

# 시스템 프로그래밍 실습

## 5차 과제



실습 일시 : 화 1,2

담당 교수님 : 김태석 교수님

학번 : 2013722095

이름 : 최재은

실습 번호 : FTP1

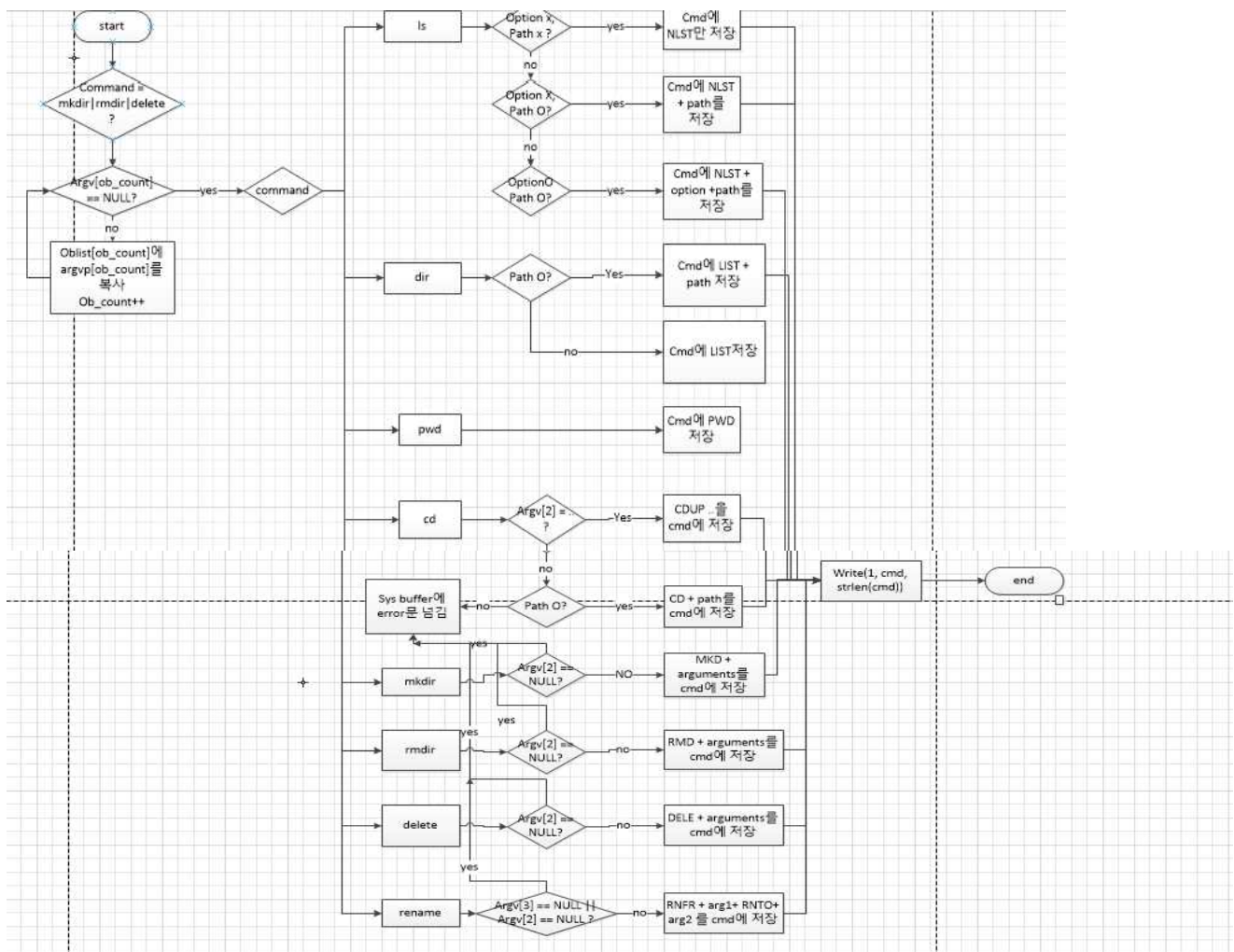
## # FTP1

### ■ Introduction

파일 전송 네트워크인 FTP를 간략하게 구현해 보고 이 프로토콜 간의 통신이 어떤 방식으로 이루어지는지 공부하고 이해한다. 이를 통해 기본적인 FTP를 이해하고 난 후 포트 넘버, IP 주소 등을 어떻게 적용할지 생각해본다.

