print error message : unknown command

        stop

    }
}

```



```

NLST(char * buf) {
    split buf by space and store in split
    detect options in split[1:len] // split[0] is "NLST"
    if invalid options are in split[]:
        print error message and stop
    if # of non-option arguments are not zero or 1:
        print error message and stop
    print command and options
    if the command has no arguments:
        pathname = "." // current directory
    else
        pathname = argument
    open directory stream by pathname
    if failed to open && the command has only -l option:
        get file status and translate it to string
        write the file information string
        stop
    else if failed to open and not -l option
        print error message and stop
    read directory entries by the directory stream
    sort the directory entries by ascii code
    if -a option exist:
        print_start_index is 0
    else:

```

print_start_index is after the last hidden file(i.e. .gitignore, ...)

if -l option exist:

for l = print_start_index to len:

write split[i]'s status information in one line

else:

for l = print_start_index to len:

write split[i]'s name, 5 names in one line

close directory stream

stop

LIST(char *buf) {

Split the input command string by space

store each part in an array called split

if there are any invalid options passed with the command:

print an error message and exit

if there are neither zero nor one arguments after options:

print an error message and exit.

Print the name of the command

if command's argument none:

pathname = "."

else

pathname = argument

Open the directory specified by the pathname

If unsuccessful, handle errors.

Read all directory entries

store their names in an array called filename[]

Sort filenames by ASCII code.

Print file stat information for each file.

Close the directory stream and exit the program.

}

PWD (char *buf) {

Split the input command string by space

store each part in an array called split

if there are any invalid options passed with the command:

print an error message and exit

if there are more than one arguments after options:

print an error message and exit

get current working directory

print the current working directory

stop

}

CWD (char *buf) {

Split the input command string by space

store each part in an array called split

if there are any invalid options passed with the command:

print an error message and exit

if the command has not one argument(none or two or more):

print an error message and exit

change current working directory by the argument

if failed:

print an error message and exit

print the current working directory

stop

}

CDUP (char *buf) {

Split the input command string by space

store each part in an array called split

if there are any invalid options passed with the command:

print an error message and exit

if command's arguments exist:

print an error message and exit.

change working directory to parent directory

if failed:

print an error message

print the current working directory

exit

}

MKD (char *buf) {

Split the input command string by space

store each part in an array called split

if there are any invalid options passed with the command:

print an error message and exit

if there are no arguments:

print an error message and exit.

for i = optind to len:

mkdir directory(its name is argument[i])

if failed:

print error message

else:

print command and directory name

exit

}

DELE (char *buf) {

Split the input command string by space

store each part in an array called split

if there are any invalid options passed with the command:

print an error message and exit

if there are no arguments:

print an error message and exit.

for i = optind to len:

remove files that correspond with argument[i]

if failed:

print error message

else:

print command and file name

exit

}

RMD (char *buf) {

Split the input command string by space

store each part in an array called split

if there are any invalid options passed with the command:

print an error message and exit

if there are no arguments after options:

print an error message and exit.

for i = optind to len:

remove empty directory(its name is argument[i])

if failed:

print error message

else:

print command and directory name

exit

}

RN (char *buf) {

Split the input command string by space

store each part in an array called split

if there are any invalid options passed with the command:

print an error message and exit

if there are not one argument after both RNFR and RNTD:

print an error message and exit.

if the file that have same name with RNTD's argument(new name):

```

        print an error message and exit

rename old name file to new name

if failed:

    print an error message and exit

    print command and file name

    exit
}

QUIT(char *buf) {

    Split the input command string by space

    store each part in an array called split

    if there are any invalid options passed with the command:

        print an error message and exit

    if there are more than one arguments after options:

        print an error message and exit.

    print success message

    stop
}

```

결과화면

```

kw2020202034@ubuntu:~/system_programming_1/1-3$ make
gcc    cli.c    -o cli
gcc    srv.c    -o srv
kw2020202034@ubuntu:~/system_programming_1/1-3$ touch test_file1 test_file2
kw2020202034@ubuntu:~/system_programming_1/1-3$ mkdir test_dir_1 test_dir_2
kw2020202034@ubuntu:~/system_programming_1/1-3$ chmod 000 test_dir_2

```

소스 코드를 make 를 이용해 컴파일하였고, 출력 예시 화면과 같은 결과를 보이기 위해 test file, test directory 를 만들었습니다. 그 중 test_dir_2 는 권한을 모두 없애 ls(read)가 불가능하도록 만들었습니다.

