

시스템프로그래밍 보고서

실험제목: assignment2-2 과제

제출일자: 2023년 04월 30일 (일)

학 과: 컴퓨터공학과

담당교수: 이기훈 교수님

실습분반: 금요일 5 6

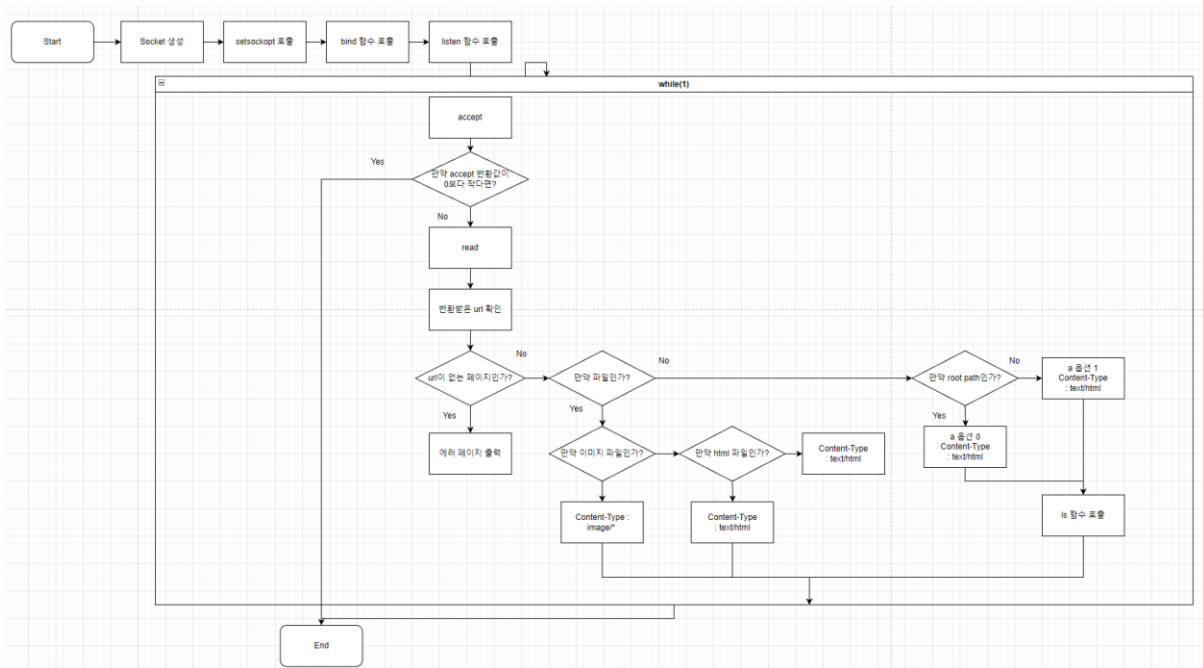
학 번: 2019202031

성 명: 장형범

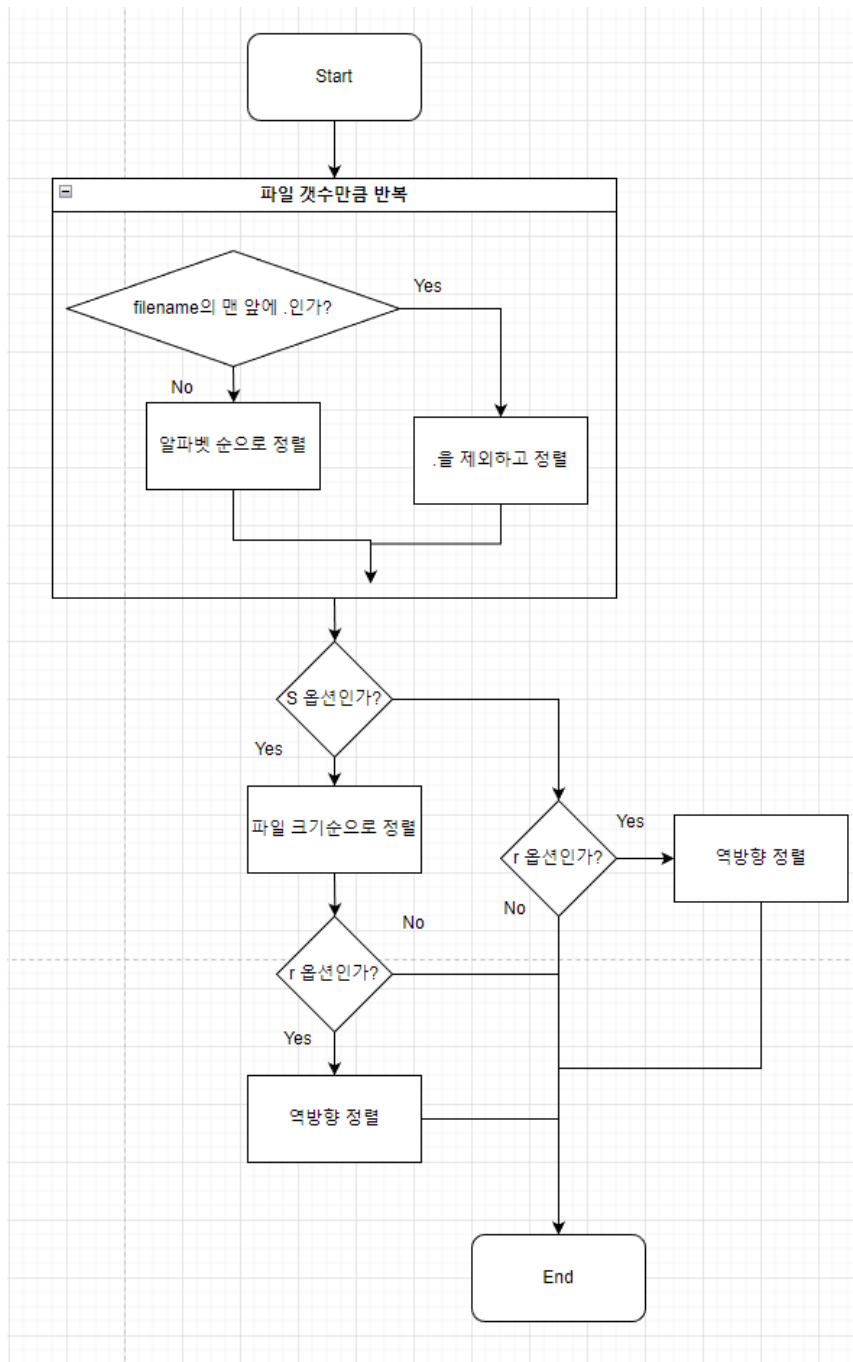
1. Introduction

기존에 만들었던 html 을 만들었던 2-1 코드에서 socket 통신을 하는 web server 를 구현했다. html 을 만들었던 fprintf 를 sprintf 로 바꾸고 write 하는 형식으로 바꿨다. 또한 -r, -h, -S 의 옵션은 없고 -l 옵션은 항상 true -a 옵션만 조건부 true 로 바뀌었다. 이미지 파일은 다운로드 할 수 있도록, text 나 html 파일은 파일이 보이게끔 한다.