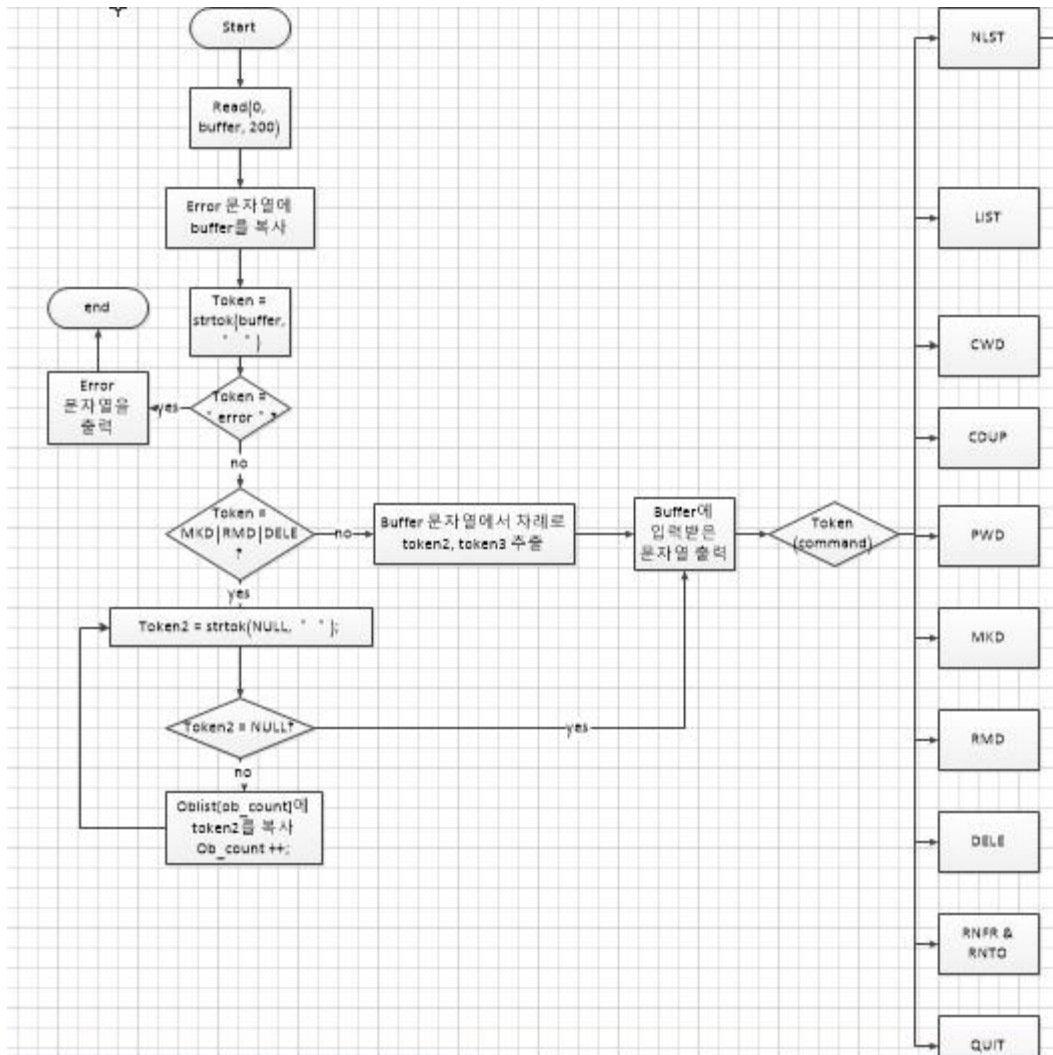
### ■ Flow Chart

#### 1. cli.c

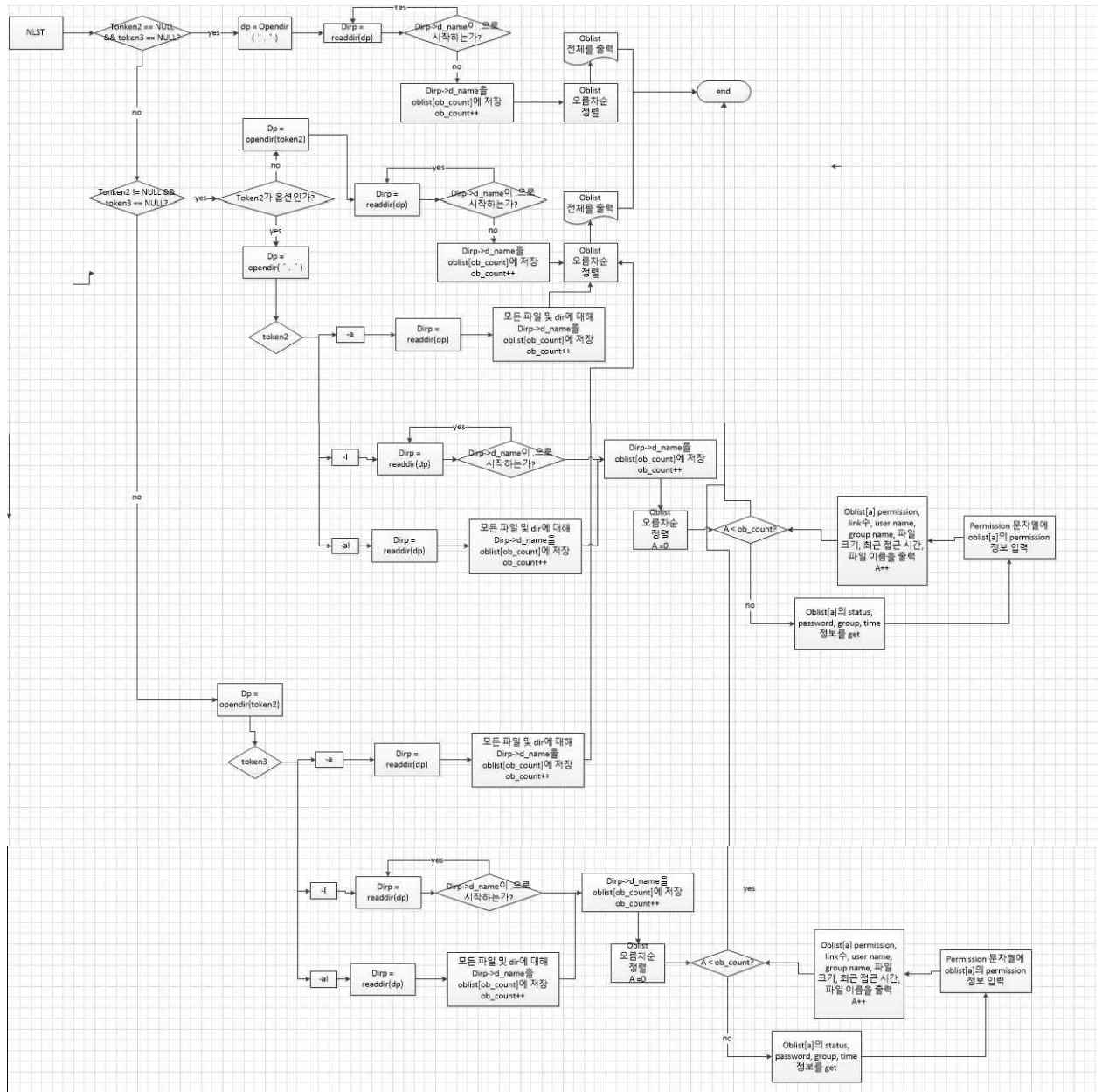


## 2. srv.c

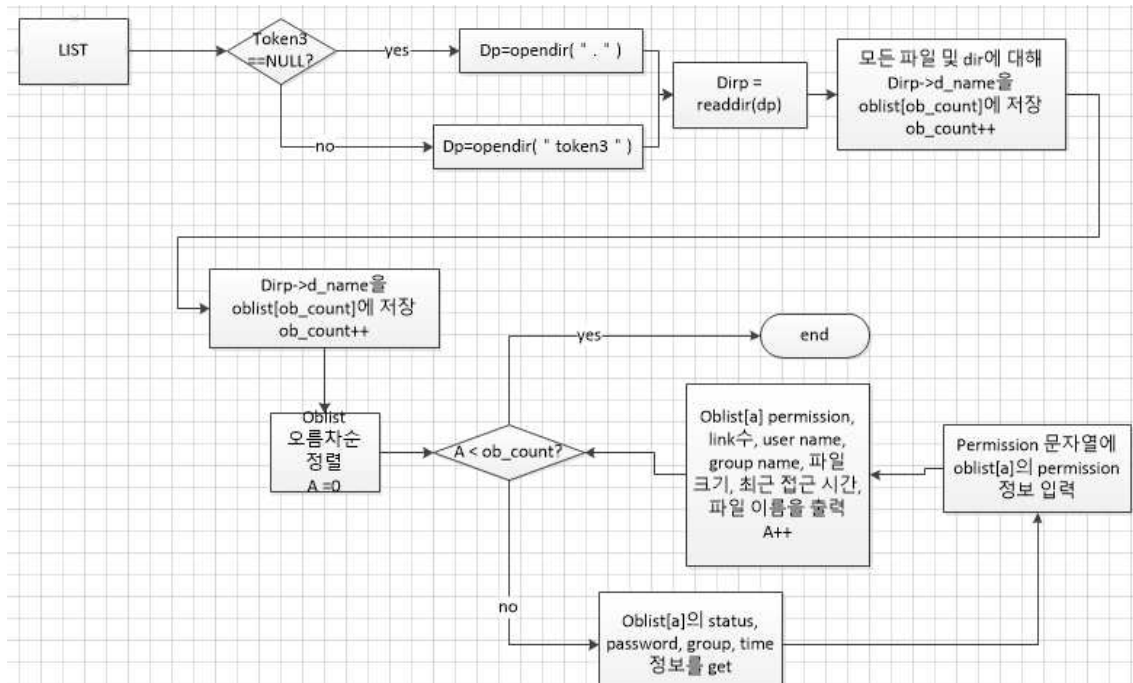
\*각 command진입 전까지



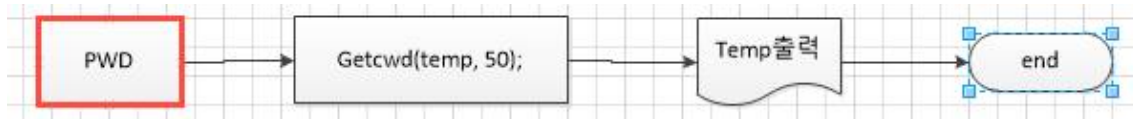
## \*NLST



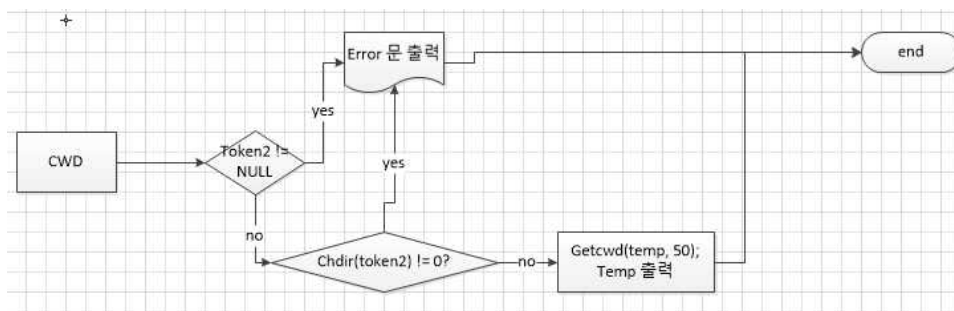
## \*LIST



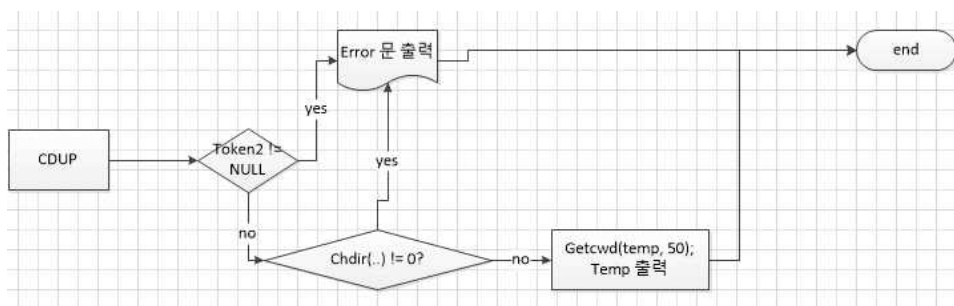
## \*PWD



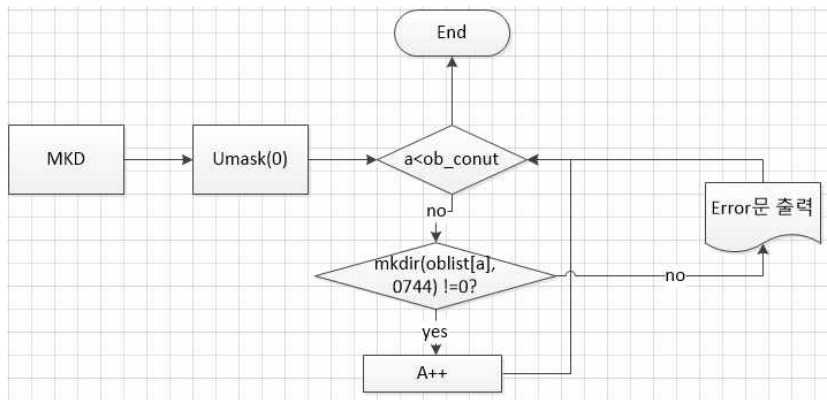
## \*CWD



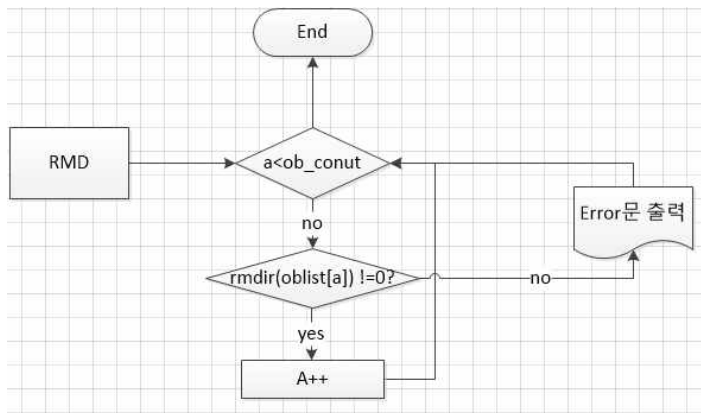
## \*CDUP



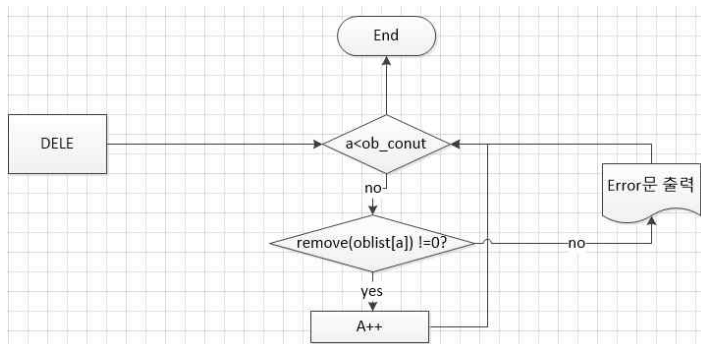
## \*MKD



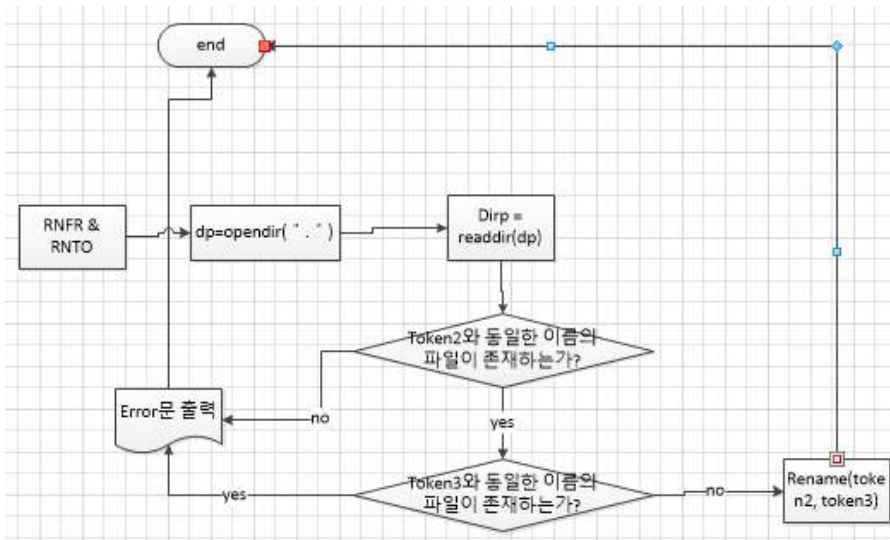
## \*RMD



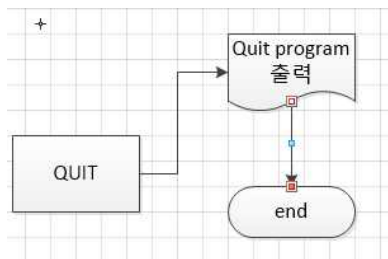
## \*DELE



## \*RNFR & RNT0



## \*QUIT



## ■ Pseudo code

### 1. cli.c

```
