

시스템프로그래밍실습 보고서

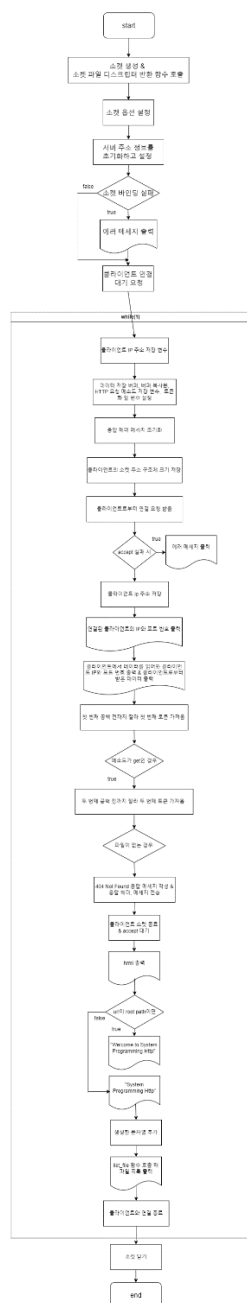
Assignment 2-2

과목	시스템프로그래밍실습
담당교수	이기훈교수님
학과	컴퓨터정보공학부
학번	2021202058
이름	송채영

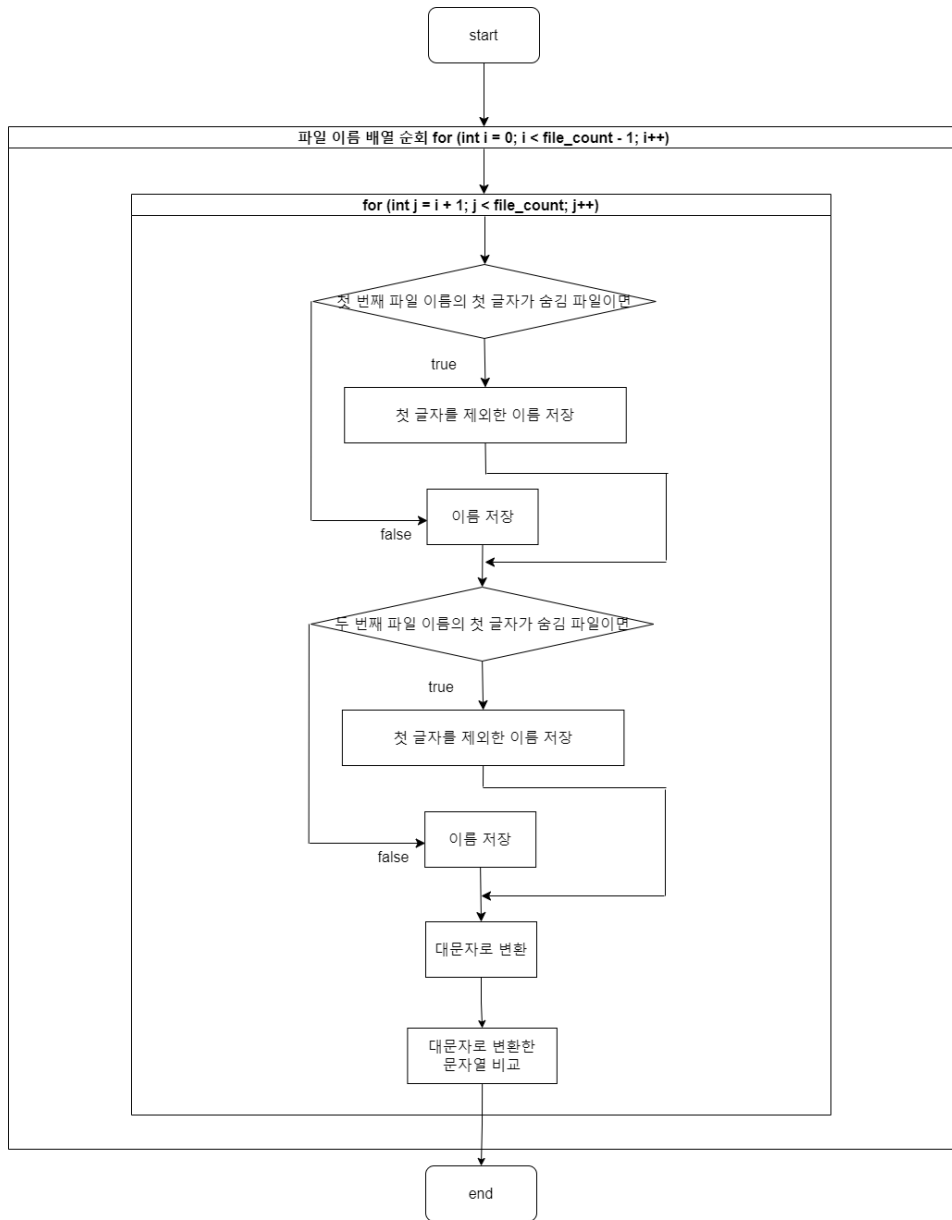
1. Introduction

이번 과제에서는 HTML_1s 의 결과를 바탕으로 HTML 파일을 다른 장치의 웹 브라우저에서 확인할 수 있도록 지원하는 서버 프로그램을 구현한다. 웹 서버를 구현해 클라이언트가 요청을 보내면 요청을 분석하고 HTTP 응답 메시지로 클라이언트에 보내 결과를 얻을 수 있다. 이번 과제를 통해 socket 의 개념을 이해하고, socket(), bind(), struct sockaddr_in, listen(), accept(), connect(), write(), read()등의 함수에 대해 이해하고 적용해본다.