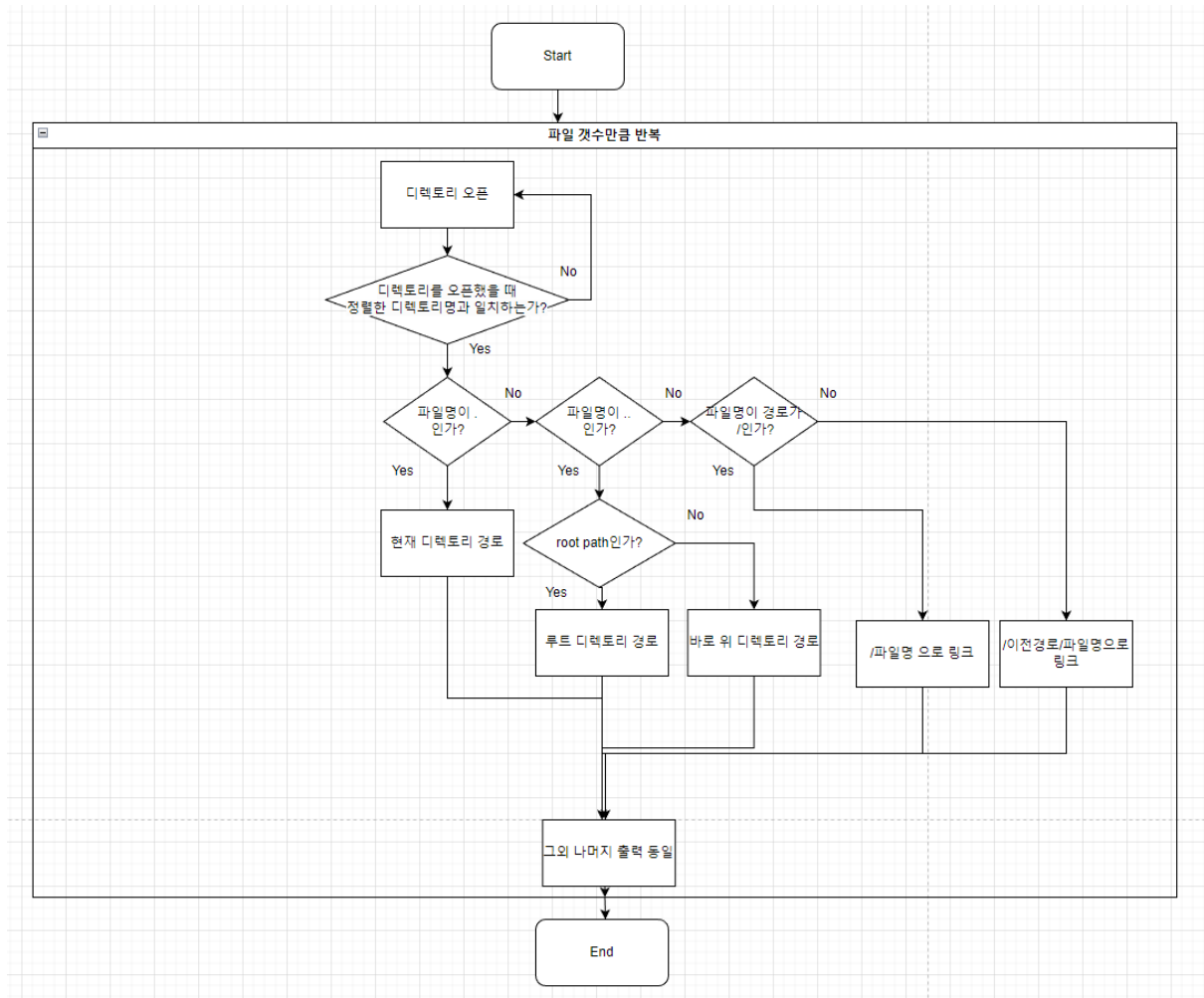
2. Flow chart



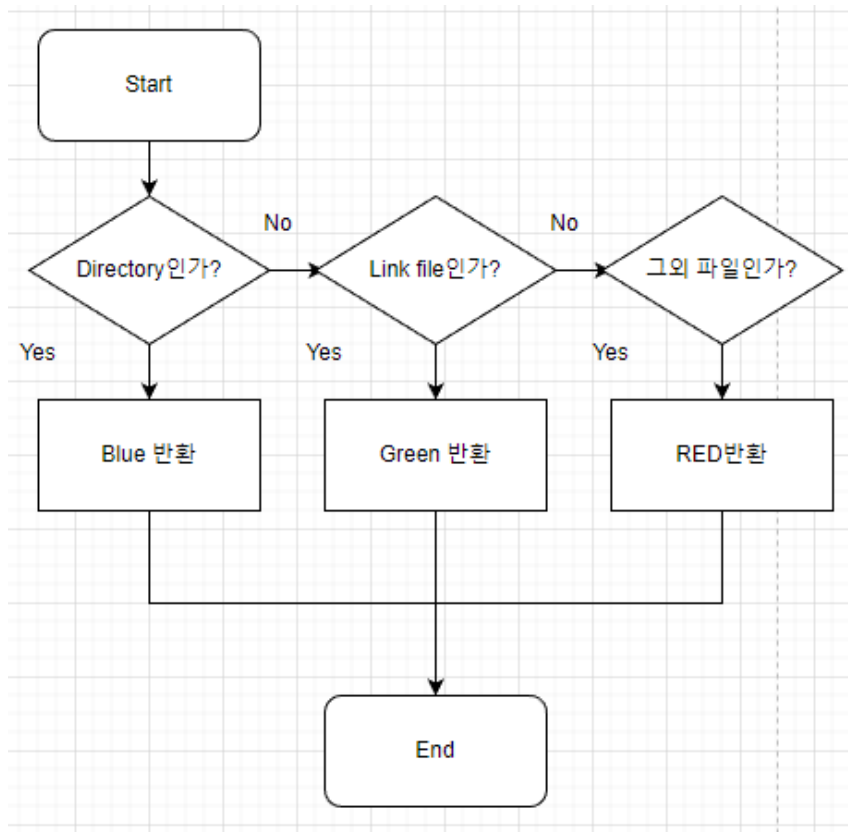