if( command가 mkdir이나 rmdir이나 delete)
{
    while(모든 argument에 대하여)
        파일 이름을 oblist 배열에 복사해둠
    }
else if(command가 ls면)
{
    ls를 cmd문자열에 복사
    if(세 옵션에 대하여)
    {
        옵션을 문자열에 복사
        if(path가 있다면)
        { if(path가 열리면) path를 문자열에 복사
          else 에러 메시지를 sys buffer에 올리고 종료
        }
    }
}
else if(command가 dir)
{
    dir을 문자열에 복사

    if(path가 열리면) path를 문자열에 복사
    else 에러 메시지를 system buffer에 올리고 종료
}
else if(command가 pwd)
{ cmd 문자열에 PWD 복사 }
if(command가 cd)
{
    if(추가적인 path없으면) 에러 메시지를 sys buffer에 올리고 종료
    elseif(추가적인 path = "..") CDUP ..을 cmd문자열에 복사
    else CWD + path를 cmd문자열에 복사
}
else if(command가 delete)
```



```

{
    if(추가적인 argument가 없으면) 에러 메시지를 sys buf에 올리고 종료
    else
        { RMD와 추가적인 argument를 cmd문자열에 복사 }
}

else if(command가 mkdir)
{
    if(추가적인 argument가 없으면) 에러 메시지를 sys buf에 올리고 종료
    else
        { MKD와 추가적인 argument를 cmd문자열에 복사}
}
else if(command가 rmdir)
{
    if(추가적인 argument가 없으면) 에러 메시지를 sys buf에 올리고 종료
    else
        { RMD와 추가적인 argument를 cmd문자열에 복사}
}

else if(command가 rename)
{
    if(추가적인 argument가 2개가 아니면) 에러 메시지를 sys buf에 올리고 종료
    else
        { RNFR arg1 RNTD arg2를 cmd문자열에 복사}
}

else if(command가 quit)
{QUIT를 cmd문자열에 복사}

else 그 외의 명령어에 대해서
    에러 메시지를 system buffer에 올리고 종료

cmd 문자열을 system buffer에 올리고 종료

```

## 2. srv.c

read함수로 buffer 문자열에 system buffer에 있는 data를 받음

if(문자열이 error로 시작하면) buffer문자열을 출력하고 종료

token을 사용하여 buffer를 command, option, path로 분할

if(command가 MKD, RMD, DELE이면)

```
{
    while(모든 additional argument에 대해) argument들을 문자열에 복사
}
```

if(command = NLST)

```
{
if(option, path들다 없는 경우)
{
```

현재 디렉토리 open

while(모든 file 및 dir에 대하여)

.으로 시작하는 대상을 제외하고 oblist 문자열에 복사

oblist를 오름차순으로 솔팅

oblist 출력

```
}
```

if(option 0, path X)

```
{
    현재 디렉토리를 open
    if(path만 있는 경우)
    {
        while(모든 파일에 대하여)
            .으로 시작하는 대상을 제외하고 oblist 문자열에 복사
        oblist를 오름차순으로 솔팅
        oblist를 출력
    }
}
```