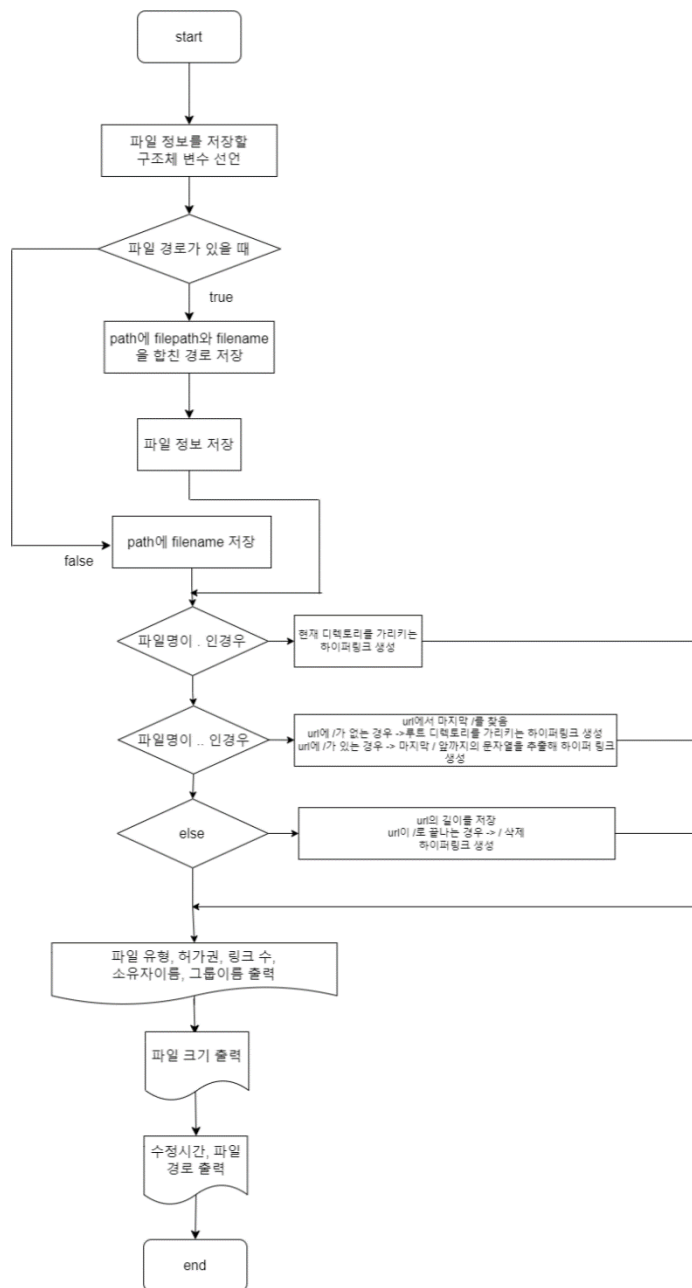
2. Flow chart



코드의 전체적인 흐름을 flow chart 로 나타내었다. 우선 main 함수에 대해 자세히 설명해보면 다음과 같다. 프로그램이 시작하면, 소켓을 생성하고 소켓 파일 디스크립터 반환 함수를 호출한 후 소켓 옵션을 설정한다. memset 을 사용하여 서버 주소를 초기화하고 설정해준다. 소켓 바인딩을 실패하면 에러 메시지를 출력해주고, listen() 함수를 사용해 클라이언트의 연결 요청을 대기한다. while 이 1 일 때 클라이언트 IP 주소 저장 변수와 클라이언트로부터 받은 데이터를 저장할 버퍼, 버퍼 복사본, HTTP 요청 메소드 저장 변수, HTTP 요청 메시지 토큰화 할 포인터를 선언한 후 응답 헤더와 메시지를 초기화한다. 클라이언트의 소켓 주소 구조체 크기를 저장한 후 클라이언트로부터 연결 요청을 받으며, accept 를 실패하면 에러 메시지를 출력해준다. 클라이언트 IP 주소를 저장한 후 연결된 클라이언트의 IP 와 포트 번호를 출력한다. 클라이언트로 데이터를 읽어온 후 클라이언트 IP 와 포트 번호를 출력하고 클라이언트로부터 받은 데이터도 출력한다. 첫 번째 공백 전까지 문자열을 잘라서 첫 번째 토큰을 가져오는데 이때 토큰이 없는 경우 무시한다. url 배열을 초기화한 후 메서드가 GET 인 경우 두 번째 공백 전까지 문자열을 잘라서 두 번째 토큰을 가져온다. 현재 작업 디렉토리를 cwd 에 저장하고 요청한 파일의 경로를 만들어, 경로가 없는 경우 404 Not Found 응답 메시지를 작성한다. 응답 헤더와 메시지를 전송한 후 클라이언트 소켓을 종료한다. 다시 accept 대기로 돌아간 후 html 출력을 위해 필요한 부분을 구현한다. url 이 root path 인 경우 lflag 를 1 로 설정해주고 "Welcome to System Programming Http"을 출력한다. 하위디렉토리일 경우 aflag 와 lflag 를 1 로 설정해주고 "System Programming Http"를 출력한다. cwd 의 파일 목록을 출력한 후 클라이언트와 연결을 종료한다. 소켓을 닫고 프로그램을 종료한다.