main 함수의 flow chart 이다.



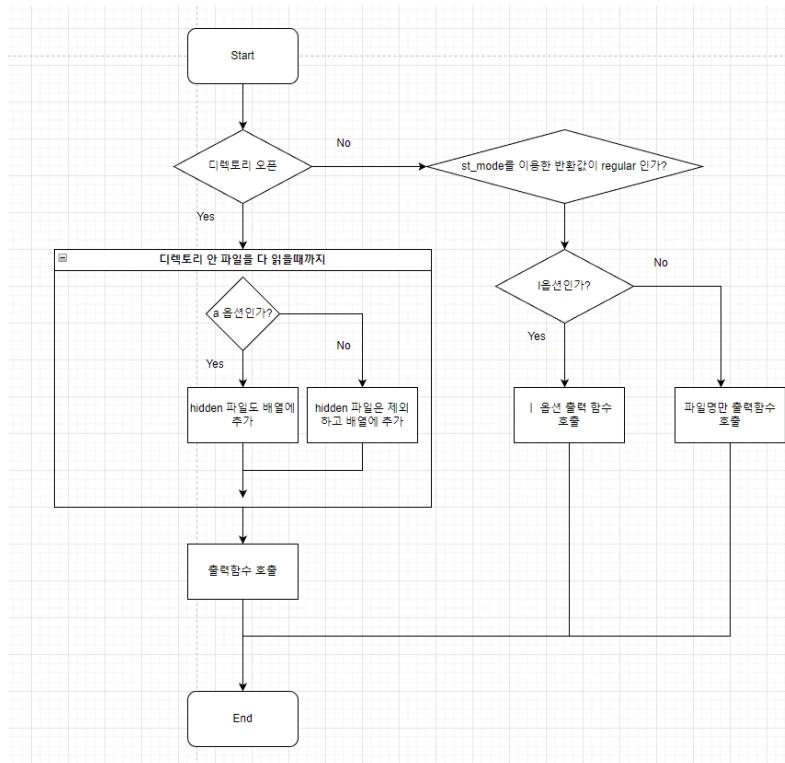
sorting 함수의 순서도이다.