else // option만 있는 경우

```
{
    if(옵션이 -a)
    {
        while(모든 file 및 dir에 대하여)
            모든 대상을 oblist 문자열에 복사
        oblist를 오름차순으로 솔팅
        oblist를 출력
    }
}
```

```

if(옵션이 -l)
{
    while(모든 file 및 dir에 대하여)
        .로 시작하는 대상을 제외한 모든 대상을 oblist 문자열에 복사
        oblist를 오름차순으로 솔팅
        for(oblist의 모든 data(file 이름)들을 대상으로)
        {
            현재 data의 status, password, group, time 정보를 get
            data의 permission 정보를 permission 문자열에 저장

            permission, link 수, user name, group name, 파일 크기,
            최근 사용일, 대상의 이름을 출력
            if(현재 data를 이름으로 하는 대상이 directory라면)
                대상의 이름 뒤에 /를 붙임
        }
    }
}
if(옵션이 -al)
{
    while(모든 file 및 dir에 대하여)
        모든 대상을 oblist 문자열에 복사
        oblist를 오름차순으로 솔팅
        for(oblist의 모든 data(file 이름)들을 대상으로)
        {
            현재 data의 status, password, group, time 정보를 get
            data의 permission 정보를 permission 문자열에 저장

            permission, link 수, user name, group name, 파일 크기,
            최근 사용일, 대상의 이름을 출력
            if(현재 data를 이름으로 하는 대상이 directory라면)
                대상의 이름 뒤에 /를 붙임
        }
    }
}

```

```

}
if(option O, path O)
{
    path로 입력받은 directory를 open
    if(옵션이 -a)

```

```

{
    while(모든 file 및 dir에 대하여)
        모든 대상을 oblist 문자열에 복사
        oblist를 오름차순으로 솔팅
        oblist를 출력
    }
if(옵션이 -l)
{
    while(모든 file 및 dir에 대하여)
        .로 시작하는 대상을 제외한 모든 대상을 oblist 문자열에 복사
        oblist를 오름차순으로 솔팅
        for(oblist의 모든 data(file 이름)들을 대상으로)
        {
            현재 data의 status, password, group, time 정보를 get
            data의 permission 정보를 permission 문자열에 저장

            permission, link 수, user name, group name, 파일 크기,
            최근 사용일, 대상의 이름을 출력
            if(현재 data를 이름으로 하는 대상이 directory라면)
                대상의 이름 뒤에 /를 붙임
        }
    }
if(옵션이 -al)
{
    while(모든 file 및 dir에 대하여)
        모든 대상을 oblist 문자열에 복사
        oblist를 오름차순으로 솔팅
        for(oblist의 모든 data(file 이름)들을 대상으로)
        {
            현재 data의 status, password, group, time 정보를 get
            data의 permission 정보를 permission 문자열에 저장

            permission, link 수, user name, group name, 파일 크기,
            최근 사용일, 대상의 이름을 출력
            if(현재 data를 이름으로 하는 대상이 directory라면)
                대상의 이름 뒤에 /를 붙임
        }
    }
}

```

```

if(command= LIST)
{
if (path O) path open
else 현재 디렉토리 open
    while(모든 file 및 dir에 대하여)
        모든 대상을 oblist 문자열에 복사
        oblist를 오름차순으로 솔팅
        for(oblist의 모든 data(file 이름)들을 대상으로)
        {
            현재 data의 status, password, group, time 정보를 get
            data의 permission 정보를 permission 문자열에 저장

            permission, link 수, user name, group name, 파일 크기,
            최근 사용일, 대상의 이름을 출력
            if(현재 data를 이름으로 하는 대상이 directory라면)
                대상의 이름 뒤에 /를 붙임
        }
}

```

```

if(command = PWD)
{ temp에 현재 경로를 저장;
  temp를 출력}

```

```

if(command = CWD)
{
if(chdir(path) != 0) 에러 메시지 띄우고 종료

```

```

temp에 현재 경로 저장
temp를 출력
}

```

```

if(command = CDUP)
{
cwd를 상위 디렉토리로 이동
현재 경로를 temp에 저장
temp를 출력
}

```

```

if(command = MKD)
{
    umask(0);
    for(ob_conut에 대하여)
    {
        oblist의 이름을 가진 directory 생성, 권한은 0744
    }
}

if(commmand = RMD)
{
    for(모든 ob_count에 대하여)
        { oblist의 이름을 갖는 대상을 remove}
}

if(command = DELE)
{
    for(모든 ob_count에 대하여)
        { oblist의 이름을 갖는 대상을 rmdir}
}

if(command = RNFR)
{
    현재 directory open
    while(모든 파일 및 dir에 대하여)
        buffer로부터 추출한 oldname과 동일한 파일 검색
        if(찾으면) flag = 1
        if(flag = 0) 에러메세지 띄우고 종료
        buffer로부터 newname 추출
        if(동일한 이름을 갖는 file, dir이 없으면) newname으로 rename
}

if(command = QUIT)
{ QuiT program 출력 후 종료}

```

## ■ Result

### 1. ls

```
jaaen1113@ubuntu:~/SP/ftp1$ ./cli ls|tee cli.out|./srv
NLST
backup cli cli.c cli.out Makefile
recup_dir.1 srv srv.c
```