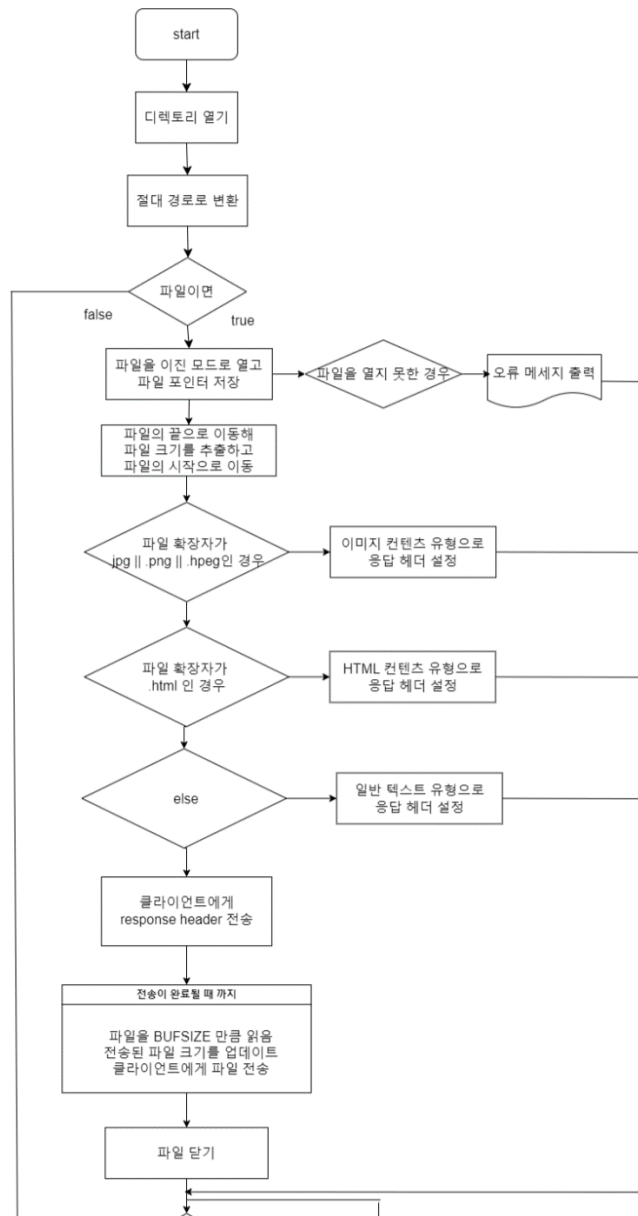


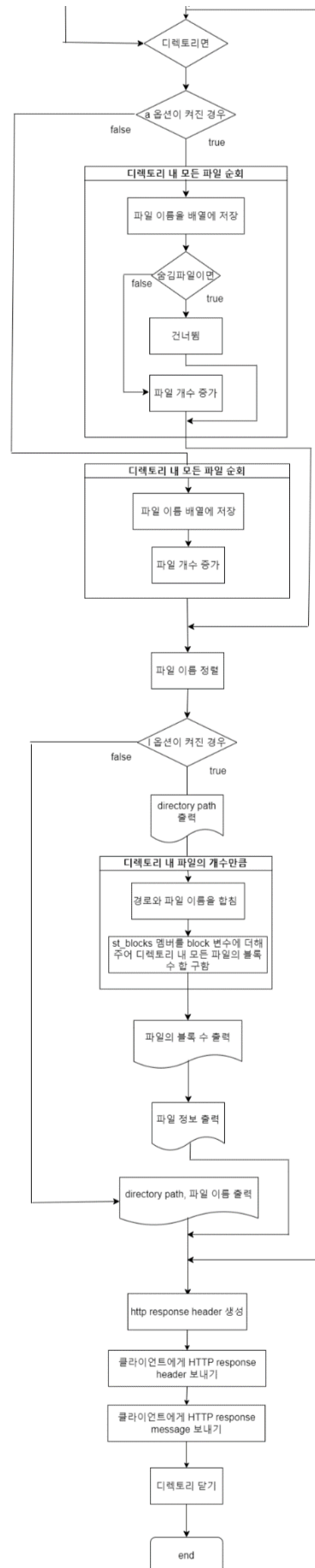
다음은 sort 함수의 flow chart 이다. 우선 파일의 개수만큼 반복한다. 첫 번째 파일이 숨김 파일일 경우, 파일의 이름을 길이만큼 반복하고 첫 글자를 제외한 이름을 저장한다. 숨김파일이 아닐 경우 이름을 저장해준다. 두 번째 파일도 동일하게 진행한다. 먼저 대문자로 변환해준 후 대문자로 변환한 문자열을 비교한다.



다음은 print_file_info, 파일 정보를 출력하는 함수의 flow chart 이다. 파일 정보를 읽어 구조체 변수인 filestat 에 저장한다. 파일 경로가 있고, filepath 와 filename 이 같지 않으면 파일 경로가 존재하는 경우이므로 경로와 파일 이름을 합치며, 파일 경로가 없는 경우 파일이름을 정보로 읽는다. 파일명이 ., .., else 인 경우로 나누어 하이퍼링크를 준다. 먼저 .인 경우 현재 디렉토리를 가리키는 하이퍼링크를 생성하며, ..일 경우 url 에서 마지막 /을 찾고 /이 있으면 루트 디렉토리를 가리키는 하이퍼링크를 생성하고, 없을 경우 마지막 / 앞까지의 문자열을 추출한 후 하이퍼링크를 생성해준다. else 인 경우 url 의 길이를 저장한 후 url 이 /로 끝나면 /을 삭제한 후 하이퍼링크를 생성하도록

하였다. 파일 유형, 허가권, 링크 수, 소유자,호출해주며, 파일 크기, 수정 시간, 파일 경로를 출력한 후 프로그램을 종료한다.





다음은 list files 함수의 flow chart 이다. 파일을 열어준 후 절대경로로 변환한다. 연
 파일이 파일일때와 디렉토리 일 때로 나눈다. 파일일 때, 파일을 이진 모드로 열고 파일
 포인터를 file 변수에 저장한다. 파일을 열지 못했을 경우 오류 메시지를 출력한다.
 파일의 끝으로 이동해 파일 크기를 추출한 후 파일의 시작으로 이동한다. 파일 확장자에
 따라 content-type 을 설정하였는데, jpg, .png, jpeg 일 때 이미지 컨텐츠 유형으로 응답
 헤더를 설정한다. 파일 확장명이 .html 인 경우 HTML 컨텐츠 유형으로 응답 헤더를
 설정한다. else 일 경우 일반 텍스트 유형으로 응답 헤더를 설정하고 클라이언트에게
 response header 를 전송하였다. 전송이 완료될 때 까지 반복하며 파일을 BUFSIZE 만큼
 읽어서 전송된 파일 크기를 업데이트 하고 클라이언트에게 파일을 전송한다. 파일을
 닫는다. 디렉토리일 경우 a 옵션이 꺼져 있을 때와 켜져 있을 때, l 옵션이 켜져 있을
 때로 나눈다. a 옵션이 꺼져 있으면 디렉토리 내 모든 파일을 순회하며 파일이름을
 배열에 저장한다. .과 ..은 건너뛰고 그 외의 파일은 파일 개수를 증가시킨다. a 옵션이
 켜져 있을 경우, 디렉토리 내 모든 파일을 파일 이름을 배열에 저장하고 파일 개수를
 증가시킨다. 파일 이름 정렬 함수를 호출해 정렬한다. l 옵션이 켜져 있으면, 디렉토리
 경로와 total 을 출력해주고 파일 정보 출력 함수를 호출해 출력한다. HTTP response
 header 를 생성해 작성해 준 후, 클라이언트에게 HTTP response header 와 message 를
 보낸 후 디렉토리를 닫고 프로그램을 종료한다.

위 함수에서 html 에 directory path 와 total 그리고 filename 을 table 형태로 출력해주기
 위해 html 코드로 출력하는 부분을 추가했다. 또한 html_ls.html 파일일 경우 건너뛰는
 코드를 추가해 html_ls.html 파일이 출력되지 않도록 하였다. flow chart 의 흐름은 기존과
 변함이 없다.

3. Pseudo Code

```
선택정렬() //sort
{
    for(파일 수만큼 파일 이름 배열 순회)
    {
        if(첫번째 파일 첫 글자가 숨김 파일이면)
        {
            filename[i]의 길이만큼 반복하고 첫 글자 제외하고 저장
        }
        else(숨김파일이 아니면)
        {
            이름 저장
        }
        if(두번째 파일 첫 글자가 숨김 파일이면)
        {
            filename[i]의 길이만큼 반복하고 첫 글자 제외하고 저장
        }
        else(숨김파일이 아니면)
        {
            이름 저장
        }
        hidden_file 1 대문자로 변환
        hidden_file 2 대문자로 변환
        대문자로 변환한 문자열 비교
    }
}
```


sort 함수의 pseudo code 이다.

```
파일 정보 출력() //print_file_info
{
    파일 정보 읽어 구조체 변수 filestat에 저장
    파일 권한 정보 저장할 변수 선언
    if(파일 경로가 있으면)
    {
        파일 경로와 파일 이름을 합치고 stat함수로 파일 정보 읽음
    }
    else
    {
        파일 이름으로 정보 읽음
    }
    if(파일명이 .인 경우) 현재 디렉토리를 가리키는 하이퍼링크 생성
    else if(파일명이 ..인 경우)
    {
        url에서 마지막 /를 찾음
        url에 /가 없는 경우 -> 루트 디렉토리를 가리키는 하이퍼링크 생성
        url에 /가 있는 경우 -> 마지막 / 앞까지의 문자열을 추출 후 하이퍼링크 생성
    }
    else
    {
        url의 길이 저장
        url이 /로 끝나는 경우 -> /를 삭제
        하이퍼링크 생성
    }
    // 파일 유형, 허가권, 링크 수 , 소유자, 그룹, 출력
    if(파일 유형이 디렉토리면) permission 배열의 0번째는 d
    else if(파일 유형이 링크면) permission 배열의 0번째는 l
    else permission 배열의 0번째는 -
    파일 허가권 정보 설정

    if(디렉토리면) 파란색으로 출력
    else if(link 파일이면) 초록색으로 출력
    else 빨간색으로 출력

    파일 크기 출력
    수정시간, 파일 경로 출력
}
```