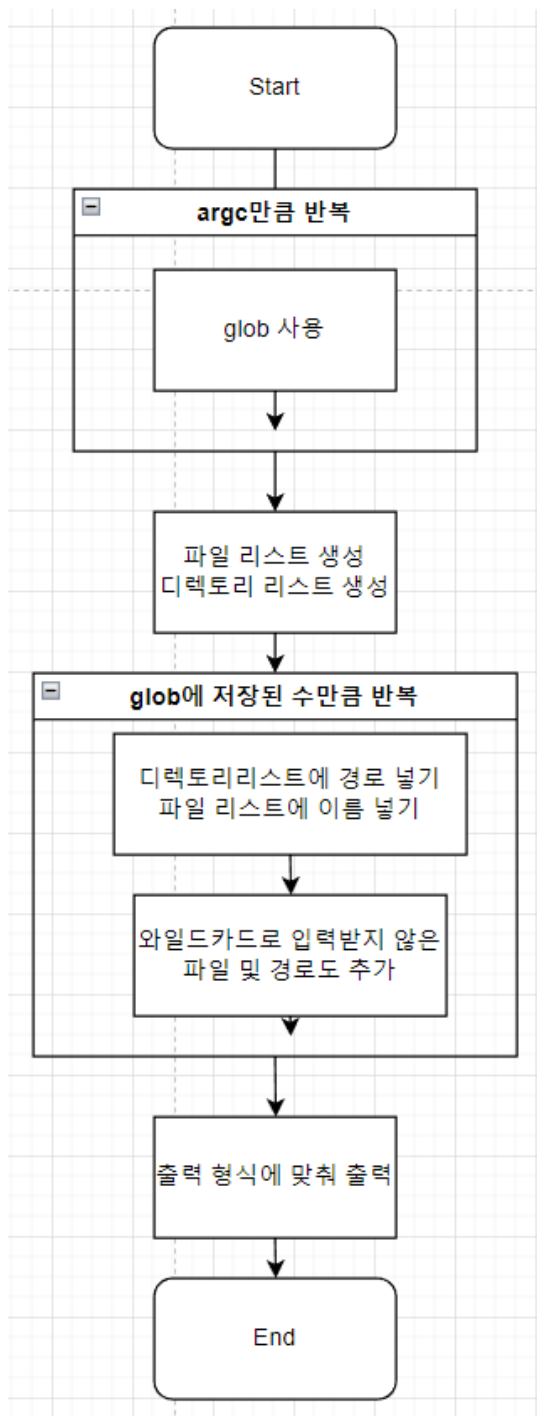
print_file_info 함수의 순서도이다.



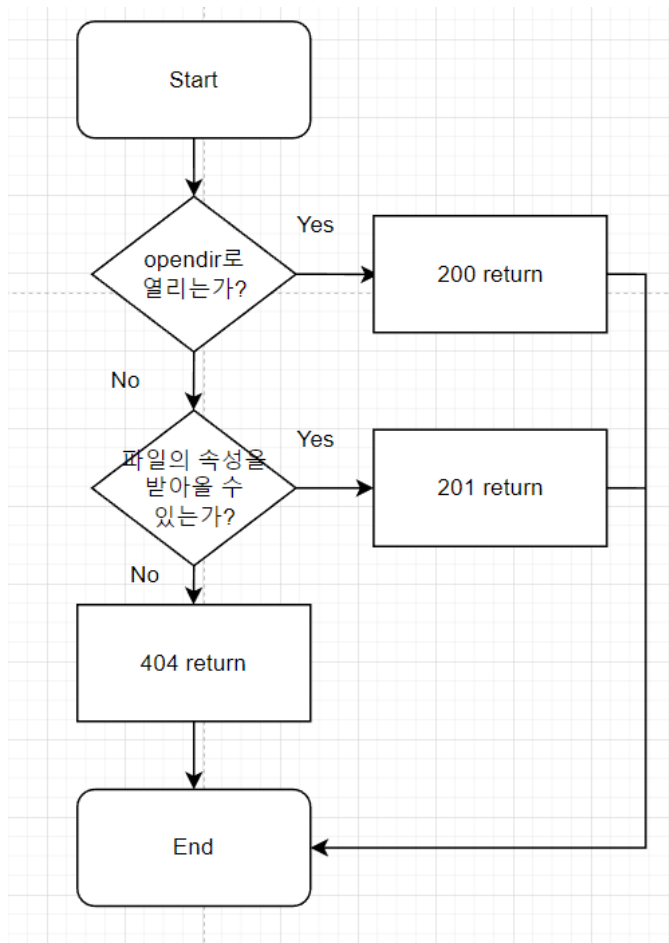
`print_filetype` 함수의 순서도이다.



ls 함수의 순서도이다.



wild card 의 순서도이다.



check_404 의 순서도이다.

