

# 시스템프로그래밍실습 보고서

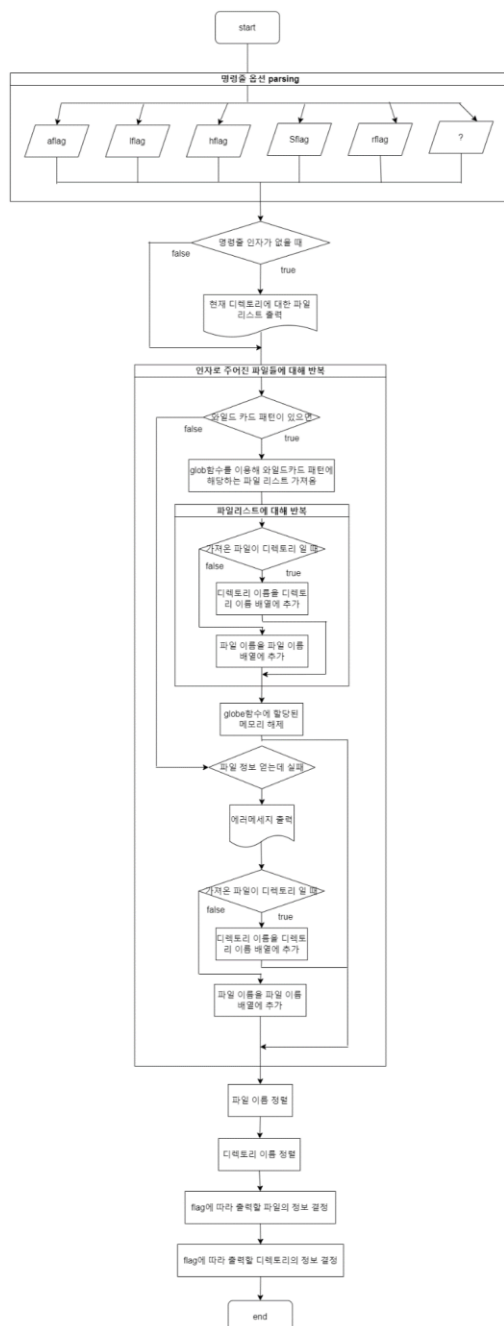
## Assignment 2-1

과목	시스템프로그래밍실습
담당교수	이기훈교수님
학과	컴퓨터정보공학부
학번	2021202058
이름	송채영

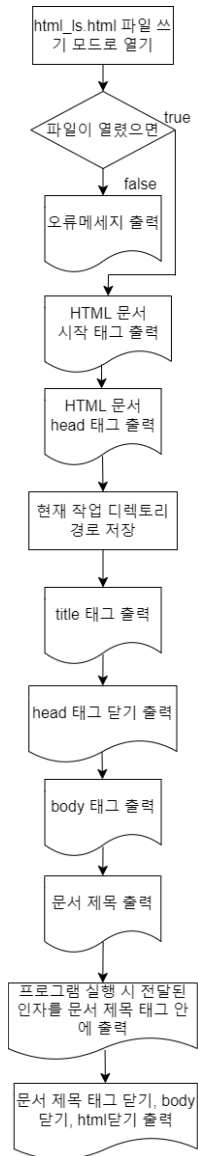
## 1. Introduction

HTML Tags 에 대해 알아보고 HTML Tag(제목 heading, title, 테이블 table, tr, td, 줄바꿈 br, 링크 a 등)를 조합해서 활용하며 1-3 과제(final\_ls)를 바탕으로 결과를 HTML 형식으로 출력해본다. HTML 형식으로 출력된 결과물을 웹 브라우저를 통해 확인하며 file 에 hyperlink 를 달아보고, file type 에 따라 색을 달아본다.