print_file_info 함수의 pseudo code 이다.

```

파일 목록 출력() //list_files
{
    디렉토리 열기
    절대경로로 변환
    if(파일이면)
    {
        파일을 이진 모드로 열고 파일 포인터 저장
        if(파일 열지 못하면) 에러메세지 출력
        파일 끝으로 이동해 파일 크기 추출한 후 파일 시작으로 이동

        파일 확장자에 따라 content-type 설정
        if (jpg, png, jpeg)
        {
            이미지 콘텐츠 유형으로 응답 헤더 설정
        }
        else if(.html)
        {
            HTML 콘텐츠 유형으로 응답 헤더 설정
        }
        else
        {
            일반 텍스트 유형으로 응답 헤더 설정
        }
        클라이언트에게 response header를 전송
        while(전송이 완료될 때까지)
        {
            파일을 BUFSIZE 만큼 읽음
            전송된 파일 크기를 업데이트
            클라이언트에게 파일을 전송
        }
        파일 닫기
    }
    else //디렉토리면
    {
        if(a 옵션이 꺼져있는 경우)
        {
            while(디렉토리 내 모든 파일 순회)
            {
                html_ls.html 파일은 건너뛸
                숨김파일이면 넘어감
                아니면 파일 개수 증가
            }
        }
        else
        {
            while(디렉토리 내 모든 파일 순회)
            {
                html_ls.html 파일은 건너뛸
                파일 이름 배열에 저장
            }
        }
        파일 이름 정렬
        if(l 옵션이 켜져있는 경우)
        {
            directory path, total 출력
            HTML 파일에 테이블 태그 출력
            HTML 파일에 테이블 헤더 삽입
            파일 정보 출력
        }
        HTTP response header 생성
        클라이언트에게 response header, message 보내기
    }
    디렉토리 닫기
}

```

list_files 함수의 pseudo code 이다.

```

main(argc, argv)
{
    소켓을 생성하고 소켓 파일 디스크립터 반환 함수를 호출함
    소켓 옵션을 설정함
    memset을 사용해 서버 주소 정보를 초기화하고 설정함
    if(소켓 바인딩 실패) 오류메세지 출력
    클라이언트 연결 요청 대기

    while(1)
    {
        응답 헤더, 메세지 초기화
        클라이언트로부터 연결 요청 받음
        if(accept 실패 시) 에러 메세지 출력
        클라이언트 IP 주소 저장 후 연결된 클라이언트의 IP와 포트 번호 출력
        클라이언트로부터 데이터를 읽어와 IP와 포트 번호 출력
        put () 사용해 클라이언트로부터 받은 데이터 출력

        첫번째 공백 전까지 문자열 잘라서 첫 번째 토큰 가져옴
        if(토큰이 없는 경우) continue
        url 배열 초기화
        if(method가 GET인 경우)
        {
            두번째 공백 전까지 문자열 잘라서 두 번째 토큰 가져옴
        }
        현재 작업 디렉토리 cwd변수에 저장
        if(파일이 없는 경우)
        {
            404 Not Found 응답 메세지 작성
            응답 헤더, 메세지 전송
            클라이언트 소켓 종료
            accpet대기로 돌아감
        }
        if(url이 root path)
        {
            lflag = 1
            "Welcome to System Programming Http" 태그
        }
        else //하위 디렉토리
        {
            aflag =1, lflag = 1
            "System Programming Http"
        }
        list_files 함수 호출해 cwd의 파일 목록 출력
        클라이언트와 연결 종료
    }
    소켓 닫기
}

```

main 함수의 pseudo code 이다.

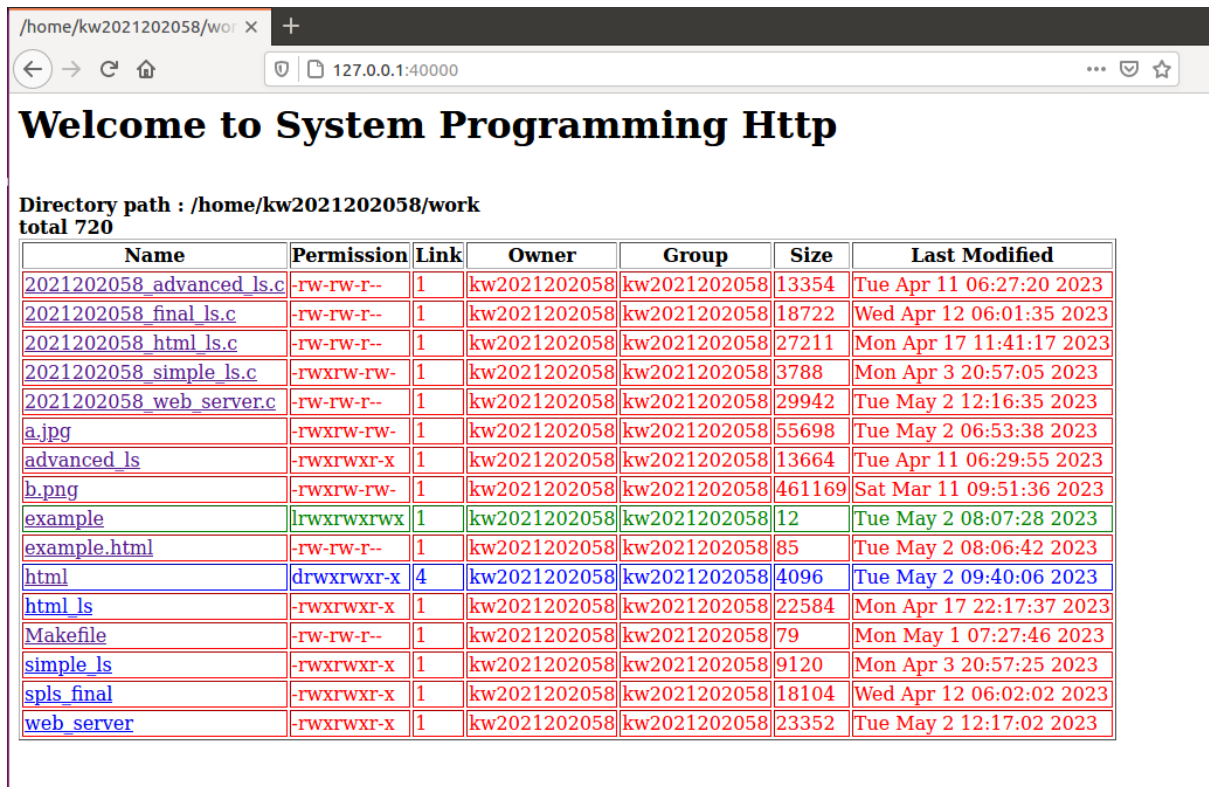
4. 결과화면

```

kw2021202058@ubuntu: ~/work
kw2021202058@ubuntu:~$ cd work
kw2021202058@ubuntu:~/work$ make
gcc -o web_server 2021202058_web_server.c
kw2021202058@ubuntu:~/work$ ./web_server