```

kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli ls -e | tee cli.out | ./srv
Error: invalid option
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli ls | tee cli.out | ./srv
NLST
Makefile cli cli.c cli.out srv
srv.c test_dir_1/ test_dir_2/ test_file1 test_file2

kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli ls -a | tee cli.out | ./srv
NLST -a
./ ../ Makefile cli cli.c
cli.out srv srv.c test_dir_1/ test_dir_2/
test_file1 test_file2
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli ls -l | tee cli.out | ./srv
NLST -l
-rw-rw-r-- 1 kw2020202034 kw2020202034 90 Apr 16 09:24 Makefile
-rwxrwxr-x 1 kw2020202034 kw2020202034 16968 Apr 17 15:11 cli
-rw-rw-r-- 1 kw2020202034 kw2020202034 2444 Apr 16 19:09 cli.c
-rw-rw-r-- 1 kw2020202034 kw2020202034 7 Apr 17 15:14 cli.out
-rwxrwxr-x 1 kw2020202034 kw2020202034 26560 Apr 17 15:11 srv
-rw-rw-r-- 1 kw2020202034 kw2020202034 29412 Apr 16 19:34 srv.c
drwxrwxr-x 2 kw2020202034 kw2020202034 4096 Apr 17 15:11 test_dir_1/
d----- 2 kw2020202034 kw2020202034 4096 Apr 17 15:11 test_dir_2/
-rw-rw-r-- 1 kw2020202034 kw2020202034 0 Apr 17 15:11 test_file1
-rw-rw-r-- 1 kw2020202034 kw2020202034 0 Apr 17 15:11 test_file2

```

NLST(cli: ls) 명령어의 구현 결과입니다. -a, -l 옵션을 받으며, 인자는 받지 않거나 하나만 받습니다.

우선, -a, -l 옵션 외의 옵션을 주면 예외처리 되도록 하였습니다.

옵션, 인자 모두 없으며 현재 디렉토리 내 파일 이름들을 출력합니다. 저는 예시처럼 한 줄에 최대 5 개의 파일 이름이 나오도록 하였고, ascii code 순으로 정렬 및 디렉토리 뒤에는 '/'를 붙여 파일 타입이 구분됨을 볼 수 있습니다.

-a 옵션만 추가하게 되면 .(current directory), ..(parent directory) 같은 .으로 시작하는 파일도 나타남을 볼 수 있습니다.

-l 옵션만 추가하게 되면 파일의 세부 정보(권한, 링크 수, UID, GID name, 용량, timestamp, 파일 이름)이 한 줄에 한 파일씩 출력되는 것을 볼 수 있습니다.

```

kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli ls -al | tee cli.out | ./srv
NLST -al
drwxrwxr-x 4 kw2020202034 kw2020202034 4096 Apr 17 15:11 ./
drwxrwxr-x 7 kw2020202034 kw2020202034 4096 Apr 17 15:08 ../
-rw-rw-r-- 1 kw2020202034 kw2020202034 90 Apr 16 09:24 Makefile
-rwxrwxr-x 1 kw2020202034 kw2020202034 16968 Apr 17 15:11 cli
-rw-rw-r-- 1 kw2020202034 kw2020202034 2444 Apr 16 19:09 cli.c
-rw-rw-r-- 1 kw2020202034 kw2020202034 8 Apr 17 15:15 cli.out
-rwxrwxr-x 1 kw2020202034 kw2020202034 26560 Apr 17 15:11 srv
-rw-rw-r-- 1 kw2020202034 kw2020202034 29412 Apr 16 19:34 srv.c
drwxrwxr-x 2 kw2020202034 kw2020202034 4096 Apr 17 15:11 test_dir_1/
d----- 2 kw2020202034 kw2020202034 4096 Apr 17 15:11 test_dir_2/
-rw-rw-r-- 1 kw2020202034 kw2020202034 0 Apr 17 15:11 test_file1
-rw-rw-r-- 1 kw2020202034 kw2020202034 0 Apr 17 15:11 test_file2
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli ls -al ./not_exist_path | tee cli.out | ./srv
NLST -al
Error : No such file or directory
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli ls -al ./test_dir_2 | tee cli.out | ./srv
NLST -al
Error : cannot access

```


-al 옵션, 즉 -a 와 -l 옵션을 동시에 주면 파일의 세부정보가 출력되면서 .나 ..같은 .으로 시작하는 파일의 정보도 출력되게 됩니다.

옵션은 -al 을 주고, 인자로 존재하지 않는 경로를 주면 이에 따른 에러 메시지를 출력합니다.

옵션은 -al을 주고, read 권한이 없는 test_dir_2를 인자로 주면, 접근 권한이 없다는 에러 메시지를 출력합니다.