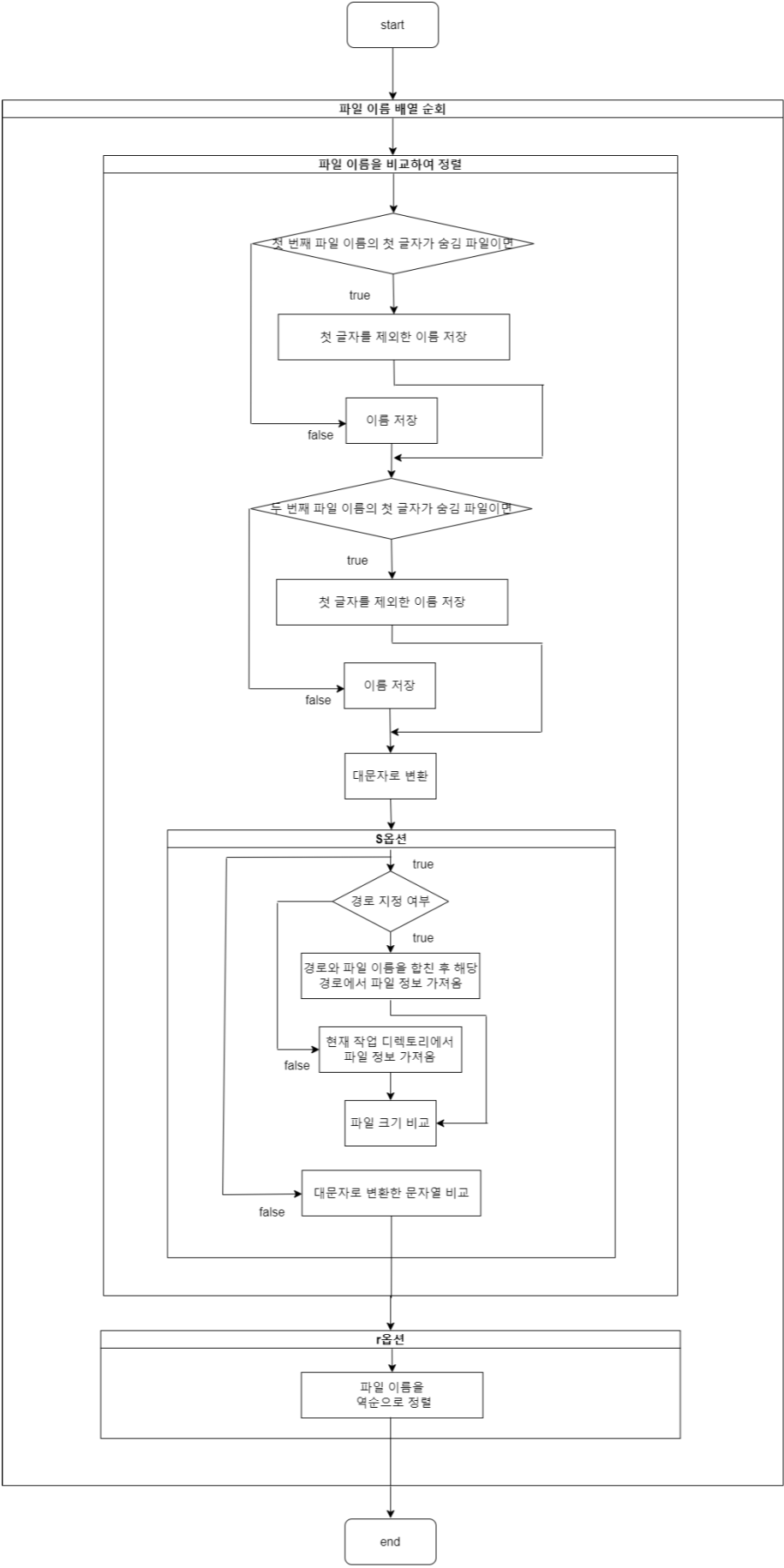
## 2. Flow chart



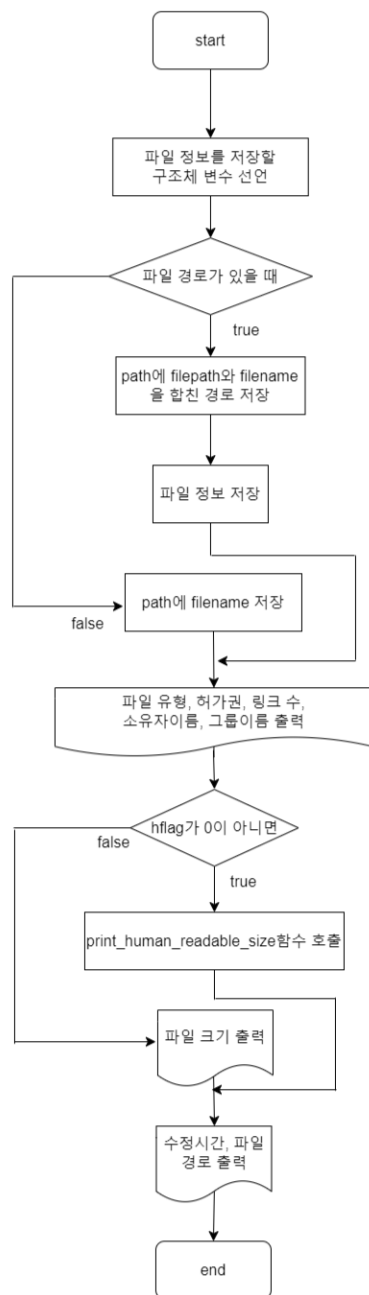
코드의 전체적인 흐름을 flow chart 로 나타내었다. 우선 main 함수에 대해 자세히 설명해보면 다음과 같다. 프로그램이 시작하면 getopt 함수를 사용해 명령줄 인자를 parsing 한다. getopt 함수는 argv 배열과 인자의 개수 argc, 그리고 옵션 문자열을 입력으로 받아 옵션 문자 하나를 읽어온다. 옵션이 있는지 없는지는 flag 로 판단한다. 인자가 없을 때, 현재 디렉토리에 대한 파일 리스트를 출력하며, 인자가 있을 때, 인자로 주어진 파일들에 대해 반복하며 와일드카드 패턴이 있으면, glob 함수를 이용해 와일드카드 패턴에 해당하는 파일들의 리스트를 가져온다. 가져온 파일이 디렉토리 일 때 디렉토리 이름을 디렉토리 이름 배열에 추가하며 파일일땐 파일 이름을 파일 이름 배열에 추가한 후 glob 함수에 할당된 메모리를 해제한다. 와일드카드 패턴이 없으면 파일 정보를 얻는 데 실패한 경우 에러 메시지를 출력해주며, 현재 가져온 파일이 디렉토리면, 디렉토리 이름을 디렉토리 이름 배열에, 파일이면 파일 이름을 파일 이름 배열에 추가한다. 이후 파일이름과 디렉토리 이름을 정렬해준 후 flag 에 따라 출력할 파일의 정보를 결정한다.



