

2019. 5. 9.

Assignment # 3-2

금요일 - 이성원 교수님

2015722087 컴퓨터정보공학부

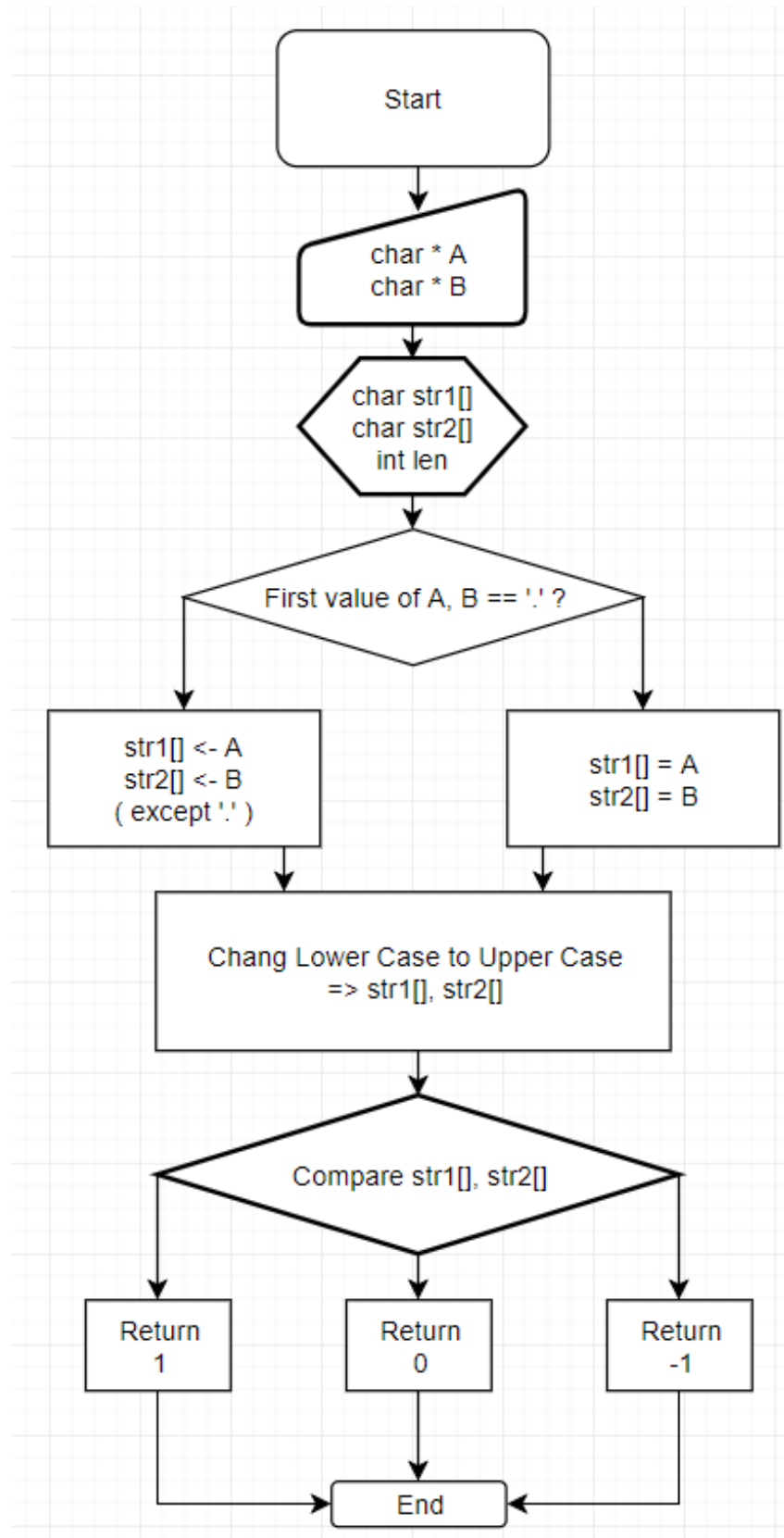
김민철

Introduction

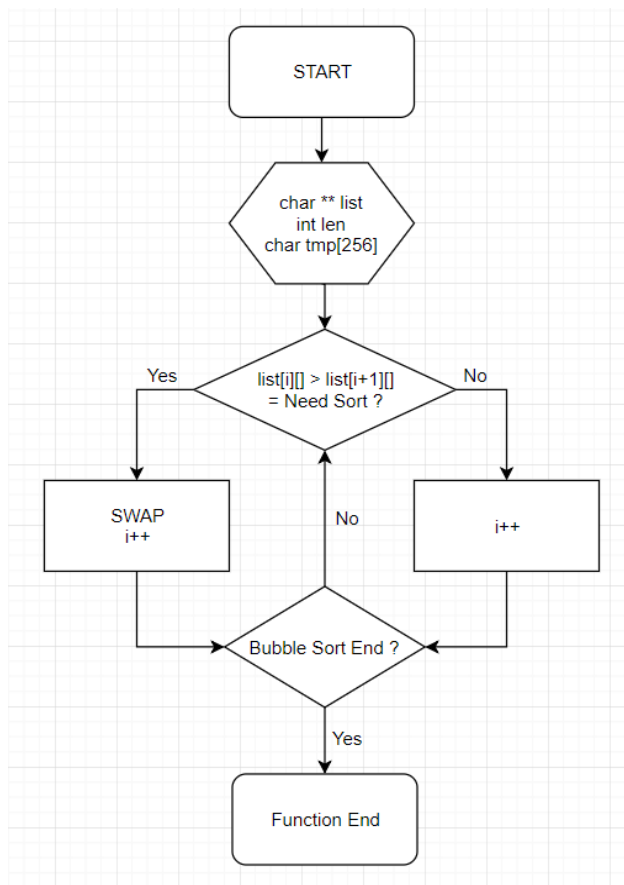
이번 과제에서는 3-1 과제에서 구현하였던 `html.js` 의 결과를 웹서버를 통해 다른 호스트의 브라우저에서 확인할 수 있게 하는 것이 목표이다. 서버 프로그램은 다른 호스트에서 웹 브라우저를 통해 접속하며, 웹 브라우저에서 보낸 `request` 메시지를 처리하여 결과를 `reponse` 메시지를 통해 전송하여 웹 브라우저에서 `html.js` 의 결과를 출력한다.