```
kw202020234@ubuntu:~/system_programming_1/1-3$ ./cli ls -a Makefile | tee cli.out | ./srv
NLST -a
Error : Not a directory
kw202020234@ubuntu:~/system_programming_1/1-3$ ./cli ls -l Makefile | tee cli.out | ./srv
NLST -l
-rw-rw-r-- 1 kw202020234 kw202020234 90 Apr 16 09:24 Makefile
```

-l 옵션을 주지 않으면 ls 의 인자로 파일 명을 주었을 때 에러를 출력합니다.

-l 옵션이 주어지면 인자로 파일명을 받았을 때 해당 파일의 세부 정보를 출력합니다.

```
kw202020234@ubuntu:~/system_programming_1/1-3$ ./cli dir | tee cli.out | ./srv
LIST
drwxrwxr-x 4 kw202020234 kw202020234 4096 Apr 17 15:11 ./
drwxrwxr-x 7 kw202020234 kw202020234 4096 Apr 17 15:08 ../
-rw-rw-r-- 1 kw202020234 kw202020234 90 Apr 16 09:24 Makefile
-rwxrwxr-x 1 kw202020234 kw202020234 16968 Apr 17 15:11 cli
-rw-rw-r-- 1 kw202020234 kw202020234 2444 Apr 16 19:09 cli.c
-rw-rw-r-- 1 kw202020234 kw202020234 4 Apr 17 15:17 cli.out
-rwxrwxr-x 1 kw202020234 kw202020234 26560 Apr 17 15:11 srv
-rw-rw-r-- 1 kw202020234 kw202020234 29412 Apr 16 19:34 srv.c
drwxrwxr-x 2 kw202020234 kw202020234 4096 Apr 17 15:11 test_dir_1/
d----- 2 kw202020234 kw202020234 4096 Apr 17 15:11 test_dir_2/
-rw-rw-r-- 1 kw202020234 kw202020234 0 Apr 17 15:11 test_file1
-rw-rw-r-- 1 kw202020234 kw202020234 0 Apr 17 15:11 test_file2
kw202020234@ubuntu:~/system_programming_1/1-3$ ./cli dir -e | tee cli.out | ./srv
Error: invalid option
kw202020234@ubuntu:~/system_programming_1/1-3$ ./cli dir ./not_exist_path | tee cli.out | ./srv
LIST
Error : No such file or directory
```

다음은 LIST(cli: dir) 명령어 구현 결과입니다. 옵션을 받지 않으며, 인자를 받지 않거나 하나만 받습니다.

인자를 주지 않고 실행하면 LIST는 ls -al 과 같은 동작을 수행하기 때문에 .으로 시작하는 파일을 포함해 파일의 세부 정보를 한 줄에 한 파일씩 출력합니다.

존재하지 않는 -e 옵션을 주면 이에 맞는 에러를 출력합니다.

인자로 존재하지 않는 경로를 주면 이에 맞는 에러를 출력합니다.

```
kw202020234@ubuntu:~/system_programming_1/1-3$ ./cli dir Makefile | tee cli.out | ./srv
LIST
-rw-rw-r-- 1 kw202020234 kw202020234 90 Apr 16 09:24 Makefile
```

인자로 파일 이름을 주면, 이에 대한 세부 정보를 출력합니다.

```
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli pwd | tee cli.out | ./srv
"/home/kw2020202034/system_programming_1/1-3" is current directory
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli pwd -e | tee cli.out | ./srv
Error: invalid option
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli pwd arg | tee cli.out | ./srv
Error: argument is not required
```

다음은 PWD(cli: pwd) 명령어를 구현한 결과입니다. 옵션과 인자를 받지 않습니다.

인자를 주지 않고 실행하면 현재 working directory 가 출력되는 것을 볼 수 있습니다.

존재하지 않는 옵션 -e 를 주면 이에 대한 에러 메시지를 출력합니다.

인자를 주게 되면, 이에 대한 에러가 출력됩니다.

```
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli cd -e .. | tee cli.out | ./srv
Error: invalid option
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli cd test_dir_1 | tee cli.out | ./srv
CWD test_dir_1
"/home/kw2020202034/system_programming_1/1-3/test_dir_1" is current directory
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli cd .. | tee cli.out | ./srv
CDUP
"/home/kw2020202034/system_programming_1" is current directory
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli cd ./not_exist_path | tee cli.out | ./srv
Error: directory not found
```

다음은 CWD, CDUP(cli: cd, cd ..)의 구현 결과입니다. CWD 와 CDUP 모두 옵션은 없지만, CWD 는 인자를 하나만 받고, CDUP 는 인자를 받지 않습니다.

CDUP 에 존재하지 않는 옵션 -e 를 주면 에러를 출력합니다.

인자로 test_dir_1 를 주면, 현재 working directory 를 test_dir_1 로 바꿔줌을 확인할 수 있습니다. 다만, 이는 프로세스의 working directory 를 바꾼 것이지, 터미널의 working directory 를 바꾼 것이 아니므로 터미널에선 확인할 수 없습니다.

CDUP 에 인자를 주지 않고 실행하면 parent directory 로 current working directory 가 변경되었음을 확인할 수 있습니다.

CWD 에 인자로 존재하지 않는 경로를 주면 에러를 출력합니다.