main 함수에서 추가된 html 부분만 flow chart로 나타낸 것이다. 다음과 같다. html.js.html 파일을 쓰기 모드로 열고 파일 포인터를 file 변수에 저장해주었다. 파일이 성공적으로 열리지 않았으면 오류 메시지를 출력하고 프로그램을 종료한다. HTML 문서의 시작 태그인 "<html>"을 출력하고 head 태그인 "<head>"를 출력하였다. 현재 작업 디렉토리 경로를 cwd 변수에 저장하고 "<title>" 태그를 출력하고, 그 안에 현재 작업 디렉토리 경로를 출력하였다. head 태그를 닫기 위해 "</head>"를 출력하고 body 태그인 "<body>"를 출력한다. 다음으로 문서 제목을 출력하기 위해 "<h1>" 태그를 출력한 후 프로그램 실행 시 전달된 인자들을 "<h1>" 태그 안에 출력한 후 "<h1>" 태그와 "<body>" 그리고 "<html>" 태그를 닫아주었다.

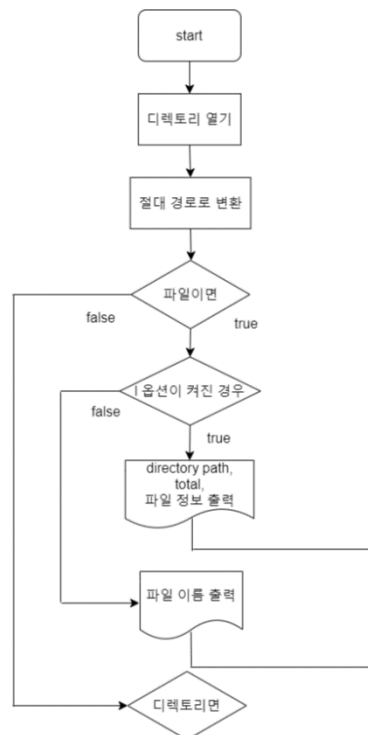
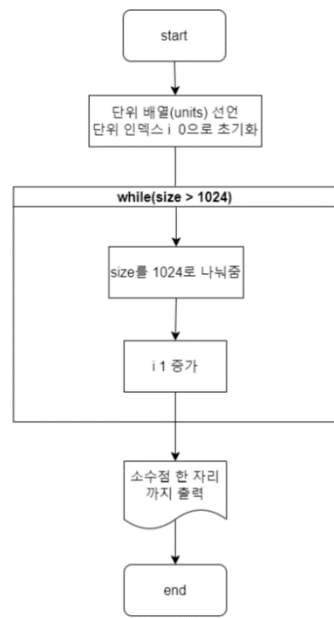