Flow Chart

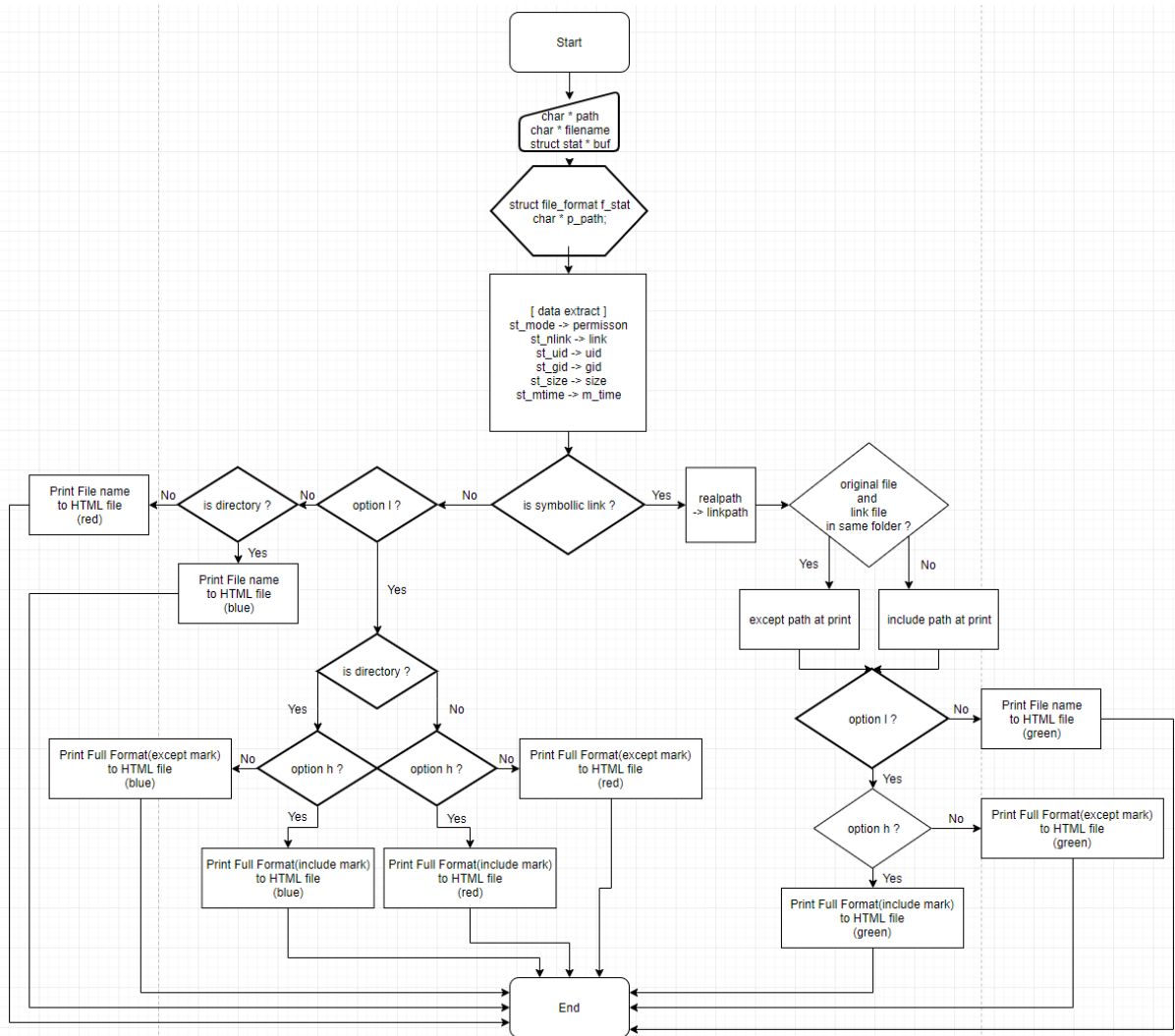
-int strcmp_i



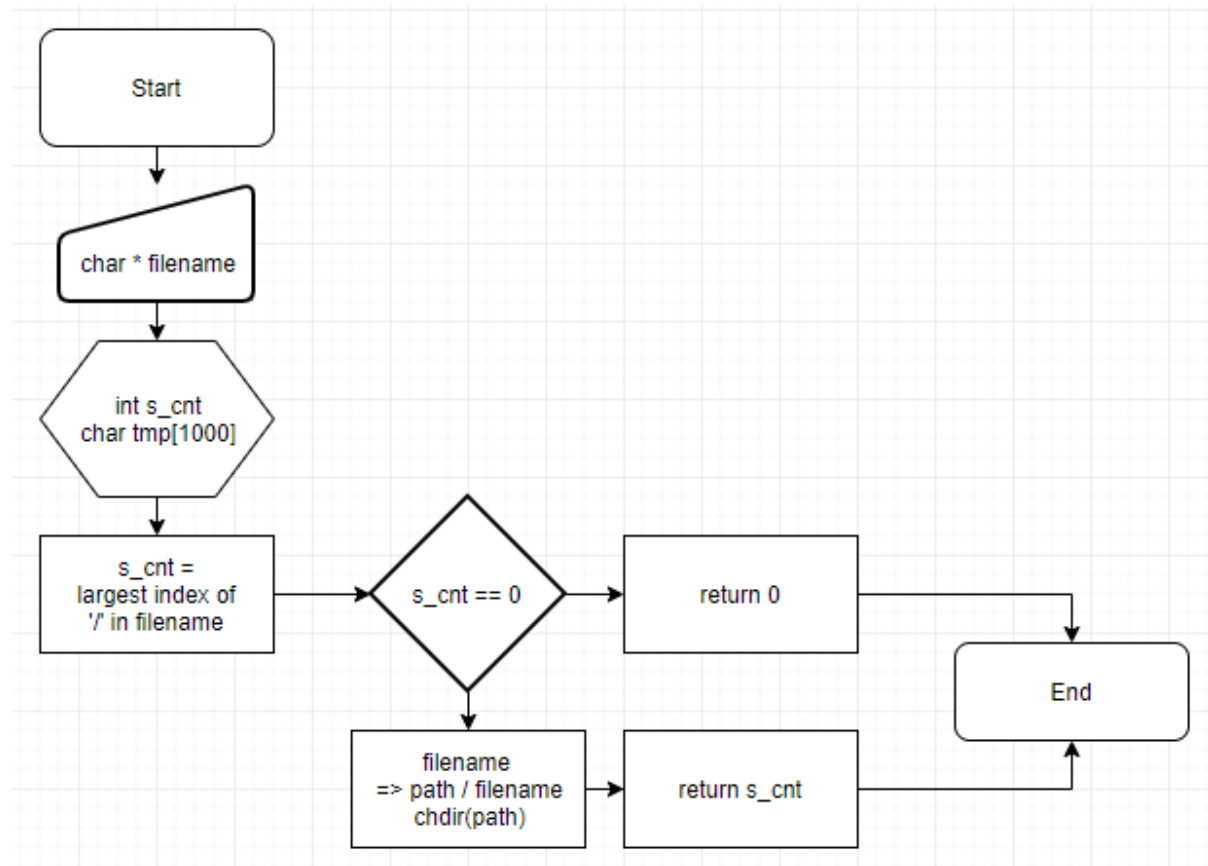
-void sort_list



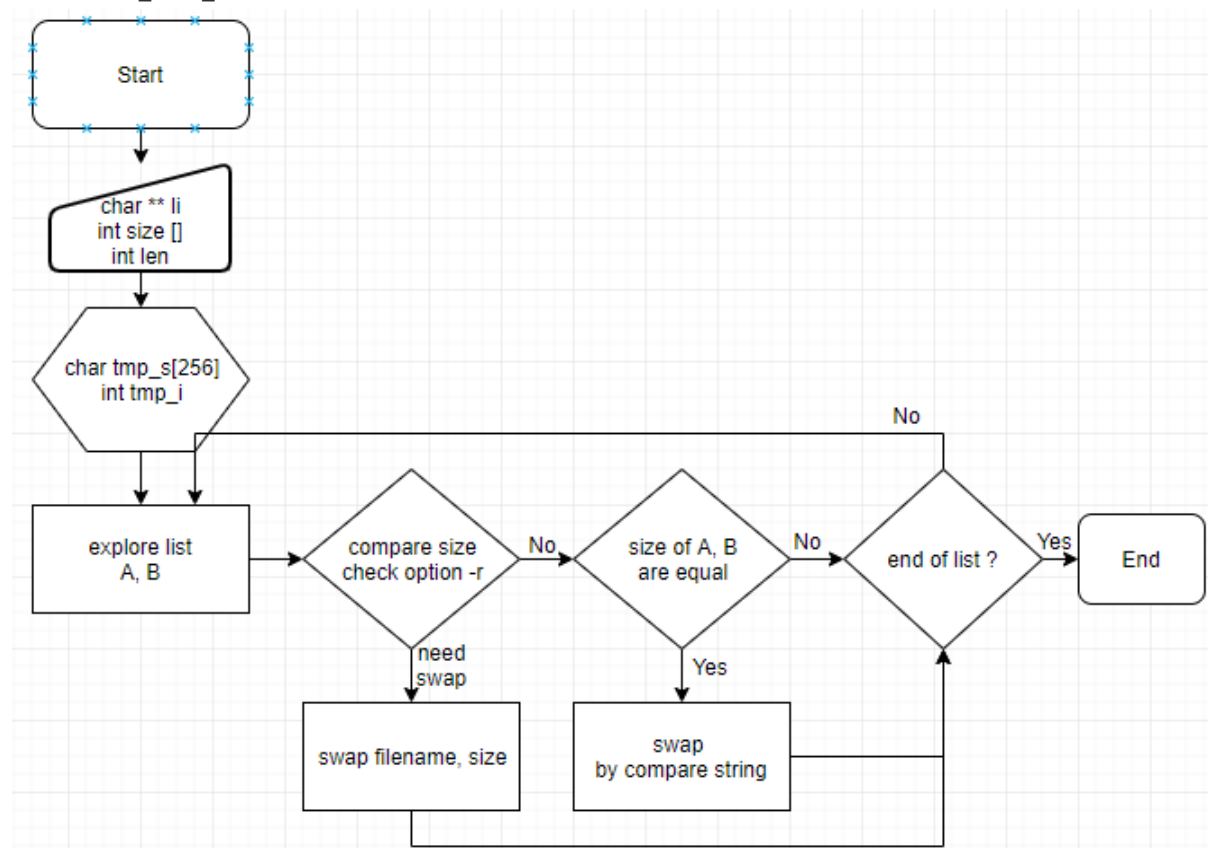
-void view_advanced_list



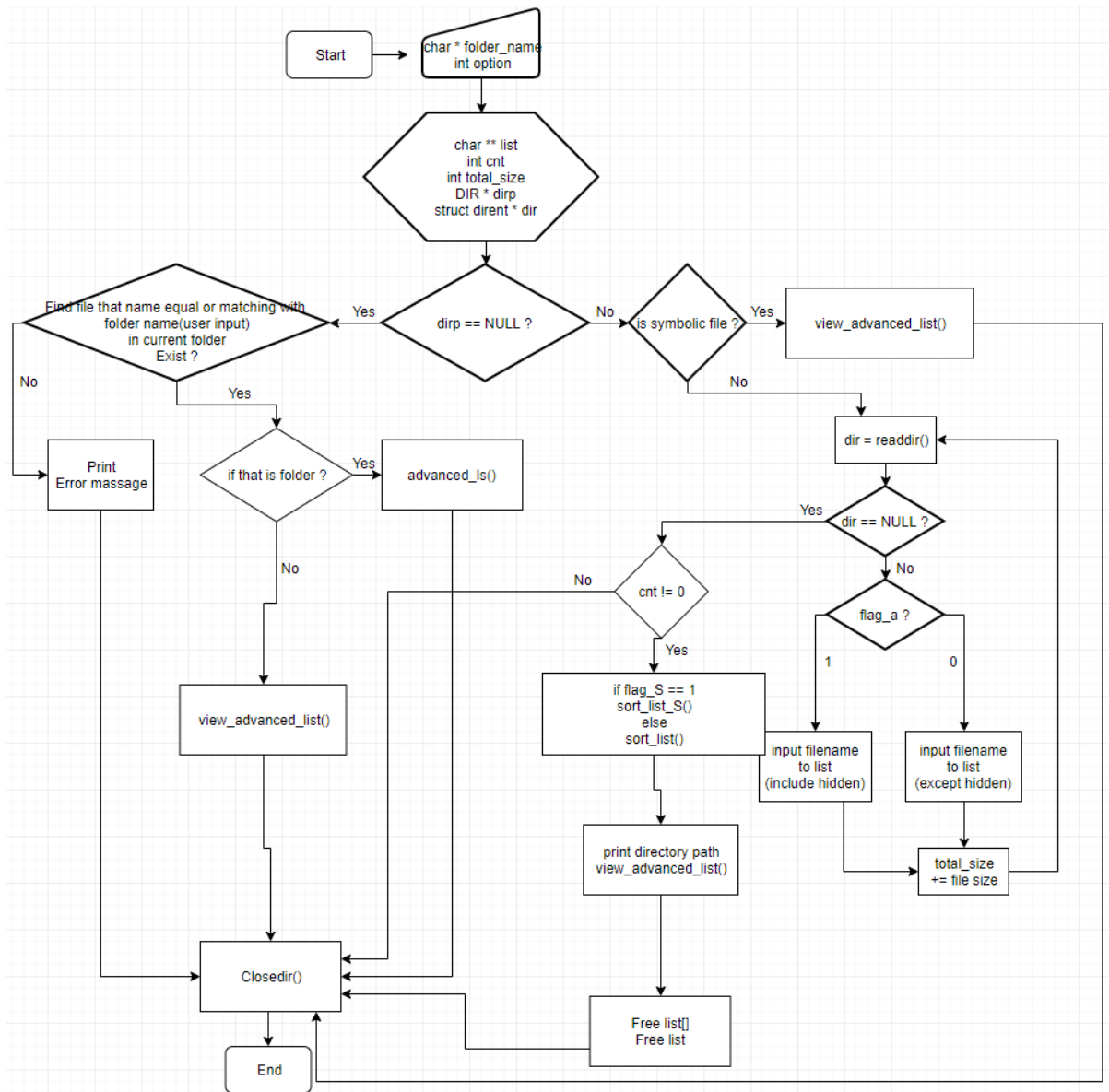
-int check_real_path



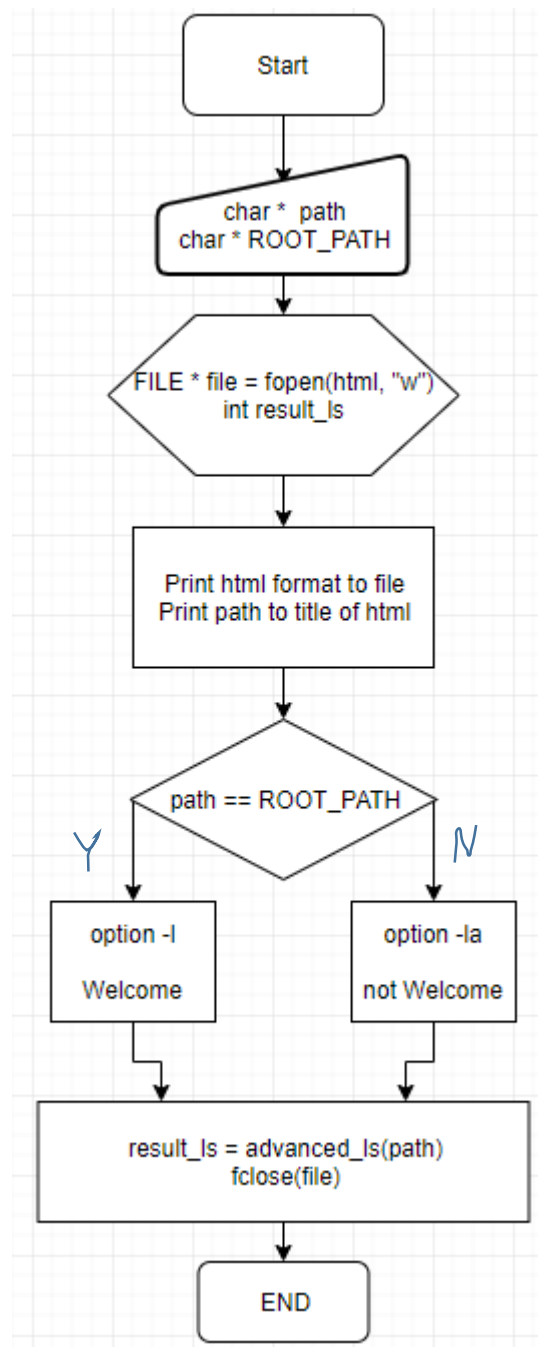
-void sort_list_S



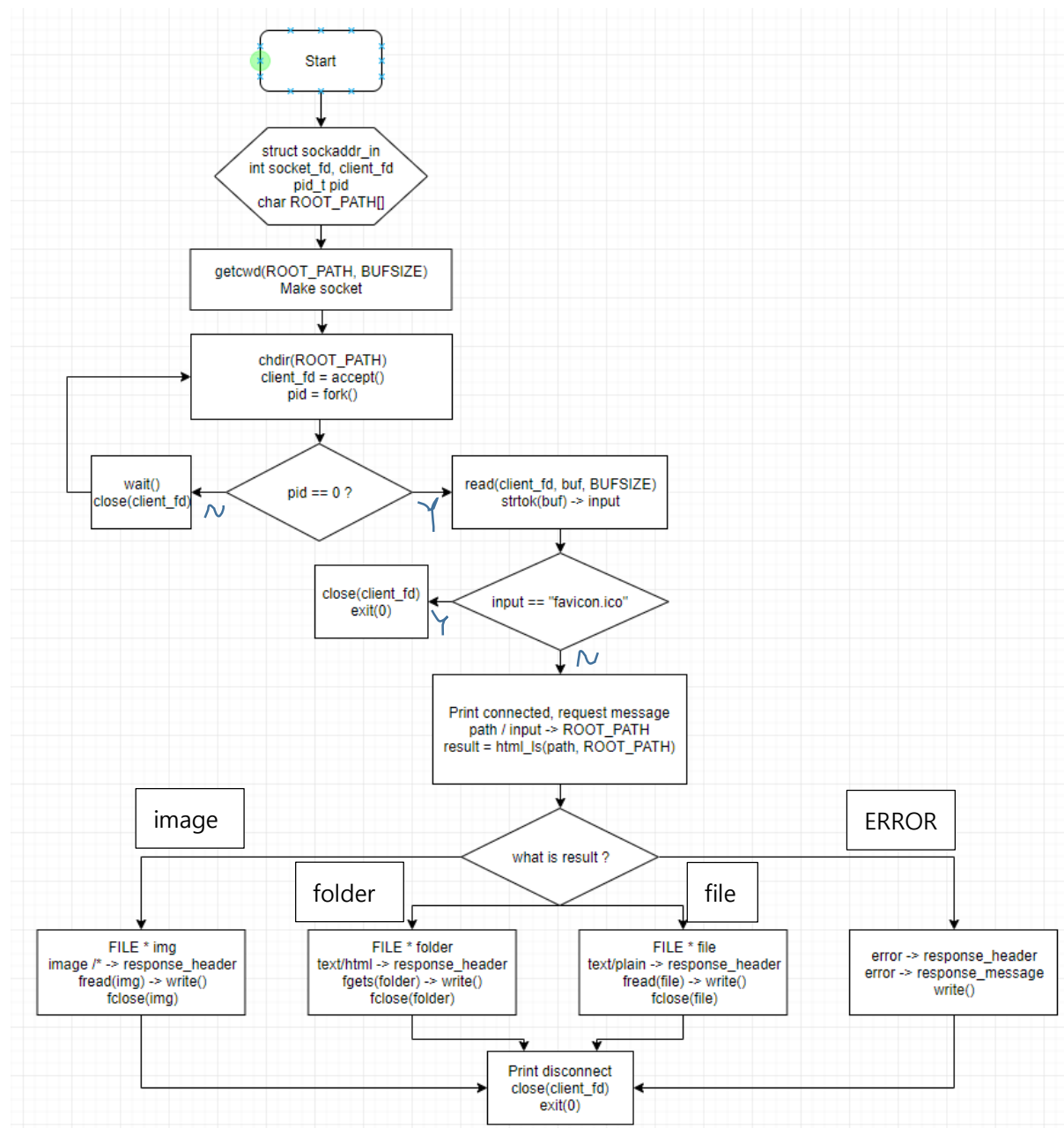
-int advanced_ls



- int html_ls



- int main



Pseudo code

```
int check_real_path(char * filename)
{
    for (int i = 0; i < strlen(filename); i++) // check directory.
    {
        입력된 파일 이름에 '/' 가 있을 경우 해당 index => s_cnt
    }
    if 파일 이름에 '/' 가 있다면 -> real path
    {
        devide < path > / < file >

        <path> 로 작업중인 디렉토리 변경
        s_cnt 값 반환
    }
    else real path 가 아니라면
        0 반환
}
```

```
void swap(char * str1, char * str2)
{
    str1과 str2 swap
}
```

```
int strcmp_i ( char * 비교할 첫 번째 문자열 A, char * 비교할 두 번째 문자열 B )
{
    <예외 처리> : r 옵션이면 거꾸로.
    A 랑 B 가 ".", ".." 이면 "." 이 뒤로,
    둘 중 하나만 "." 이면 "."이 앞으로
    둘 중 하나만 ".." 이면 ".."이 앞으로

    if( A 가 B 보다 길면 )
        len = B 의 길이;
    else
        len = A 의 길이

    for(len 만큼 반복)
    {
        if(소문자라면)
            대문자로 바꾼다
    }
    If 옵션이 -r 이라면
```

```

        if( A > B )      return = -1;
        else if( B > A )  return = 1;
        else             return = 0;

    else

        if( A > B )      return = 1;
        else if( B > A )  return = -1;