### 2. ls -a / -l / -al

```
jaaen1113@ubuntu:~/SP/ftp1$ ./cli ls -a|tee cli.out|./srv
NLST -a
. .. .srv.c.swp backup cli
cli.c cli.out Makefile recup_dir.1 srv
srv.c
```

```
jaaen1113@ubuntu:~/SP/ftp1$ ./cli ls -l|tee cli.out|./srv
NLST -l
drwxrwxr-x 2 jaaen1113 jaaen1113 4096 4 28 19:12 backup/
-rwxrwxr-x 1 jaaen1113 jaaen1113 12749 4 28 19:59 cli
-rw-rw-r-- 1 jaaen1113 jaaen1113 7676 4 28 19:59 cli.c
-rw-rw-r-- 1 jaaen1113 jaaen1113 9 4 28 14:35 cli.out
-rw-rw-r-- 1 jaaen1113 jaaen1113 114 4 28 14:32 Makefile
drwxr-xr-x 2 root root 4096 4 28 12:40 recup_dir.1/
-rwxrwxr-x 1 jaaen1113 jaaen1113 34067 4 28 20:0 srv
-rw-rw-r-- 1 jaaen1113 jaaen1113 31657 4 28 20:1 srv.c
```

```
jaaen1113@ubuntu:~/SP/ftp1$ ./cli ls -la|tee cli.out|./srv
NLST -al
drwxrwxr-x 4 jaaen1113 jaaen1113 4096 4 28 20:5 ./
drwxrwxr-x 7 jaaen1113 jaaen1113 4096 4 28 12:39 ../
-rw-r--r-- 1 jaaen1113 jaaen1113 16384 4 28 19:32 .srv.c.swp
drwxrwxr-x 2 jaaen1113 jaaen1113 4096 4 28 19:12 backup/
-rwxrwxr-x 1 jaaen1113 jaaen1113 12749 4 28 19:59 cli
-rw-rw-r-- 1 jaaen1113 jaaen1113 7676 4 28 19:59 cli.c
-rw-rw-r-- 1 jaaen1113 jaaen1113 10 4 28 14:35 cli.out
-rw-rw-r-- 1 jaaen1113 jaaen1113 114 4 28 14:32 Makefile
drwxr-xr-x 2 root root 4096 4 28 12:40 recup_dir.1/
-rwxrwxr-x 1 jaaen1113 jaaen1113 34067 4 28 20:0 srv
-rw-rw-r-- 1 jaaen1113 jaaen1113 31657 4 28 20:1 srv.c
```

### 3. ls + path

```
jaaen1113@ubuntu:~/SP/ftp1$ ./cli ls /home/jaaen1113/SP|tee cli.out|./srv
NLST /home/jaaen1113/SP
ftp1 prac1 prac2 prac3 splab
```

#### 4. ls + option(-a -l -al) + path

- 현재 세 번째 토큰이 경로인 것을 잘 확인할 수 있고 옵션 별로 파일등을 잘 보여주는 것을 확인하였습니다. 다만 세 번째 토큰을 통해 opendir을 하면 코어 덤프가 일어나는데 원인이 무엇인지 모르겠습니다. 다른 부분의 동작 여부를 확인하기 위해 현재 디렉토리를 열도록 설정했습니다.

```
jaeen1113@ubuntu:~/SP/ftp1$ ./cli ls -a /home|tee cli.out|./srv
NLST -a /home
/home.  ..  .srv.c.swp  backup  cli
cli.c  cli.out  Makefile  recup_dir.1  srv
srv.c
```

```
jaeen1113@ubuntu:~/SP/ftp1$ ./cli ls -l /home|tee cli.out|./srv
NLST -l /home
/homedrwxrwxr-x 2 jaeen1113 jaeen1113 4096 4 28 19:12 backup/
-rwxrwxr-x 1 jaeen1113 jaeen1113 12749 4 28 19:59 cli
-rw-rw-r-- 1 jaeen1113 jaeen1113 7676 4 28 19:59 cli.c
-rw-rw-r-- 1 jaeen1113 jaeen1113 14 4 28 14:35 cli.out
-rw-rw-r-- 1 jaeen1113 jaeen1113 114 4 28 14:32 Makefile
drwxr-xr-x 2 root root 4096 4 28 12:40 recup_dir.1/
-rwxrwxr-x 1 jaeen1113 jaeen1113 34067 4 28 20:0 srv
-rw-rw-r-- 1 jaeen1113 jaeen1113 31657 4 28 20:8 srv.c
```

```
jaeen1113@ubuntu:~/SP/ftp1$ ./cli ls -la /home|tee cli.out|./srv
NLST -al /home
/homedrwxrwxr-x 4 jaeen1113 jaeen1113 4096 4 28 20:8 ./
drwxrwxr-x 7 jaeen1113 jaeen1113 4096 4 28 12:39 ../
-rw-r--r-- 1 jaeen1113 jaeen1113 16384 4 28 19:32 .srv.c.swp
drwxrwxr-x 2 jaeen1113 jaeen1113 4096 4 28 19:12 backup/
-rwxrwxr-x 1 jaeen1113 jaeen1113 12749 4 28 19:59 cli
-rw-rw-r-- 1 jaeen1113 jaeen1113 7676 4 28 19:59 cli.c
-rw-rw-r-- 1 jaeen1113 jaeen1113 15 4 28 14:35 cli.out
-rw-rw-r-- 1 jaeen1113 jaeen1113 114 4 28 14:32 Makefile
drwxr-xr-x 2 root root 4096 4 28 12:40 recup_dir.1/
-rwxrwxr-x 1 jaeen1113 jaeen1113 34067 4 28 20:0 srv
-rw-rw-r-- 1 jaeen1113 jaeen1113 31657 4 28 20:8 srv.c
```