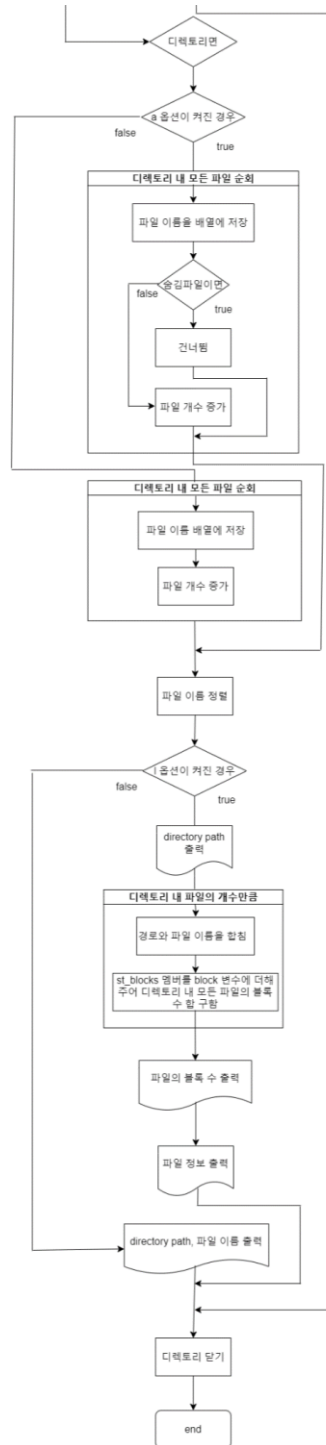
다음은 sort 함수의 flow chart 이다. 우선 파일의 개수만큼 반복한다. 첫 번째 파일이 숨김 파일일 경우, 파일의 이름을 길이만큼 반복하고 첫 글자를 제외한 이름을 저장한다. 숨김파일이 아닐 경우 이름을 저장해준다. 두 번째 파일도 동일하게 진행한다. 먼저 대문자로 변환해준 후 s 옵션을 선택한 경우와 아닌 경우로 나눈다. 전자는 경로가 지정되지 않은 경우 파일 이름만 stat 에 넣어주며, 경로가 있는 경우 경로와 파일 이름을 합친다. 이후 파일 크기를 비교해준다. 후자는 대문자로 변환한 문자열을 비교한다. r 옵션을 선택한 경우 파일 이름을 역순으로 정렬한다. 프로그램을 종료한다.



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 경로와 파일 이름을 합치며, 파일 경로가 없는 경우 파일이름을 정보로 읽는다. 파일 유형, 허가권, 링크 수 , 소유자를 출력한다. hflag 가 0 이 아니면 함수를 호출해주며, 0 일 경우 파일 크기를 출력한다. 이어서 수정 시간, 파일 경로를 출력한 후 프로그램을 종료한다.

위 함수에서 html 에 파일 정보를 출력해주기 위해 html 코드로 출력하는 부분을 추가했다. flow chart 의 흐름은 기존과 변함이 없다.