        else             return = 0;

}

```

```

void sort_list(char ** li, int len)
{
    for (저장된 파일이름 개수-1 만큼 반복)
    {
        if (더 이상 내용이 없다)
            break;

        for (저장된 파일이름 개수-1 만큼 반복)
        {
            if ( 앞에 저장된 파일이름 > 뒤에 저장된 파일 이름)
                두 파일 이름을 swap
        }
    };
}

```

```

void sort_list_S(char ** li, int size[], int len)
{
    // sort like bubble sort
    for (int i = 0; i < len - 1; i++) // total cycle
    {
        if 리스트가 끝났다면
            break;

        for (int j = 0; j < len - 1; j++) // one cycle
        {
            if 파일 사이즈가 다르다면, 정렬 필요 -r 옵션에 따라서.
            {
                swap file name
            }
        }
    }
}

```

```

        swap file size
    }
    else if 파일 사이즈가 같다면, -> 문자열로 정렬
    {
        if 정렬이 필요하다면
        {
            swap file name
        }
    }
}
return;
}

```

```

void view_advanced_list(char * 파일 경로, char * 파일 이름, struct stat * 파일 정보)
{

```

```

    St_mode에 저장되어 있는 파일 타입에 따라 permission[0] 결정
    St_mode에 저장되어 있는 user 권한에 따라 permission[1~3] 결정
    St_mode에 저장되어 있는 group 권한에 따라 permission[4~6] 결정
    St_mode에 저장되어 있는 other 권한에 따라 permission[7~9] 결정
    ➔ Permission = "- --- --- ---"

```

```

    st_nlink -> f_stat->nlink
    st_uid -> f_stat->uid
    st_gid -> f_stat->gid
    st_size -> f_stat->size
    st_mtime -> f_stat->mtime

```

```

    if 옵션이 -h 라면