3. sudo code

```

int main(int argc, char** argv)
{
    소켓 함수 호출
    setsockopt 로 셋팅
    bind 함수로 소켓과 구조체 정보 binding
    listen 함수로 client로부터 connection을 위해 대기상태로 변경
    while(1)
    {
        accept 로 서버가 클라이언트 접속 허용
        read 로 정보 받음
        들어온 정보에 맞게 결과값 도출
        if(없는 페이지)
            404 에러 출력
        이미지, text, 일반 파일이면 조건에 맞게 출력
    }
}
  
```


root path 라면 a 옵션 off
root path 가 아니면 a 옵션 on

```
    }  
    return 0;  
}  
int check_404  
{  
    디렉토리라면  
        return 200; //exist  
    }  
    파일이라면  
    {  
        return 201; //exist  
    }  
    else  
        return 404; //not exist  
}  
  
void wild_card  
{  
    argc 만큼 반복문을 사용해 glob 함수 사용  
    파일과 디렉토리 갯수 체크  
    파일과 디렉토리 배열 생성  
    배열에 정보 저장  
    와일드카드가 아니었던 경로나 파일을 배열에 추가  
    테이블 생성  
    값 write 하기  
}  
  
void no_dir  
{  
    테이블 생성  
    디렉토리가 아닌 파일의 출력 -1 옵션일때 사용  
    write 하기  
}  
  
void ls  
{  
    디렉토리를 열고 안에 있는 파일 read  
    파일의 size 함수, 배열에 파일명 넣기  
    옵션에 맞는 파일 정렬  
}  
  
void sorting  
{  
    알파벳 순서로 배열 정렬 만약 파일의 이름 맨 앞이 .이라면 그 다음 알파벳  
    부터 비교
```

```
    s 옵션이 1 이라면 dir 를 파일의 크기로 정렬
    r 옵션이 1 이라면 역순으로 파일 정렬

}

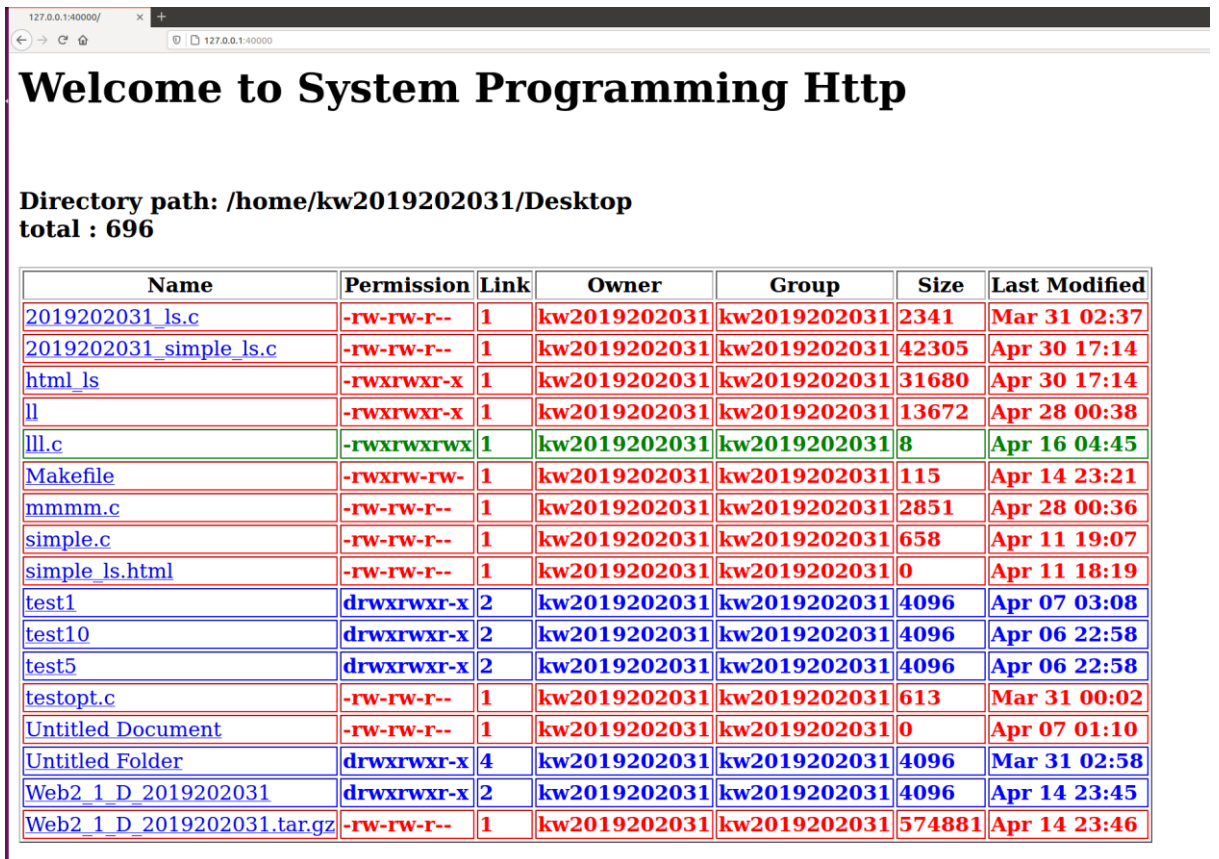
char* check_filetype
{
    type 반환
}

void print_file_info
{
    테이블 생성
    l 옵션이라면 파일의 세부 정보 html 테이블에 입력

    옵션이 없다면 파일의 이름만 테이블에 입력
}

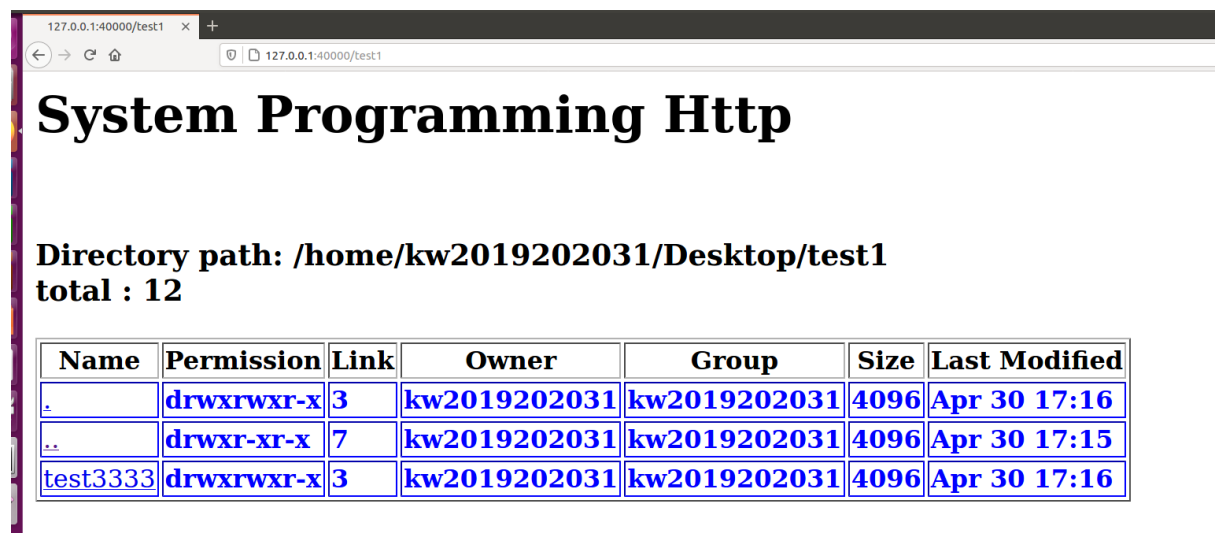
}
```