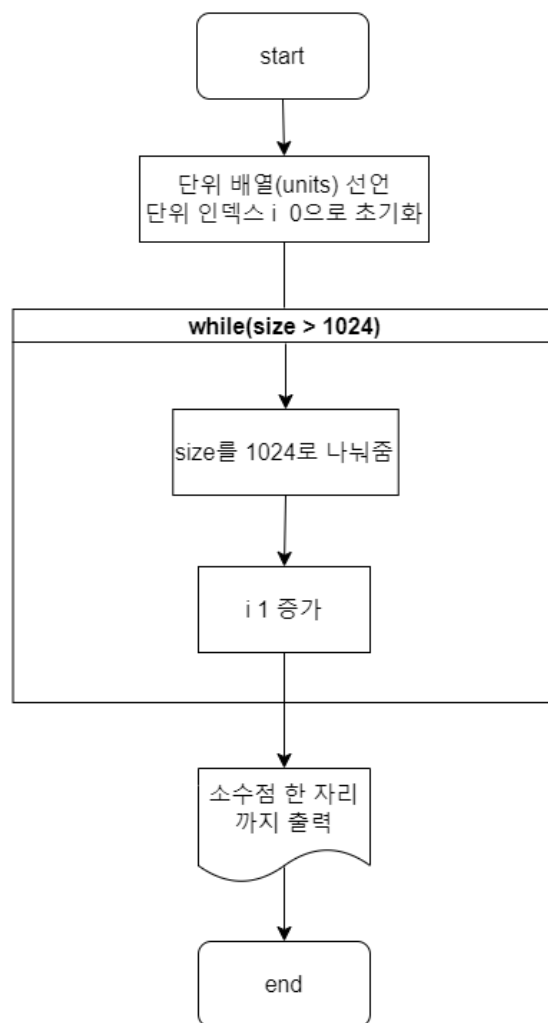




다음은 list files 함수의 flow chart 이다. 파일을 열어준 후 절대경로로 변환한다. 연  
 파일이 파일일때와 디렉토리 일 때로 나눈다. 우선 파일이고 l 옵션이 켜져 있으면  
 directory path 와 total 과 파일 정보를 출력한다. l 옵션이 꺼져 있을 경우 파일 이름만  
 출력한다. 디렉토리일 경우 먼저 -a 옵션이 꺼져 있으면, 디렉토리 내의 모든 파일을  
 순회하며, 파일 이름을 배열에 저장한다. 이때 숨김 파일이면 건너뛰며 아닐 경우 파일  
 개수를 증가한다. a 옵션이 켜져 있을 경우, 디렉토리 내의 모든 파일을 순회하며 파일  
 이름을 배열에 저장한다. 이후 파일 이름을 정렬한다. 다음으로 l 옵션이 켜져 있을 경우

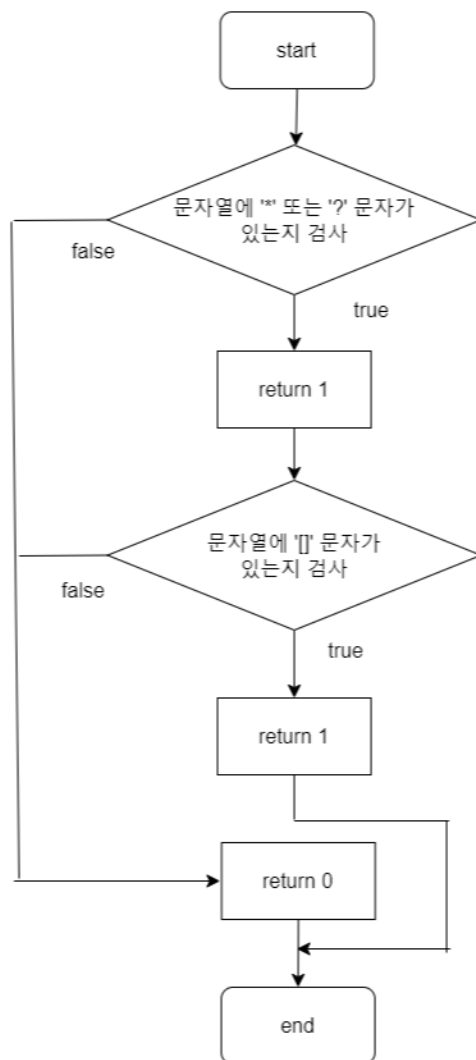
directory path, total 을 출력한다. 디렉토리 내의 파일 개수만큼 경로와 파일을 합친 후 block 변수에 st\_blocks 멤버를 더해주어 디렉토리 내의 모든 파일의 블록수의 합을 구한 후 출력해준다. 파일 정보를 출력한다. l 옵션이 꺼져 있을 경우, directory path 와 파일 이름을 출력하고 디렉토리를 닫고 프로그램을 종료한다.

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.



다음은 `print_human_readable_size` 함수이며, 먼저 단위 배열인 `units` 를 선언하고 단위 인덱스 `i` 를 0 으로 초기화한다. `size` 가 1024 가 넘어가면 단위를 증가시키며 나눠준다. 소수점 한 자리까지 출력해주고 프로그램을 종료한다.





마지막으로 wildcardpattern 함수이다. 먼저 문자열에 '\*', '?' 문자가 있는지 확인하여 있으면 return 1 을 해준다. 문자열에 '[' 문자가 있는지 확인해서 문자가 있으면 return 1 을 한다. 와일드카드 문자가 없으면 0 을 반환하며 프로그램을 종료한다.