```

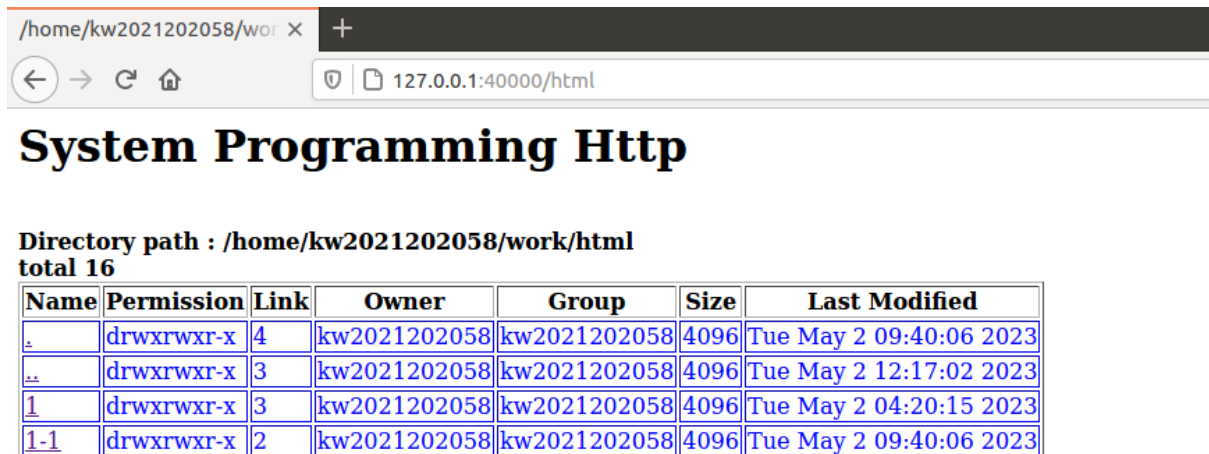
먼저 web_server 을 실행해 server 가 clinet 의 연결을 대기하고 있는 상태이다.



Directory path : /home/kw2021202058/work
total 720

Name	Permission	Link	Owner	Group	Size	Last Modified
2021202058_advanced_ls.c	-rw-rw-r--	1	kw2021202058	kw2021202058	13354	Tue Apr 11 06:27:20 2023
2021202058_final_ls.c	-rw-rw-r--	1	kw2021202058	kw2021202058	18722	Wed Apr 12 06:01:35 2023
2021202058_html_ls.c	-rw-rw-r--	1	kw2021202058	kw2021202058	27211	Mon Apr 17 11:41:17 2023
2021202058_simple_ls.c	-rwxrw-rw-	1	kw2021202058	kw2021202058	3788	Mon Apr 3 20:57:05 2023
2021202058_web_server.c	-rw-rw-r--	1	kw2021202058	kw2021202058	29942	Tue May 2 12:16:35 2023
a.jpg	-rwxrw-rw-	1	kw2021202058	kw2021202058	55698	Tue May 2 06:53:38 2023
advanced_ls	-rwxrwxr-x	1	kw2021202058	kw2021202058	13664	Tue Apr 11 06:29:55 2023
b.png	-rwxrw-rw-	1	kw2021202058	kw2021202058	461169	Sat Mar 11 09:51:36 2023
example	lrwxrwxrwx	1	kw2021202058	kw2021202058	12	Tue May 2 08:07:28 2023
example.html	-rw-rw-r--	1	kw2021202058	kw2021202058	85	Tue May 2 08:06:42 2023
html	drwxrwxr-x	4	kw2021202058	kw2021202058	4096	Tue May 2 09:40:06 2023
html_ls	-rwxrwxr-x	1	kw2021202058	kw2021202058	22584	Mon Apr 17 22:17:37 2023
Makefile	-rw-rw-r--	1	kw2021202058	kw2021202058	79	Mon May 1 07:27:46 2023
simple_ls	-rwxrwxr-x	1	kw2021202058	kw2021202058	9120	Mon Apr 3 20:57:25 2023
spls_final	-rwxrwxr-x	1	kw2021202058	kw2021202058	18104	Wed Apr 12 06:02:02 2023
web_server	-rwxrwxr-x	1	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023

root path 일 때 -l 옵션의 결과가 출력되며, 상단에 "Welcome to System Programming Http"가 출력이 잘 되고 있다.



Directory path : /home/kw2021202058/work/html
total 16

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	4	kw2021202058	kw2021202058	4096	Tue May 2 09:40:06 2023
..	drwxrwxr-x	3	kw2021202058	kw2021202058	4096	Tue May 2 12:17:02 2023
1	drwxrwxr-x	3	kw2021202058	kw2021202058	4096	Tue May 2 04:20:15 2023
1-1	drwxrwxr-x	2	kw2021202058	kw2021202058	4096	Tue May 2 09:40:06 2023

하위 디렉토리 폴더로 이동했을 때 -al 옵션의 결과가 출력되며, 상단에 "System Programming Http"가 출력이 잘 되고 있다.

```

kw2021202058@ubuntu:~/work$ ./web_server
[127.0.0.1 : 18140] client was connected
=====
Request from [127.0.0.1 : 18140]
GET / HTTP/1.1
Host: 127.0.0.1:40000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:88.0) Gecko/20100101 Firefox/88.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

=====
Directory path: /home/kw2021202058/work
total: 720
-rw-rw-r-- 1 kw2021202058 kw2021202058 13354 Tue Apr 11 06:27:20 2023 2021202058_advanced_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 18722 Wed Apr 12 06:01:35 2023 2021202058_final_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 27211 Mon Apr 17 11:41:17 2023 2021202058_html_ls.c
-rwxrwxrwx 1 kw2021202058 kw2021202058 3788 Mon Apr 3 20:57:05 2023 2021202058_simple_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 29942 Tue May 2 12:16:35 2023 2021202058_web_server.c
-rwxrwxrwx 1 kw2021202058 kw2021202058 55698 Tue May 2 06:53:38 2023 a.jpg
-rwxrwxr-x 1 kw2021202058 kw2021202058 13664 Tue Apr 11 06:29:55 2023 advanced_ls
-rwxrwxrwx 1 kw2021202058 kw2021202058 461169 Sat Mar 11 09:51:36 2023 b.png
lrwxrwxrwx 1 kw2021202058 kw2021202058 12 Tue May 2 08:07:28 2023 example
-rw-rw-r-- 1 kw2021202058 kw2021202058 85 Tue May 2 08:06:42 2023 example.html
drwxrwxr-x 4 kw2021202058 kw2021202058 4096 Tue May 2 09:40:06 2023 html
-rwxrwxr-x 1 kw2021202058 kw2021202058 22584 Mon Apr 17 22:17:37 2023 html_ls
-rw-rw-r-- 1 kw2021202058 kw2021202058 79 Mon May 1 07:27:46 2023 Makefile
-rwxrwxr-x 1 kw2021202058 kw2021202058 9120 Mon Apr 3 20:57:25 2023 simple_ls
-rwxrwxr-x 1 kw2021202058 kw2021202058 18104 Wed Apr 12 06:02:02 2023 spls_final
-rwxrwxr-x 1 kw2021202058 kw2021202058 23352 Tue May 2 12:17:02 2023 web_server
[127.0.0.1 : 18652] client was connected
=====
Request from [127.0.0.1 : 18652]
GET /favicon.ico HTTP/1.1
Host: 127.0.0.1:40000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:88.0) Gecko/20100101 Firefox/88.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://127.0.0.1:40000/

=====
[127.0.0.1 : 19164] client was connected