    size = size / 1024.0, check_h++(단위가 몇번 상승했는지 체크)
    check_h 값에 따라서 단위(K, M, G) 결정

```

```

    // Full Format 출력
    if 파일 type 이 symbolic link 라면
    {
        if | 옵션이 아니라면
            파일 이름만 HTML 파일로 출력 (green)
        else Full Format 출력
            If 옵션이 -h 라면, 사이즈 출력시 단위와 함께 출력
            Symbolic link 파일과 가리키는 경로 파일이 같은 폴더에 있다면 ->
가리키는 파일명만 HTML 파일로 출력(green)
            Symbolic link 파일과 가리키는 경로 파일이 다른 폴더에 있다면 -> 절대
경로 모두 HTML 파일로 출력(green)
    }
    else symbolic link 가 아니라면
    {
        if | 옵션이 아니라면
            directory일 경우 파일 이름만 HTML 파일로 출력 (blue)
            아닐 경우 파일 이름만 HTML 파일로 출력 (red)
        else | 옵션 이라면
            if directory라면 (blue)
                If 옵션이 -h 라면, FullFormat 출력시 단위도 HTML 파일로 출력

```

```

        else FullFormat HTML 파일로 출력
    else directory가 아니라면 (red)
        If 옵션이 -h 라면, FullFormat 출력시 단위도 HTML 파일로 출력
        else FullFormat HTML 파일로 출력
}

```

```

int advanced_ls(char * 입력한 폴더)
{
    if 폴더가 아니라면
    {
        혹시 절대경로로 입력된 폴더인지 확인        -> 맞으면 해당 디렉토리로 변경
                                                    -> 아니면 현재 디렉토리

        HTML 파일이면 return
        if table이 만들어져 있지 않고, flag_p가 -1이 아니라면,
            옵션 l 에 따라 테이블 생성 -> flag_table=0;

        while 디렉토리 탐색
        {
            if 파일을 찾았다면
                view_advanced_list()
            If fnmatch 로 매칭이 된다면,
                폴더면 다시 advanced_ls()