## 5. dir

- 위는 path 없이, 아래는 path 있음.

```
jaeen1113@ubuntu:~/SP/ftp1$ ./cli dir|tee cli.out|./srv
LIST
drwxrwxr-x 4 jaeen1113 jaeen1113 4096 4 28 23:14 ./
drwxrwxr-x 7 jaeen1113 jaeen1113 4096 4 28 12:39 ../
-rw----- 1 jaeen1113 jaeen1113 4096 4 28 23:13 .cli..swp
-rw-r--r-- 1 jaeen1113 jaeen1113 16384 4 28 19:32 .srv.c.swp
-rw-rw-r-- 1 jaeen1113 jaeen1113 0 4 28 20:47 aaa.c
drwxrwxr-x 2 jaeen1113 jaeen1113 4096 4 28 19:12 backup/
-rwxrwxr-x 1 jaeen1113 jaeen1113 12749 4 28 23:14 cli
-rw-rw-r-- 1 jaeen1113 jaeen1113 7592 4 28 23:14 cli.c
-rw-rw-r-- 1 jaeen1113 jaeen1113 6 4 28 23:12 cli.out
-rw-rw-r-- 1 jaeen1113 jaeen1113 114 4 28 14:32 Makefile
drwxr-xr-x 2 root root 4096 4 28 12:40 recup_dir.1/
-rwxrwxr-x 1 jaeen1113 jaeen1113 29971 4 28 23:12 srv
-rw-rw-r-- 1 jaeen1113 jaeen1113 30038 4 28 23:11 srv.c

jaeen1113@ubuntu:~/SP/ftp1$ ./cli dir /home|tee cli.out|./srv
LIST /home
drwxrwxr-x 4 jaeen1113 jaeen1113 4096 4 28 23:14 ./
drwxrwxr-x 7 jaeen1113 jaeen1113 4096 4 28 12:39 ../
drwxrwxr-x 7 jaeen1113 jaeen1113 4096 4 28 12:39 jaeen1113/
```

## 6. pwd

```
jaeen1113@ubuntu:~/SP/ftp1$ ./cli pwd|tee cli.out|./srv
PWD
/home/jaeen1113/SP/ftp1
```

## 7. cd (.. / 'path')

```
jaeen1113@ubuntu:~/SP/ftp1$ ./cli cd ..|tee cli.out|./srv
CDUP ..
/home/jaeen1113/SP<- cwd
```

```
jaeen1113@ubuntu:~/SP/ftp1$ ./cli cd /home/jaeen1113|tee cli.out|./srv
CWD /home/jaeen1113
/home/jaeen1113<- cwd
```

## 7. mkdir

```
jaeen1113@ubuntu:~/SP/ftp1$ ls
backup cli cli.c cli.out Makefile recup_dir.1 srv srv.c
jaeen1113@ubuntu:~/SP/ftp1$ ./cli mkdir 123 222 aaa bbb|tee cli.out|./srv
MKD 123 222 aaa bbb

jaeen1113@ubuntu:~/SP/ftp1$ ls
123 222 aaa backup bbb cli cli.c cli.out Makefile recup_dir.1 srv srv.c
```

## 8. delete

```
jaeen1113@ubuntu:~/SP/ftp1$ ./cli delete aaa bbb|tee cli.out|./srv
DELE aaa bbb

jaeen1113@ubuntu:~/SP/ftp1$ ls
123 222 backup cli cli.c cli.out Makefile recup_dir.1 srv srv.c
```

## 9. rmdir

```
jaeen1113@ubuntu:~/SP/ftp1$ ls
123 222 backup cli cli.c cli.out Makefile recup_dir.1 srv srv.c
jaeen1113@ubuntu:~/SP/ftp1$ ./cli rmdir 123 222|tee cli.out|./srv
RMD 123 222

jaeen1113@ubuntu:~/SP/ftp1$ ls
backup cli cli.c cli.out Makefile recup_dir.1 srv srv.c
```

## 10. rename

```
jaeen1113@ubuntu:~/SP/ftp1$ ls
123 backup cli cli.c cli.out Makefile recup_dir.1 srv srv.c
jaeen1113@ubuntu:~/SP/ftp1$ ./cli rename 123 aaa.c|tee cli.out|./srv
RNFR 123 RNT0 aaa.c

jaeen1113@ubuntu:~/SP/ftp1$ ls
aaa.c backup cli cli.c cli.out Makefile recup_dir.1 srv srv.c
```

## 11. quit

```
jaeen1113@ubuntu:~/SP/ftp1$ ./cli quit|tee cli.out|./srv
QUIT
Quit program
```

## ■ 결론 및 고찰

컴퓨터 네트워크 시간에 배웠던 프로토콜 간의 데이터 전송이 어떻게 실제로 이루어지는지를 직접 구현해보면서 심도있게 알아보는 계기가 되었다.

아직은 포트 넘버와 IP주소를 통해 연결하지 않지만 차후 write를 할 때에 1, 0대신에 port number+ipaddress를 넘겨주면서 해결하면 되지 않을까 싶다.