```

client 가 sever 에 HTTP request 를 전송한 결과의 사진이다.

```
127.0.0.1:40000/2021202058_advanced_ls.c
// File Name : 20212058_advanced_ls.c
// Date : 2023/04/05
// Os : Ubuntu 16.04 LTS 64bits
// Author : Song ChaeYoung
// Student ID : 2021202058
// Title : System Programming Assignment #1-2
// Description : This program is a simple implementation of the 'ls' command,
// listing the files and directories in the specified directory,
// along with permissions, owners, groups, sizes, and
// modification times. The program first checks to see if the
// argument has been passed. If arguments are not passed, the
// program lists all files and directories in the current
// directory. When the argument is passed, the program lists all
// files and directories in the specified directory. Save the
// names of the files and directories, sort them, and output
// them according to the options.
//
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <dirent.h>
#include <sys/stat.h>
#include <unistd.h>
#include <pwd.h>
#include <grp.h>
#include <time.h>

#define MAX_PATH_LEN 1024 //이름의 길이
#define MAX_FILES 1000 //파일의 개수

// sort
// Description: Given an array of file names and its count,
// sorts the array in ascending order based on file names.
// If the first character of a file name is '.', it is treated
// as a hidden file. The comparison is case-insensitive.
// Parameters: filename - an array of file names to be sorted
// file count - the number of files in the array
//
void sort(char filename[][1024], int file_count)
{
    char temp[1024];

    for (int i = 0; i < file_count - 1; i++)
    {
        for (int j = i+1; j < file_count; j++)
        {
            char hidden_file1[1024];
            char hidden_file2[1024];

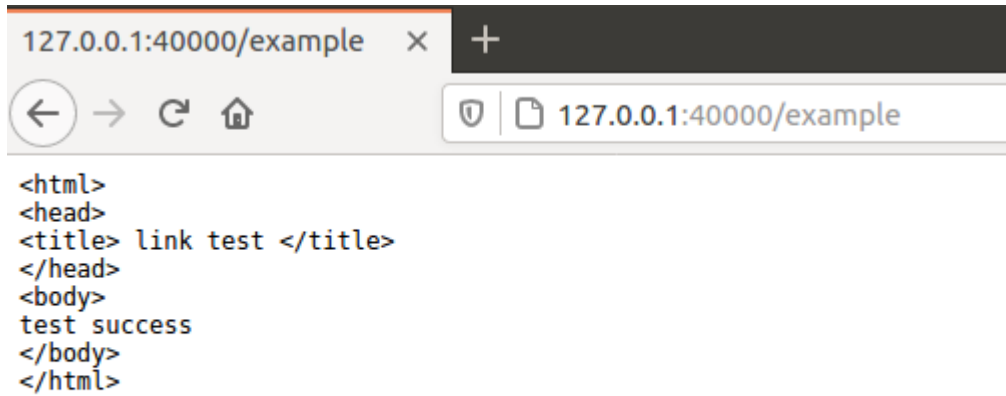
            if(filename[i][0] == '.') //이름이 '.'로 시작하는 파일은 숨김 파일로 처리
            {
                for(int k = 0; k<strlen(filename[i]);k++) //filename[i]의 내용을 hidden_file1에 복사
                {
                    hidden_file1[k] = filename[i][k+1]; //이름의 첫 번째 문자를 생략
                }
            }
            else //이름이 '.'로 시작하지 않는 파일은 일반 파일로 처리
            {
                for(int k = 0; k<strlen(filename[i]);k++) //filename[i]의 내용을 hidden_file2에 복사
                {
                    hidden_file2[k] = filename[i][k];
                }
            }

            //hidden_file1과 hidden_file2를 비교하여 알파벳 순서대로 정렬
            if(strcmp(hidden_file1, hidden_file2) > 0)
            {
                strcpy(temp, hidden_file1);
                strcpy(hidden_file1, hidden_file2);
                strcpy(hidden_file2, temp);
            }
        }
    }
}
```

텍스트파일이나 소스 코드를 클릭했을 때의 결과로 파일의 내용이 출력되는 것을 확인할 수 있다.

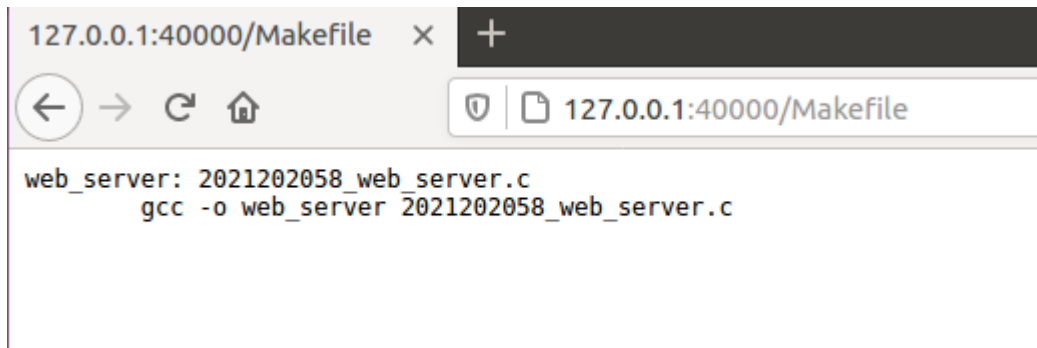
```
link test
test success
```

.html 코드를 클릭했을 때 html 파일의 유형으로 결과가 출력되는 것을 확인할 수 있다.



```
<html>
<head>
<title> link test </title>
</head>
<body>
test success
</body>
</html>
```

링크파일을 클릭 시 원본파일 내용을 출력해야 한다. example
파일은 .html 파일을 링크한 파일이므로, html 형식의 결과가 출력되는
것을 확인할 수 있다.



```
web_server: 2021202058_web_server.c
    gcc -o web_server 2021202058_web_server.c
```

makefile 을 클릭했을 때 파일의 내용을 출력하는 것을 확인할 수 있다.

/home/kw2021202058/work X +
 127.0.0.1:40000

Welcome to System Programming Http

Directory path : /home/kw2021202058/work
 total 720

Name	Permission	Link	Owner	Group	Size	Last Modified
2021202058_advanced_ls.c	-rw-rw-r--	1	kw2021202058	kw2021202058	13354	Tue Apr 11 06:27:20 2023
2021202058_final_ls.c	-rw-rw-r--	1	kw2021202058	kw2021202058	18722	Wed Apr 12 06:01:35 2023
2021202058_html_ls.c	-rw-rw-r--	1	kw2021202058	kw2021202058	27211	Mon Apr 17 11:41:17 2023
2021202058_simple_ls.c	-rwxrwx-rw-	1	kw2021202058	kw2021202058	2788	Mon Apr 20 20:57:05 2023
2021202058_web_server.c	-rw-rw-r--	1	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023
a.jpg	-rwxrwx-rw-	1	kw2021202058	kw2021202058	79	Mon May 1 07:27:46 2023
advanced_ls	-rwxrwxr-x	1	kw2021202058	kw2021202058	9120	Mon Apr 3 20:57:25 2023
b.png	-rwxrwx-rw-	1	kw2021202058	kw2021202058	18104	Wed Apr 12 06:02:02 2023
example	lrwxrwxrwx	1	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023
example.html	-rw-rw-r--	1	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023
html	drwxrwxr-x	4	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023
html_ls	-rwxrwxr-x	1	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023
Makefile	-rw-rw-r--	1	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023
simple_ls	-rwxrwxr-x	1	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023
spls_final	-rwxrwxr-x	1	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023
web_server	-rwxrwxr-x	1	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023

Opening advanced_ls

You have chosen to open:

advanced_ls
which is: application/octet-stream (13.3 KB)
from: http://127.0.0.1:40000

Would you like to save this file?