결과화면



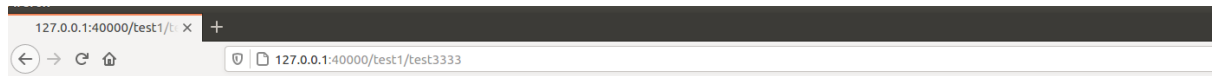
Name	Permission	Link	Owner	Group	Size	Last Modified
2019202031_ls.c	-rw-rw-r--	1	kw2019202031	kw2019202031	2341	Mar 31 02:37
2019202031_simple_ls.c	-rw-rw-r--	1	kw2019202031	kw2019202031	42305	Apr 30 17:14
html_ls	-rwxrwxr-x	1	kw2019202031	kw2019202031	31680	Apr 30 17:14
ll	-rwxrwxr-x	1	kw2019202031	kw2019202031	13672	Apr 28 00:38
lll.c	-rwxrwxrwx	1	kw2019202031	kw2019202031	8	Apr 16 04:45
Makefile	-rwxrw-rw-	1	kw2019202031	kw2019202031	115	Apr 14 23:21
mmmm.c	-rw-rw-r--	1	kw2019202031	kw2019202031	2851	Apr 28 00:36
simple.c	-rw-rw-r--	1	kw2019202031	kw2019202031	658	Apr 11 19:07
simple_ls.html	-rw-rw-r--	1	kw2019202031	kw2019202031	0	Apr 11 18:19
test1	drwxrwxr-x	2	kw2019202031	kw2019202031	4096	Apr 07 03:08
test10	drwxrwxr-x	2	kw2019202031	kw2019202031	4096	Apr 06 22:58
test5	drwxrwxr-x	2	kw2019202031	kw2019202031	4096	Apr 06 22:58
testopt.c	-rw-rw-r--	1	kw2019202031	kw2019202031	613	Mar 31 00:02
Untitled Document	-rw-rw-r--	1	kw2019202031	kw2019202031	0	Apr 07 01:10
Untitled Folder	drwxrwxr-x	4	kw2019202031	kw2019202031	4096	Mar 31 02:58
Web2_1_D_2019202031	drwxrwxr-x	2	kw2019202031	kw2019202031	4096	Apr 14 23:45
Web2_1_D_2019202031.tar.gz	-rw-rw-r--	1	kw2019202031	kw2019202031	574881	Apr 14 23:46

Desktop 에서 화면을 출력했을때의 결과화면이다. Welcome to System Programming Http 가 나오고 안에있는 파일들이 출력되는 것을 확인할 수 있다.



Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	3	kw2019202031	kw2019202031	4096	Apr 30 17:16
..	drwxr-xr-x	7	kw2019202031	kw2019202031	4096	Apr 30 17:15
test3333	drwxrwxr-x	3	kw2019202031	kw2019202031	4096	Apr 30 17:16

하위 디렉토리인 test1 로 들어가면 System Programming Http 로 변환되며 a 옵션인 .파일과 ..파일이 나오게 된다.