                파일이면 view_advanced_list()
        }
        If 출력한 파일이 없다면
            에러 메시지 출력 -> 반환 -1
        else 1 반환
    }
    else // is folder
    {
        p 플래그가 표시되어있다면 return 3 => 폴더
        flag_table == 0 이라면 테이블이 열려있으므로 close
        flag_table = 1 set -> 테이블이 안만들어져있다.

        심볼릭 링크로 연결된 폴더인지 확인
        -> 심볼릭 링크 폴더이고 옵션이 l 이면 심볼릭 링크만 Full Format 출력

        아니면 탐색 시작
        while 입력된 폴더 탐색
        {
            HTML 파일은 처리하지 않음.

            if 파일이 없다면
                반복문 탈출

            if 히든 파일이고 옵션이 -l 이나 default 라면
            {
                List에 파일 이름 입력
                Total size += 파일 size
            }
            else if 옵션이 -a 이나 -la 라면 모든 파일 입력
        }
    }
}

```

```

        {
            List에 파일 이름 입력
            Total size += 파일 size
        }
    }

    Directory path 출력
    if 파일이 존재한다면
    {
        List 정렬.
        Total size 출력

        if l 옵션이라면 -> Full Format Table 생성
        else -> 파일 이름 Table 생성
        for 파일 개수 만큼 반복
            view_advanced_list()

        테이블 close
        flag_table=1
    }
}
폴더 닫기
}

```

```

int html(char * 찾을 경로, char * 루트 경로)
{
    origin_wd에 루트 경로 복사
    HTML 파일 기본 포맷 입력, 타이틀에 현재 디렉토리 입력.

    HTML_FILE 에 쓰기위한 fopen

    title에 찾을 경로 입력

    찾을 경로가 루트 경로이면 -l 옵션하고 Welcome, 아니면 -la 옵션하고 Welcome 생략

    result = advanced_ls(path);

    테이블 close
    html 파일 포맷 close
    open한 파일 close

    result 반환.
}