```
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli mkdir -e new_dir | tee cli.out | ./srv
Error: invalid option
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli mkdir new_dir1 | tee cli.out | ./srv
MKD new_dir1
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli mkdir new_dir2 new_dir3 | tee cli.out | ./srv
MKD new_dir2
MKD new_dir3
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli mkdir new_dir1 | tee cli.out | ./srv
Error: cannot create directory 'new_dir1': File exists
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli mkdir | tee cli.out | ./srv
Error: argument is required
```

다음은 MKD 명령어(cli: mkdir)의 구현 결과입니다. 옵션은 받지 않고, 인자는 한 개 이상 받습니다.

없는 옵션 -e 를 주면 에러를 출력함을 볼 수 있습니다.

인자로 new_dir1 하나를 주면 new_dir1 이름을 가진 디렉토리 하나를 생성하고 결과를 출력합니다.

인자로 만들 디렉토리 이름 여러 개(사진에선 2 개)를 주면, 각 줄마다 성공 여부를 출력합니다.

이미 존재하는 이름으로 디렉토리를 만들려고 하면 에러를 출력합니다.

인자를 주지 않으면 이에 대한 에러를 출력합니다.

```
kw2020202034@ubuntu:~/system_programming_1/1-3$ touch test1.dat test2.dat test3.dat
kw2020202034@ubuntu:~/system_programming_1/1-3$ ls -al *.dat
-rw-rw-r-- 1 kw2020202034 kw2020202034 0 Apr 17 15:25 test1.dat
-rw-rw-r-- 1 kw2020202034 kw2020202034 0 Apr 17 15:25 test2.dat
-rw-rw-r-- 1 kw2020202034 kw2020202034 0 Apr 17 15:25 test3.dat
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli delete test1.dat | tee cli.out | ./srv
DELE test1.dat
kw2020202034@ubuntu:~/system_programming_1/1-3$ ls -al test1.dat
ls: cannot access 'test1.dat': No such file or directory
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli delete test2.dat test3.dat | tee cli.out | ./srv
DELE test2.dat
DELE test3.dat
kw2020202034@ubuntu:~/system_programming_1/1-3$ ls -al *.dat
ls: cannot access '*.dat': No such file or directory
kw2020202034@ubuntu:~/system_programming_1/1-3$ touch test.dat
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli delete -e test.dat | tee cli.out | ./srv
Error: invalid option
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli delete not_exist_file test.dat | tee cli.out | ./srv
Error: failed to delete 'not_exist_file'
DELE test.dat
```

DELE(cli: delete)의 구현 결과입니다. 옵션은 없고, 인자는 한 개 이상 받습니다.

테스트를 위해 test1.dat, test2.dat, test3.dat 를 만들고 ls -la 로 확인하였습니다.

인자로 파일명 하나를 주었을 때 해당 파일을 삭제하고 결과를 출력함을 볼 수 있습니다(ls -al 로 확인).

인자로 파일명 2 개를 주면, 차례로 실행 결과를 출력합니다.

존재하지 않는 -e 옵션을 주면 에러를 출력합니다.

인자로 존재하지 않는 파일과 존재하는 파일을 차례로 주면, 우선 존재하지 않는 파일에 대한 에러를 출력 후, 다음 줄에 존재하는 파일을 지운 결과를 출력합니다.