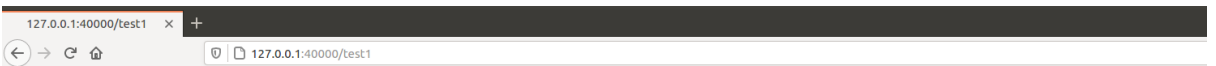


System Programming Http

Directory path: /home/kw2019202031/Desktop/test1/test3333
total : 12

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	3	kw2019202031	kw2019202031	4096	Apr 30 17:16
..	drwxrwxr-x	3	kw2019202031	kw2019202031	4096	Apr 30 17:16
asdf	drwxrwxr-x	2	kw2019202031	kw2019202031	4096	Apr 30 17:16

그 밑의 디렉토리인 test3333 으로 들어갔을 때의 결과이다.



System Programming Http

Directory path: /home/kw2019202031/Desktop/test1
total : 12

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	3	kw2019202031	kw2019202031	4096	Apr 30 17:16
..	drwxr-xr-x	7	kw2019202031	kw2019202031	4096	Apr 30 17:15
test3333	drwxrwxr-x	3	kw2019202031	kw2019202031	4096	Apr 30 17:16

이후 ..을 누르면 다시 그 위의 디렉토리로 올라간다.

```

127.0.0.1:40000/ll.c
127.0.0.1:40000/ll.c

#include<stdio.h>
#include<dirent.h>
#include<unistd.h>

int main()
{
    FILE *file = fopen("simple_ls.html","w");
    if(file == 0)
    {
        printf("error\n");
        return -1;
    }
    <!doctype html>
    <html>
    <head>
    <title>Current Directory</title>
    </head>
    <body>
    <h1>Command : ls -l</h1>
    <table>
    <tr>
    <th>File Name</th>
    <th>File Type</th>
    <th>File Size</th>
    </tr>
    <?php
    $output = shell_exec('ls -l | awk \' {print "<tr><td><a href=\""$9\"">"$9\"</a></td><td>"$1\"</td><td>"$5\"</td></tr>" } \');
    echo $output;
    ?>
    </table>
    </body>
    </html>
    fclose(file);
    return 0;
}

```

.c 파일을 클릭했을 때 내용이 보인다.

kw2019202031 - VMware Workstation

File Edit View VM Tabs Help

Firefox Web Browser

127.0.0.1:40000/

Welcome to System Programming Http

Directory path: /home/kw2019202031/Desktop
total : 712

Opening bonobono.jpg

You have chosen to open:
bonobono.jpg
which is: JPEG image (97.7 KB)
from: http://127.0.0.1:40000

What should Firefox do with this file?

☒ Open with Image Viewer (default)

☐ Save File

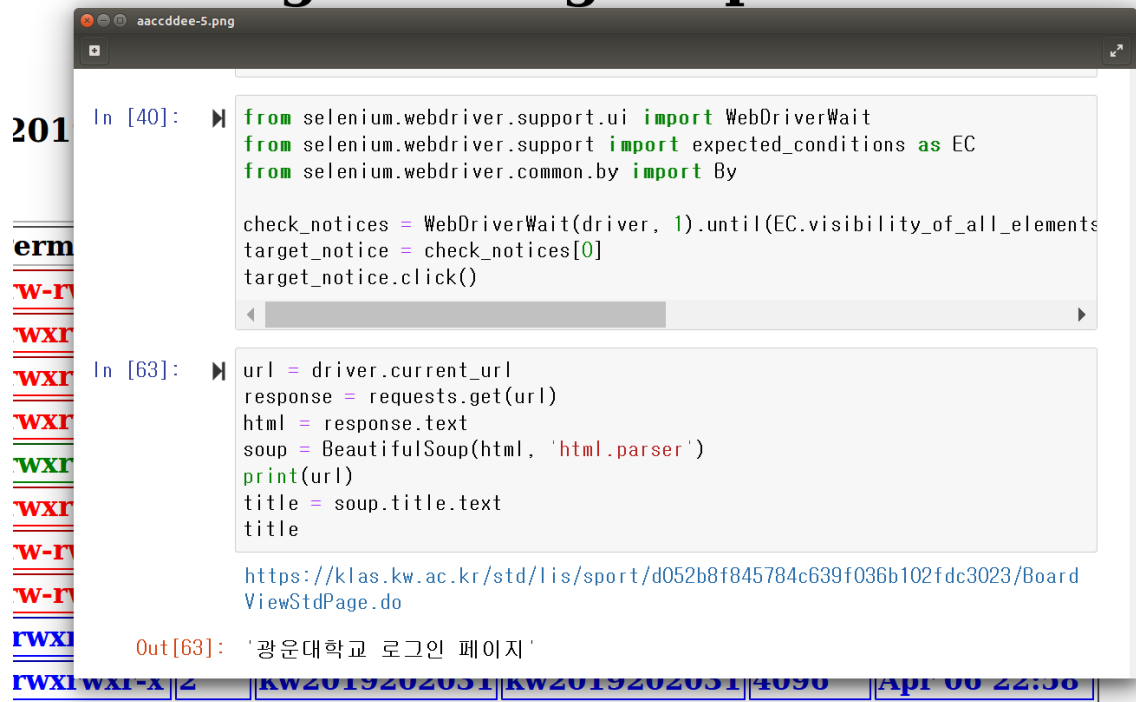
☐ Do this automatically for files like this from now on.