```

```

- int main
{
    소켓 생성 -> 연결 준비

    while(1)
    {

```

ROOT PATH 로 디렉토리 변경
클라이언트로 부터의 연결을 기다림

연결이 되면 fork()

자식 프로세스인 경우

request 메시지를 읽고 분석
URL 칸의 내용을 input 에 저장. -> favicon.ico 무시
연결이 되었고, 요청 메시지 출력.
루트 경로가 아닌 경우 루트 경로 / input -> path
루트 경로인 경우 루트 경로 -> path

result 에 html_ls(path) 의 결과 출력 (폴더:3, 파일:1, 오류:-1)

if 이미지 파일인 경우
 이미지 파일 open.
 헤더 메시지 : image/*
 응답 메시지에 이미지 파일 내용 전송
else if 폴더인 경우
 html_ls 의 결과인 html 파일 open
 헤더 메시지 : text/html
 응답 메시지에 html 파일 내용 전송
else if 파일인 경우
 파일 open
 헤더 메시지 : text/plain
 응답 메시지에 파일 내용 전송
else
 에러 예외처리 메시지 전송

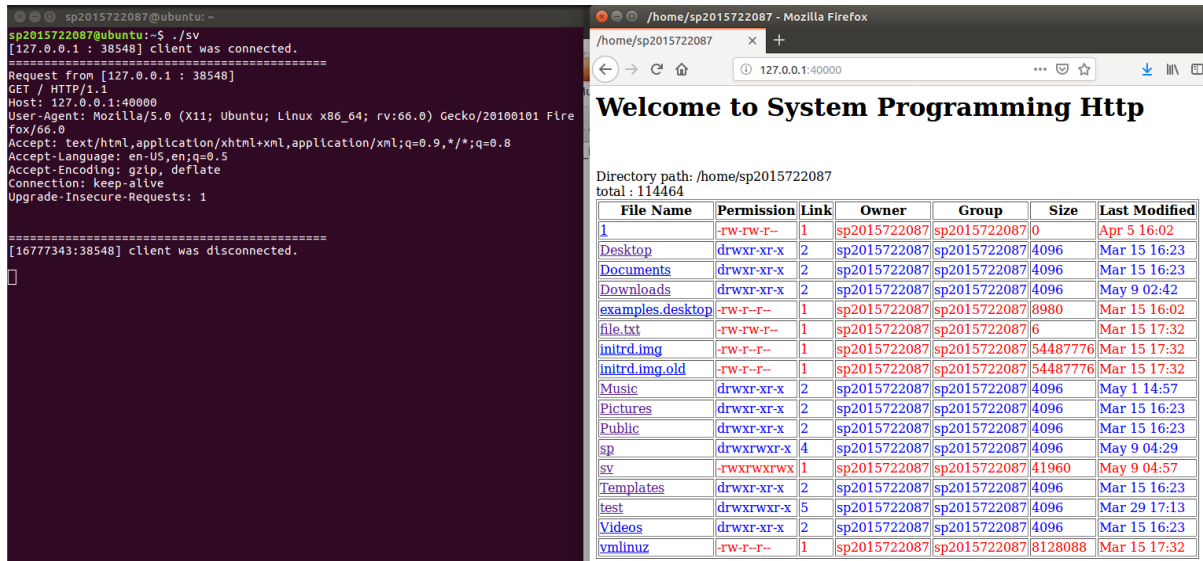
연결 종료.
프로세스 종료

부모 프로세스인 경우

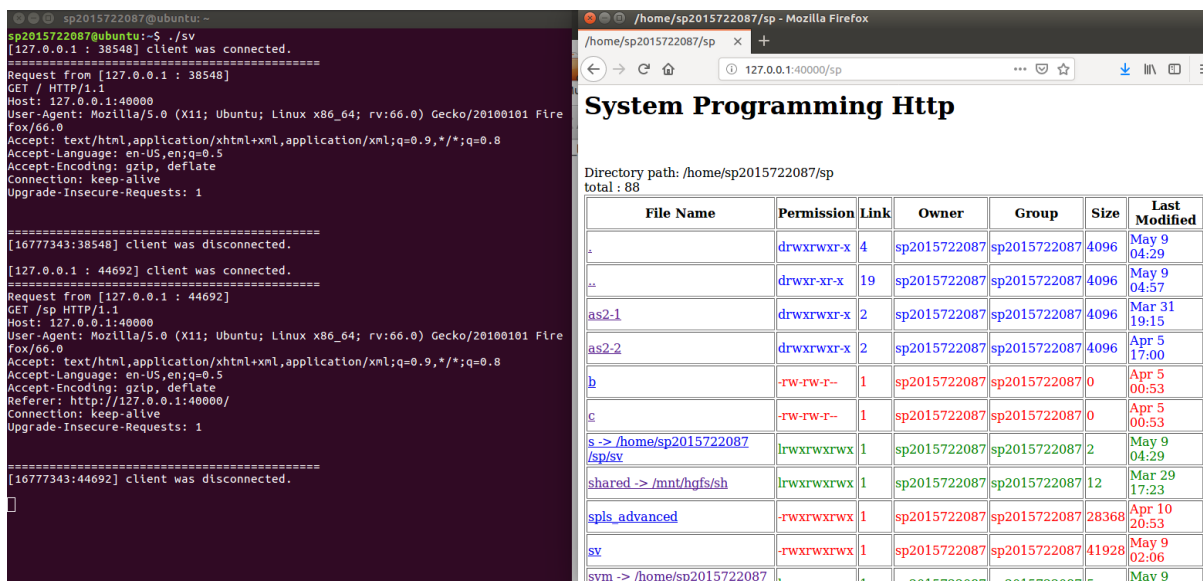
자식 프로세스가 종료되길 기다림
연결 종료

}
소켓 종료
return 0
}

Result



- 홈 디렉토리에서 서버를 실행한 모습이다. 루트 경로는 홈 디렉토리이고 -i 옵션으로 출력한다.



- 첫 화면에서 sp 디렉토리를 클릭한 이후 모습이다. 루트 경로의 하위 디렉토리 이므로 해당 디렉토리의 내용을 -al 옵션으로 출력한다.


```

sp2015722087@ubuntu: ~/sp/shared/server
sp2015722087@ubuntu:~/sp/shared/server$ ./sv
[127.0.0.1 : 27797] client was connected.
=====
Request from [127.0.0.1 : 27797]
GET / HTTP/1.1
Host: 127.0.0.1:40000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Firefox/66.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

=====
[16777343:27797] client was disconnected.