```
kw2020202034@ubuntu:~/system_programming_1/1-3$ mkdir new_dir_1 new_dir_2 new_dir_3
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli rmdir new_dir_1 | tee cli.out | ./srv
RMD new_dir_1
kw2020202034@ubuntu:~/system_programming_1/1-3$ ls -al new_dir_1
ls: cannot access 'new_dir_1': No such file or directory
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli rmdir new_dir_1 | tee cli.out | ./srv
Error: failed to remove 'new_dir_1'
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli rmdir new_dir_2 new_dir_3 | tee cli.out | ./srv
RMD new_dir_2
RMD new_dir_3
kw2020202034@ubuntu:~/system_programming_1/1-3$ mkdir new_dir
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli rmdir not_exist_dir new_dir | tee cli.out | ./srv
Error: failed to remove 'not_exist_dir'
RMD new_dir
kw2020202034@ubuntu:~/system_programming_1/1-3$ ./cli rmdir | tee cli.out | ./srv
Error: argument is required
```

RMD(cli: rmdir)의 실행 결과입니다. 옵션은 없고, 인자는 한 개 이상만 받습니다.

위의 DELE 예시와 비슷하게 test 했는데, 인자를 한 개 이상 주면 이에 대한 실행 결과를 한 줄에 한 파일씩 출력합니다. 그리고 이미 지워진, 즉 존재하지 않는 디렉토리를 지우려고 하면 에러를 출력합니다.

마찬가지로 존재하지 않는 디렉토리와 존재하는 디렉토리를 같이 인자로 주면 삭제에 실패한 디렉토리에 대해선 에러를 출력하고, 성공한 디렉토리는 결과를 출력합니다.

인자가 주어지지 않으면 이에 대한 에러를 출력합니다.

```
kw202020234@ubuntu:~/system_programming_1/1-3$ touch a b
kw202020234@ubuntu:~/system_programming_1/1-3$ ls -al a b
-rw-rw-r-- 1 kw202020234 kw202020234 0 Apr 17 15:28 a
-rw-rw-r-- 1 kw202020234 kw202020234 0 Apr 17 15:28 b
kw202020234@ubuntu:~/system_programming_1/1-3$ ./cli rename a b | tee cli.out | ./srv
Error: name to change already exists
kw202020234@ubuntu:~/system_programming_1/1-3$ ./cli rename a c | tee cli.out | ./srv
RNFR a
RNT0 c
kw202020234@ubuntu:~/system_programming_1/1-3$ ls -al a b c
ls: cannot access 'a': No such file or directory
-rw-rw-r-- 1 kw202020234 kw202020234 0 Apr 17 15:28 b
-rw-rw-r-- 1 kw202020234 kw202020234 0 Apr 17 15:28 c
kw202020234@ubuntu:~/system_programming_1/1-3$ ./cli rename | tee cli.out | ./srv
Error: two argumens are required
```

RNFR & RNT0(cli: rename)의 구현 결과입니다. 각 명령어는 옵션 없이 인자를 하나씩 받는데, cli 에서 rename 을 변환시킬 때 인자 2 개(예: a, b)가 있었다면 RNFO a RNT0 b 로 변환시킵니다.

a, b 파일을 만들어 이중 a 파일의 이름을 b 로 바꾸려고 하면 이미 존재하는 이름이기 때문에 에러를 출력함을 볼 수 있습니다.

c 라는 이름의 파일은 없기 때문에 a 를 c 로 이름을 바꾸면 명령어를 출력함을 볼 수 있습니다.

인자를 주지 않으면 이에 맞는 에러를 출력합니다.

```
kw202020234@ubuntu:~/system_programming_1/1-3$ ./cli quit | tee cli.out | ./srv
QUIT success
kw202020234@ubuntu:~/system_programming_1/1-3$ ./cli quit -e | tee cli.out | ./srv
Error: invalid option
kw202020234@ubuntu:~/system_programming_1/1-3$ ./cli quit arg | tee cli.out | ./srv
Error: argument is not required
```

QUIT(cli: quit) 명령어의 구현 결과입니다. 옵션, 인자를 받지 않아 받게 되면 에러를 출력함을 볼 수 있고, 인자를 주지 않으면 성공함을 출력하였습니다.

```
kw202020234@ubuntu:~/system_programming_1/1-3$ ./cli unknown_cmd | tee cli.out | ./srv
Unknown command
kw202020234@ubuntu:~/system_programming_1/1-3$ ./cli | tee cli.out | ./srv
Error: cli needs instruction
Unknown command
```

cli 의 인자로 잘못된 명령어가 들어오거나 인자가 들어오지 않으면 이에 대한 에러를 출력하도록 하였습니다.

고찰

ls 명령어를 저번 과제보다 더 자세히 구현해야 해서 시간을 많이 소모했습니다. -l 옵션을 주었을 때 출력하는 파일의 정보를 어떻게 얻어야 하나 고민했는데, 이론 강의 자료를 복습하던 중 stat()의 내용을 찾아 이를 이용해 해결했습니다.

rename system call 이 기본적으로 newpath 가 이미 존재하면 이를 덮어쓰기 하는데, 과제의 내용을 구현하기 위해 고민하던 중 stat() 함수는 인자로 주어진 파일이 존재하지 않으면 -1 을 반환하는 성질을 이용해 newpath 가 존재하는 지를 확인하였습니다.