### 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
        }
        대문자로 변환
        if(S옵션을 선택한 경우)
        {
            경로가 지정되지 않은 경우 파일 이름 저장
            경로가 있으면 경로와 파일 이름을 합쳐 저장
        }
        파일 크기 비교
        else
        {
            대문자로 변환한 문자열 비교
        }
    }
    if(r 옵션을 선택한 경우)
    {
        파일 이름을 역순으로 정렬
    }
}

```

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
    }
    // 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

    if(h옵션이 0이 아닌 경우 파일 크기를 사람이 읽을 수 있는 형태로 출력)
    else(파일 크기 출력)
    수정시간, 파일 경로 출력
}

```

print\_file\_info 함수의 pseudo code 이다.

```

인자로 전달된 파일 크기에 대한 단위를 계산하여 출력하는 함수 //print_human_readable_size
{
    단위 배열 선언
    단위 인덱스 선언 & 초기화
    size가 1024가 넘어가면 단위를 증가시키며 나눔
    소수점 첫번째 자리까지 출력함
}

```

print\_human\_readable\_size 함수의 pseudo code 이다.

```

파일 목록 출력() //list_files
{
    디렉토리 열기
    절대경로로 변환
    if(파일이면)
    {
        if(1옵션이 켜진경우)
        {
            directory path출력
            total출력
            파일 이름 출력
            HTML 파일에 테이블 태그 삽입
            HTML 파일에 테이블 헤더 삽입
            파일 정보 출력
        }
        else
        {
            HTML 파일에 테이블 태그 삽입
            HTML 파일에 파일 이름 헤더 삽입
            파일 이름 출력
        }
        HTML 파일에 테이블 태그 닫기
    }
    else
    {
        if(a 옵션이 꺼져있는 경우)
        {
            while(디렉토리 내 모든 파일 순회)
            {
                html_ls.html 파일은 건너뛴
                숨김파일이면 넘어감
                아니면 파일 개수 증가
            }
        }
        else
        {
            while(디렉토리 내 모든 파일 순회)
            {
                html_ls.html 파일은 건너뛴
                파일 이름 배열에 저장
            }
        }
        파일 이름 정렬
        if(1 옵션이 켜져있는 경우)
        {
            directory path, total 출력
            HTML 파일에 테이블 태그 출력
            HTML 파일에 테이블 헤더 삽입
            파일 정보 출력
        }
        else
        {
            HTML 파일에 테이블 태그 출력
            HTML 파일에 이름 컬럼, 파일 이름 출력
            파일 이름 배열에 저장
        }
    }
    디렉토리 닫기
    HTML 파일에 테이블 태그 닫기
}

```

list\_files 함수의 pseudo code 이다.

```

파일 이름에 와일드 카드 문자가 있는지 검사하는 함수 //wildcardpattern
{
    if( 문자열에 '*' 또는 '?' 문자가 있는지 검사)
        있으면 1반환
    문자열에 '[' 문자가 있는지 검사
        있으면 1 반환
    없으면 0 반환
}