```

Welcome to System Programming Http

Directory path: /mnt/hgfs/sh/server
total : 320

File Name	Permission	Link	Owner	Group	Size	Last Modified
backup	drwxrwxrwx	1	root	root	4096	May 9 02:20
cli.c	-rwxrwxrwx	1	root	root	1230	May 3 15:28
html.js.c	-rwxrwxrwx	1	root	root	36430	May 3 18:10
html.js.h	-rwxrwxrwx	1	root	root	26113	May 9 04:53
Makefile	-rwxrwxrwx	1	root	root	49	May 7 16:00
sch.jpg	-rwxrwxrwx	1	root	root	119958	Mar 12 14:59
sv.c	-rwxrwxrwx	1	root	root	1688	May 3 15:28
sv	-rwxrwxrwx	1	root	root	41960	May 9 04:57
sv.c	-rwxrwxrwx	1	root	root	6982	May 9 04:57
test.PNG	-rwxrwxrwx	1	root	root	87627	May 2 16:55

- server 폴더에서 서버를 실행한 모습이다. 루트 경로는 server 폴더이다. 루트 경로이므로 -i 옵션으로 출력한다.

```

sp2015722087@ubuntu: ~/sp/shared/server
sp2015722087@ubuntu:~/sp/shared/server$ ./sv
[127.0.0.1 : 27797] client was connected.
=====
Request from [127.0.0.1 : 27797]
GET / HTTP/1.1
Host: 127.0.0.1:40000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Firefox/66.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

=====
[16777343:27797] client was disconnected.

[127.0.0.1 : 33429] client was connected.
=====
Request from [127.0.0.1 : 33429]
GET /backup HTTP/1.1
Host: 127.0.0.1:40000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:66.0) Gecko/20100101 Firefox/66.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://127.0.0.1:40000/
Connection: keep-alive
Upgrade-Insecure-Requests: 1

=====
[16777343:33429] client was disconnected.

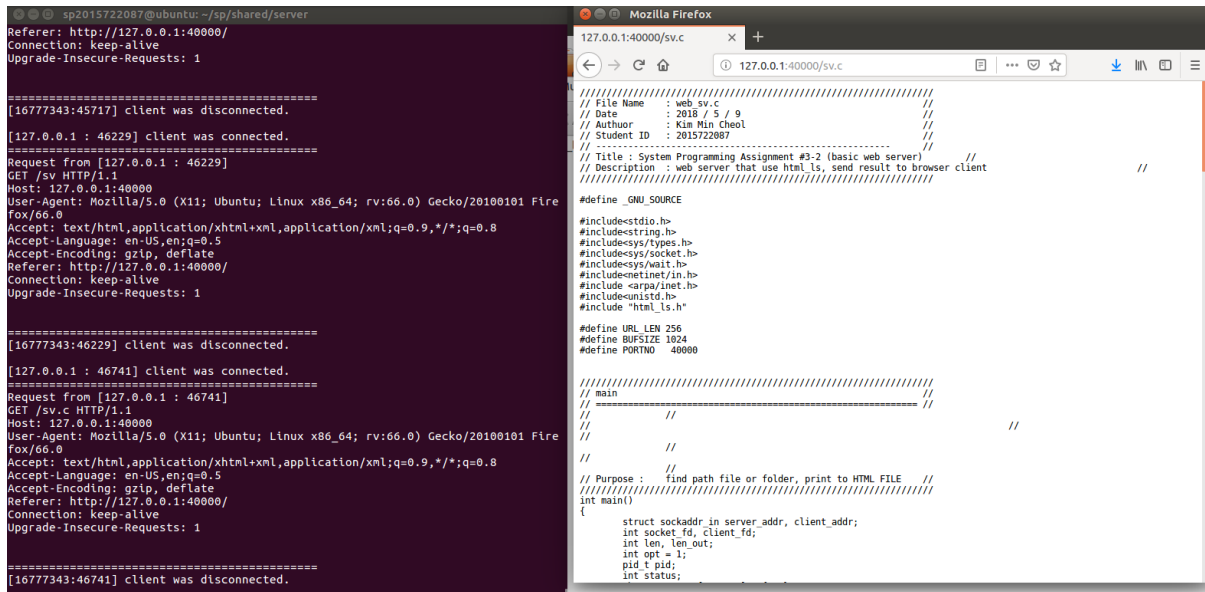
```

System Programming Http

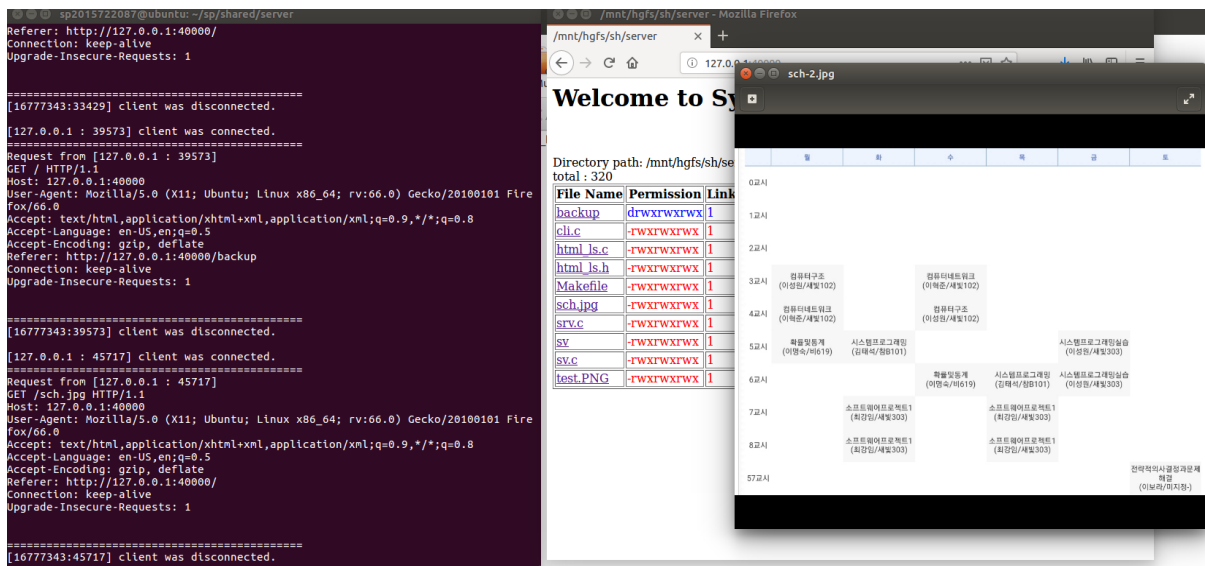
Directory path: /mnt/hgfs/sh/server/backup
total : 79

File Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxrwx	1	root	root	4096	May 9 02:20
..	drwxrwxrwx	1	root	root	4096	May 9 04:57
backup	-rwxrwxrwx	1	root	root	0	May 9 02:01
html.js.h	-rwxrwxrwx	1	root	root	25928	May 9 02:04
Makefile	-rwxrwxrwx	1	root	root	49	May 7 16:00
sv	-rwxrwxrwx	1	root	root	41944	May 9 02:20
sv.c	-rwxrwxrwx	1	root	root	5627	May 9 02:18