Cancel Save File

실행파일을 클릭했을 때 파일 download 가 이루어지는 것을 볼 수 있다.

/home/kw2021202058/work X +
 127.0.0.1:40000

Welcome to System Programming Http

Directory path : /home/kw2021202058/work
 total 720

Name	Permission	Link	Owner	Group	Size	Last Modified
2021202058_advanced_ls.c	-rw-rw-r--	1	kw2021202058	kw2021202058	13354	Tue Apr 11 06:27:20 2023
2021202058_final_ls.c	-rw-rw-r--	1	kw2021202058	kw2021202058	18722	Wed Apr 12 06:01:35 2023
2021202058_html_ls.c	-rw-rw-r--	1	kw2021202058	kw2021202058	27211	Mon Apr 17 11:41:17 2023
2021202058_simple_ls.c	-rwxrwx-rw-	1	kw2021202058	kw2021202058	2788	Mon Apr 20 20:57:05 2023
2021202058_web_server.c	-rw-rw-r--	1	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023
a.jpg	-rwxrwx-rw-	1	kw2021202058	kw2021202058	79	Mon May 1 07:27:46 2023
advanced_ls	-rwxrwxr-x	1	kw2021202058	kw2021202058	9120	Mon Apr 3 20:57:25 2023
b.png	-rwxrwx-rw-	1	kw2021202058	kw2021202058	18104	Wed Apr 12 06:02:02 2023
example	lrwxrwxrwx	1	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023
example.html	-rw-rw-r--	1	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023
html	drwxrwxr-x	4	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023
html_ls	-rwxrwxr-x	1	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023
Makefile	-rw-rw-r--	1	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023
simple_ls	-rwxrwxr-x	1	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023
spls_final	-rwxrwxr-x	1	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023
web_server	-rwxrwxr-x	1	kw2021202058	kw2021202058	23352	Tue May 2 12:17:02 2023

Opening a.jpg

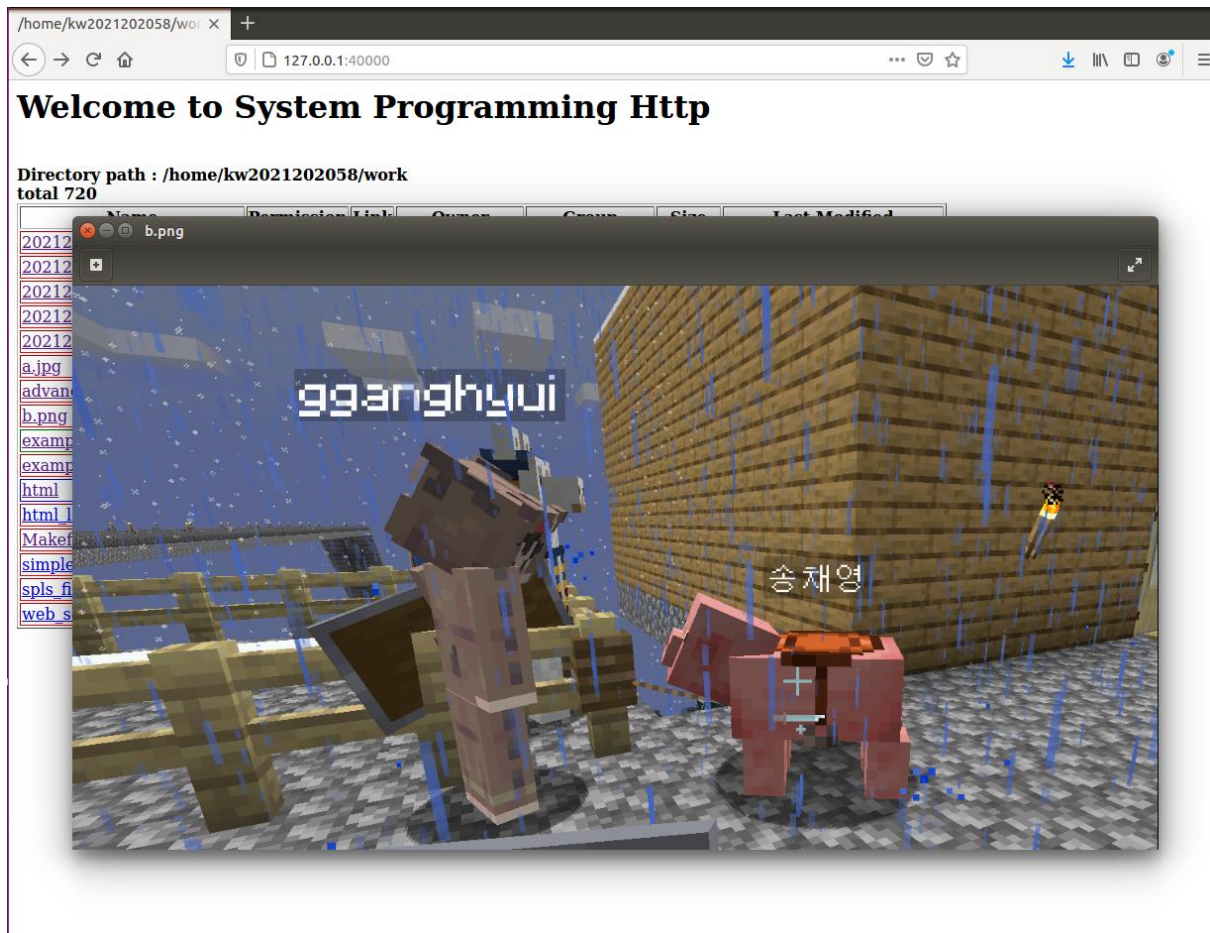
You have chosen to open:

a.jpg
which is: JPEG image (54.4 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



.jpg 파일을 클릭했을 때 open 할지 download 할지 뜨며 open 할 경우 사진이 정상적으로 열리며 download 할 때 역시 정상적으로 download 되는 것을 확인하였다.