Cancel OK

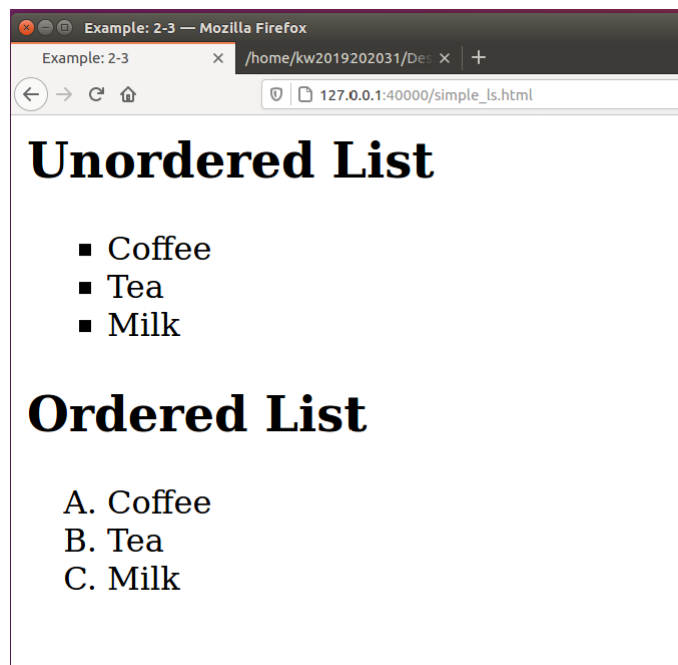
Name	Owner	Group	Size	Last Modified		
2019202031_ls.c	2019202031	kw2019202031	2341	Mar 31 02:37		
2019202031_simple_ls	2019202031	kw2019202031	42305	Apr 30 17:14		
bonobono.jpg	2019202031	kw2019202031	12360	Apr 30 17:20		
html_ls	2019202031	kw2019202031	31680	Apr 30 17:14		
ll	2019202031	kw2019202031	13672	Apr 28 00:38		
ll.c	-rwxrwxrwx	1	kw2019202031	kw2019202031	8	Apr 16 04:45
Makefile	-rwxrw-rw-	1	kw2019202031	kw2019202031	115	Apr 14 23:21
mmmm.c	-rw-rw-r--	1	kw2019202031	kw2019202031	2851	Apr 28 00:36
simple.c	-rw-rw-r--	1	kw2019202031	kw2019202031	658	Apr 11 19:07
simple_ls.html	-rw-rw-r--	1	kw2019202031	kw2019202031	0	Apr 11 18:19
test1	drwxrwxr-x	3	kw2019202031	kw2019202031	4096	Apr 30 17:16
test10	drwxrwxr-x	2	kw2019202031	kw2019202031	4096	Apr 06 22:58
test5	drwxrwxr-x	2	kw2019202031	kw2019202031	4096	Apr 06 22:58
testopt.c	-rw-rw-r--	1	kw2019202031	kw2019202031	613	Mar 31 00:02
Untitled Document	-rw-rw-r--	1	kw2019202031	kw2019202031	0	Apr 07 01:10
Untitled Folder	drwxrwxr-x	4	kw2019202031	kw2019202031	4096	Mar 31 02:58
Web2_1_D_2019202031	drwxrwxr-x	2	kw2019202031	kw2019202031	4096	Apr 14 23:45
Web2_1_D_2019202031.tar.gz	-rw-rw-r--	1	kw2019202031	kw2019202031	574881	Apr 14 23:46

jpg 파일을 클릭했을 때 다운로드가 뜬다.

stem Programming Help



다운받은 이미지가 보인다.



html 파일을 클릭했을 때는 html 이 열려서 보인다.

```
127.0.0.1:40000/llc x /home/kw2019202031/D... x +
127.0.0.1:40000/llc
< > < > 127.0.0.1:40000/llc 200% ... ☆

#include<stdio.h>
#include<dirent.h>
#include<unistd.h>

int main()
{
    FILE *file = fopen("simple_ls.html", "w");
    if(file == 0)
    {
        printf("error\n");
        return -1;
    }
    <!doctype html>
    <html>
    <head>
    <title>Current Directory</title>
    </head>
    <body>
    <h1>Command : ls -l</h1>
    <table>
    <tr>
    <th>File Name</th>
    <th>File Type</th>
    <th>File Size</th>
    </tr>
    <?php
    $output = shell_exec('ls -l | awk \'{print "<tr><td><a href=\""$9\"">"$9"</a></td><td>"$1"</td><td>"$5"</td></tr>"\'}');
    echo $output;
    ?>
    </table>
    </body>
    </html>
    fclose(file);
    return 0;
}
```

링크 파일을 클릭했을 때 해당 파일의 내용을 보여준다. 위의 사진은 .c 파일이다.

고찰

과제를 하면서도 이게 돌아가네 싶은게 여러 개 있었다. 이게 이렇게 하면 되는구나인지 이게 돌아가는지 헷갈린다. 과제를 하면서도 서버 통신의 개념이 완벽하게 이해가 된건지 잘 모르겠다.

txt 와 source 코드를 화면에 보이게 하는 방법이 내용을 가져온 뒤 write 하는건가? 라는 생각을 했는데 정말 그렇게 해야하는 거였어서 아 서버 통신이 이렇게 돌아가는구나 라고 생각했다. header 를 결정할 때 띄어쓰기와 \r\n 을 적지 않아 오류가 발생하기도 했다.

reference

강의자료 참고