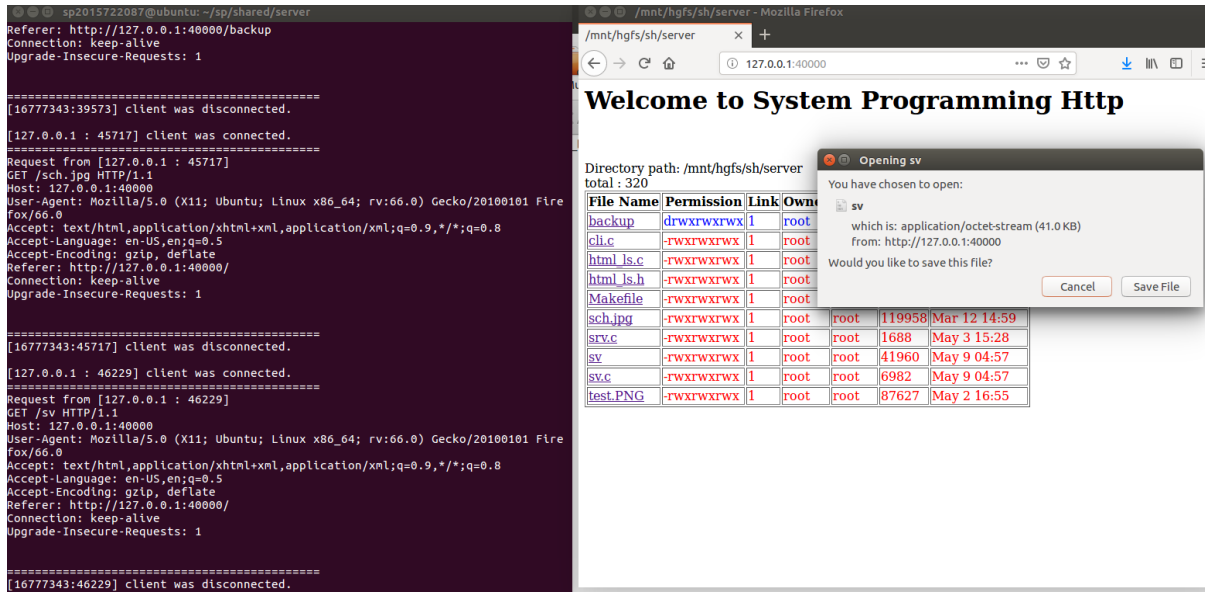
- 위 화면에서 backup 디렉토리를 클릭한 이후 모습이다. 루트 경로의 하위 디렉토리 이므로 -i 옵션으로 출력한다.



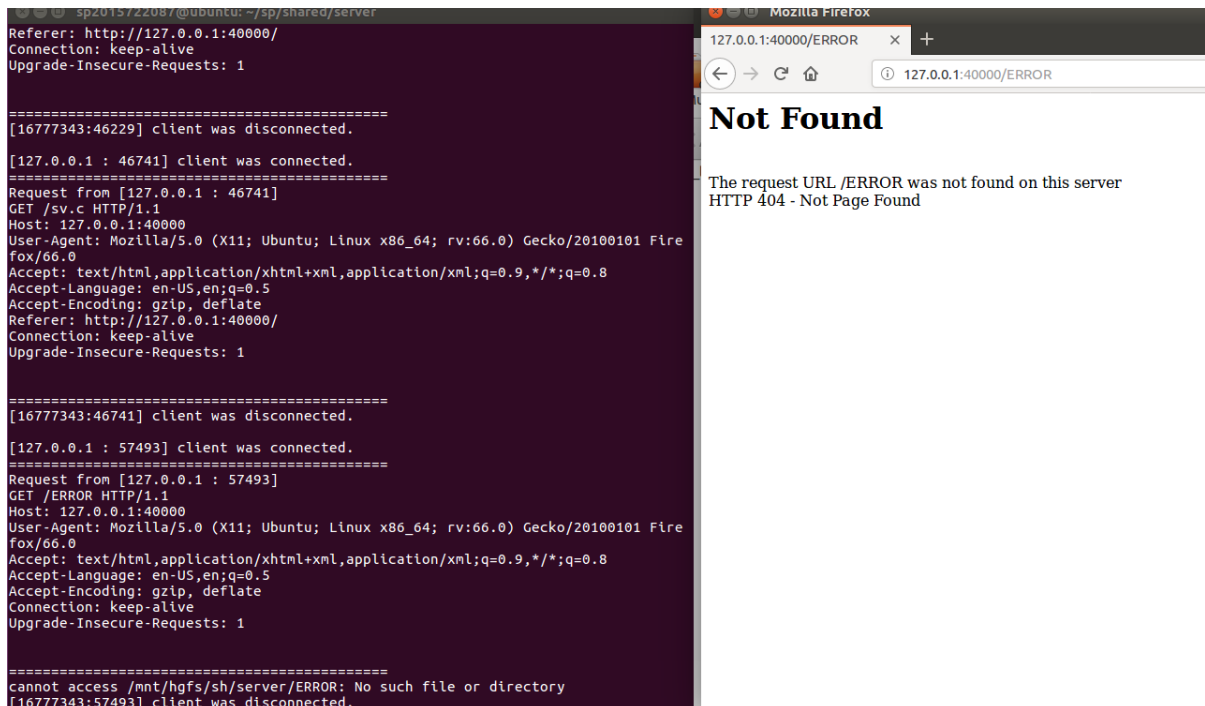
- text 파일을 클릭하여 브라우저에 출력하게한 모습이다.



- 이미지 파일을 클릭하여 바로 이미지 파일을 실행한 모습이다.



- 이진 파일을 클릭한 모습이다. 저장할 것인지 묻는 메시지가 나오고, 저장을 누를 경우 서버로부터 다운로드 받는다.



- 존재하지 않는, 혹은 찾을 수 없는 경로 혹은 파일이 입력된 경우 예외처리를 하였다.

Conclusion

이번 과제는 처음으로 웹 서버와 관련된 프로그래밍을 해보는 과제였다. 실습 시간에 조교님의 설명을 들으면서 강의 자료를 보는데도 이해가 되지 않았고, 집에서 강의 자료를 아무리 봐도 뭘해야 될지, 뭘하는 과제인지 이해가 되질 않았다.

그래서 시스템프로그래밍 과목을 같이 공부를 하면서 인터넷으로 알아보면서 공부하고, 그러면서 예제 코드를 따라서 이것 저것 하다보니 조금씩 이해가 되기 시작했다.

요청 메시지, 응답 메시지, 서버, 클라이언트, 프로토콜까지 이론으로만 외우고 실제로 어떻게 동작하는지는 모르고 있다가, 이렇게 과제를 진행하면서 실제로 메시지를 작성해보고 서버를 구현해보니 조금 감이 잡히게 되어 시스템 프로그래밍 과목 뿐만 아니라 컴퓨터와 관련된 공부를 하는데 있어서 도움이 되는 것 같다.