/home/kw2021202058/work X +

← → ↻ 🏠 127.0.0.1:40000 ... ☆

Welcome to System Programming Http

Directory path : /home/kw2021202058/work
total 720

Name	Permission	Link	Owner	Group	Size	Last Modified
2021202058_advanced_ls.c	-rw-rw-r--	1	kw2021202058	kw2021202058	13354	Tue Apr 11 06:27:20 2023
2021202058_final_ls.c	-rw-rw-r--	1	kw2021202058	kw2021202058	18722	Wed Apr 12 06:01:35 2023
2021202058_html_ls.c	-rw-rw-r--	1	kw2021202058	kw2021202058	27211	Mon Apr 17 11:41:17 2023
2021202058_simple_ls.c	-rwxrwx-rw-	1	kw2021202058	kw2021202058	2788	Mon Apr 2 20:57:05 2023
2021202058_web_server.c	-rw-rw-r--	1				
a.jpg	-rwxrwx-rw-	1				
advanced_ls	-rwxrwxr-x	1				
b.png	-rwxrwx-rw-	1				
example	lrwxrwxrwx	1				
example.html	-rw-rw-r--	1				
html	drwxrwxr-x	4				
html_ls	-rwxrwxr-x	1				
Makefile	-rw-rw-r--	1				
simple_ls	-rwxrwxr-x	1				
spls_final	-rwxrwxr-x	1				
web_server	-rwxrwxr-x	1				

Opening b.png

You have chosen to open:
b.png
which is: PNG image (450 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

/home/kw2021202058/work X +

← → ↻ 🏠 127.0.0.1:40000 ... ☆

Welcome to System Programming Http

Directory path : /home/kw2021202058/work
total 720

Name
2021202058_advanced_ls.c
2021202058_final_ls.c
2021202058_html_ls.c
2021202058_simple_ls.c
2021202058_web_server.c
a.jpg
advanced_ls
b.png
example
example.html
html
html_ls
Makefile
simple_ls
spls_final
web_server

a.jpg

과제 제출

출제 과제 정보

제목	2-2차 과제
제출기한	2023-04-19 00시00분 ~ 2023-05-03 23시59분
추가제출기한	-
제출방법	개인별
상태	미제출

web_server 과제 공유합니다.

파일 명: 23-1_SPLab_Web_2-2_v2.pdf

내요/주의사항

.png 파일을 클릭했을 때 open 할지 download 할지 뜨며 open 할 경우 사진이 정상적으로 열리며 download 할 때 역시 정상적으로 download 되는 것을 확인하였다.



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:40000/html/1/2/3'. The page title is 'System Programming Http'. Below the title, the directory path is shown as '/home/kw2021202058/work/html/1/2/3' with a total of 8 items. A table lists the directory contents:

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	2	kw2021202058	kw2021202058	4096	Tue May 2 04:20:30 2023
..	drwxrwxr-x	3	kw2021202058	kw2021202058	4096	Tue May 2 04:20:30 2023

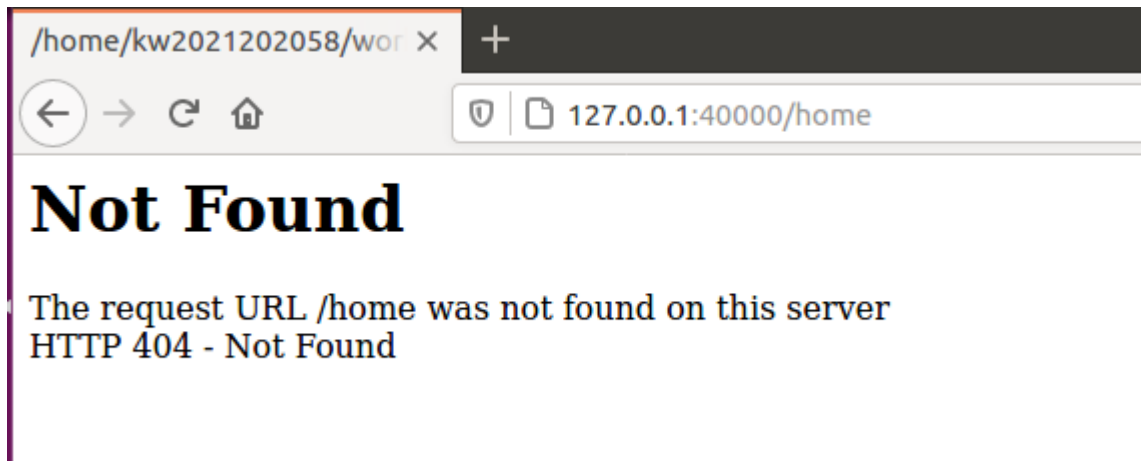
하위 디렉토리 폴더 -> 하위 디렉토리 폴더 -> 하위 디렉토리 폴더 -> 하위 디렉토리 폴더로 이동했을 때의 결과로 결과가 잘 출력되는 것을 알 수 있다.



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:40000/html/1/2'. The page title is 'System Programming Http'. Below the title, the directory path is shown as '/home/kw2021202058/work/html/1/2' with a total of 12 items. A table lists the directory contents:

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	3	kw2021202058	kw2021202058	4096	Tue May 2 04:20:30 2023
..	drwxrwxr-x	3	kw2021202058	kw2021202058	4096	Tue May 2 04:20:15 2023
3	drwxrwxr-x	2	kw2021202058	kw2021202058	4096	Tue May 2 04:20:30 2023

filename 이 .일때와 ..일때도 잘 작동하는 것을 확인하였다.



존재하지 않는 디렉토리의 url 을 입력했을 때, 예시로는 home 을 입력했을 때 에러 메시지를 출력해주었다.

5. 고찰

이번 과제를 수행하면서 list_files 함수, 파일 목록을 출력하는 함수에서 block 을 0 으로 초기화 해주지 않아 결과에서 새로 고침 할 때 마다 total 이 계속해서 늘어났다. 코드의 다른 부분에서도 0 으로 초기화를 해주어 오류를 잡은 부분이 많았고 초기화의 중요성에 대해 다시 한 번 깨달았다. 또한 FNM_CASEFOLD 옵션을 위해 #define _GNU_SOURCE 을 해주었는데, 헤더파일 아래에 선언해주었을 때는 오류가 발생했다. #define 은 컴파일러가 코드를 해석하기 전에 먼저 처리가 되어야 한다는 것을 알고 있었지만, 헤더파일 위에 선언하지 않았을 때 오류가 생긴다는 점은 처음 알게 된 것 같다. 또한 print_file_info, 파일 정보를 출력하는 함수에서 url 을 계속 바꿔주었는데, 전역변수로 url 을 뺏기 때문에 url 의 값을 바꾸면 오류가 생겼던 것 같다. 따라서 임시변수를 선언해 복사해서 넣어준 후 사용하도록 하였다. response_message 역시 temp_message 를 선언해 계속 문자열을 추가해 준 후 마지막에 한 번에 넣어주도록 하였다. 또한 파일 사이즈만큼 클라이언트에게 파일을 전송해주는 부분에서 처음에는 while 문을 사용하지 않고 fread 함수를 사용하였는데, 코드가 끝까지 나오지 않아 하나씩 버퍼사이즈만큼 파일을 전송하도록 수정하였고 이 과정을 통해, while 문 안에 fread 를 직접 넣어주어 해결할 수도 있다는 것을 직접 해보며 알았다.

6. Reference

시스템프로그래밍실습 강의자료