```

wildcardpattern 함수의 pseudo code 이다.

```

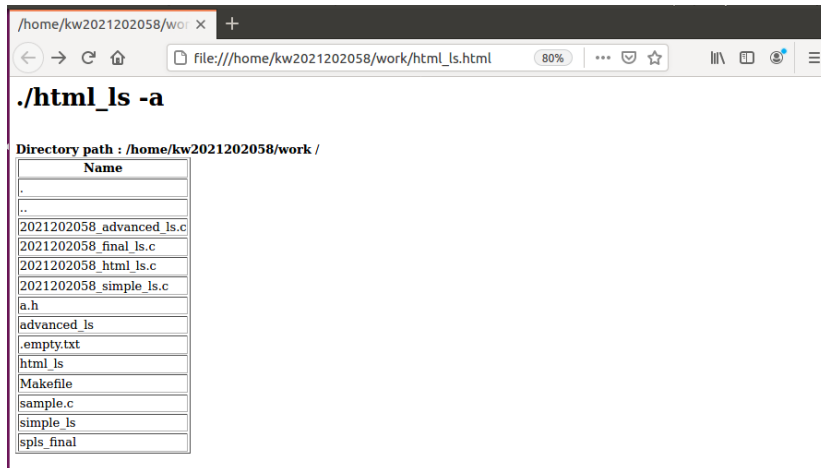
main(argc, argv)
{
    html_ls.html 파일을 쓰기 모드로 열고 파일 포인터를 file 변수에 저장
    if(파일이 열리지 않으면)
        에러메세지 출력 후 -1 반환
    HTML 문서 시작 태그 출력
    HTML head 태그 출력
    현재 작업 디렉토리를 cwd에 저장
    타이틀 태그 출력
    head 태그 닫기, body 태그 출력
    h1 태그 출력 후 프로그램 실행 시 전달된 인자를 h1 태그 내에 출력
    h1 태그 닫기

    명령줄 옵션 parsing
    {
        a 옵션일 때 aflag +1
        l 옵션일 때 lflag +1
        h 옵션일 때 hflag +1
        S 옵션일 때 Sflag +1
        r 옵션일 때 rflag +1
        알 수 없는 옵션일 때 에러메세지 출력
    }
    if(인자가 없을 때)
    {
        현재 디렉토리에 대한 파일 리스트 출력
    }
    else(인자가 있을 때)
    {
        if(와일드카드 패턴이 있으면)
        {
            glob함수를 이용해 와일드카드 패턴에 해당하는 파일 리스트 가져옴
            for(파일 리스트 반복)
                파일 이름이 "html_ls.html"과 일치하면 건너뛸
                폴더이면 폴더 배열에 저장
                파일이면 파일 배열에 저장
        }
        glob함수에 할당된 메모리 해제
        else
        {
            파일 정보 얻는데 실패시 에러메세지 출력
            가져온 파일이 폴더이면 폴더 배열에 저장
            가져온 파일이 파일이면 파일 배열에 저장
        }
    }
    파일 이름 정렬
    디렉토리 이름 정렬
    flag에 따라 출력할 파일 정보 결정
}

```

main 함수의 pseudo code 이다.

#### 4. 결과화면

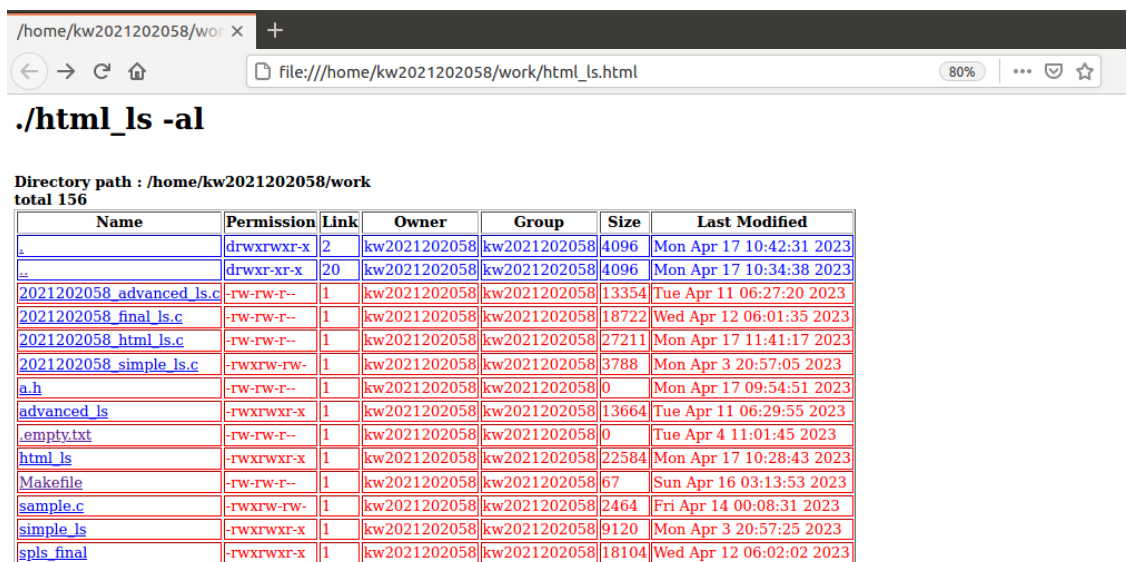


우선 title 에 current working directory 가 뜨는 것을 볼 수 있으며, head 에는 옵션이 뜨는 것을 볼 수 있다. 결과를 table 을 이용하여 출력하였으며 -a 옵션을 했을 때 파일 이름만 출력하는 것을 확인할 수 있다. 또한 html\_ls.html 파일은 출력 되지 않는 것 역시 확인할 수 있다.

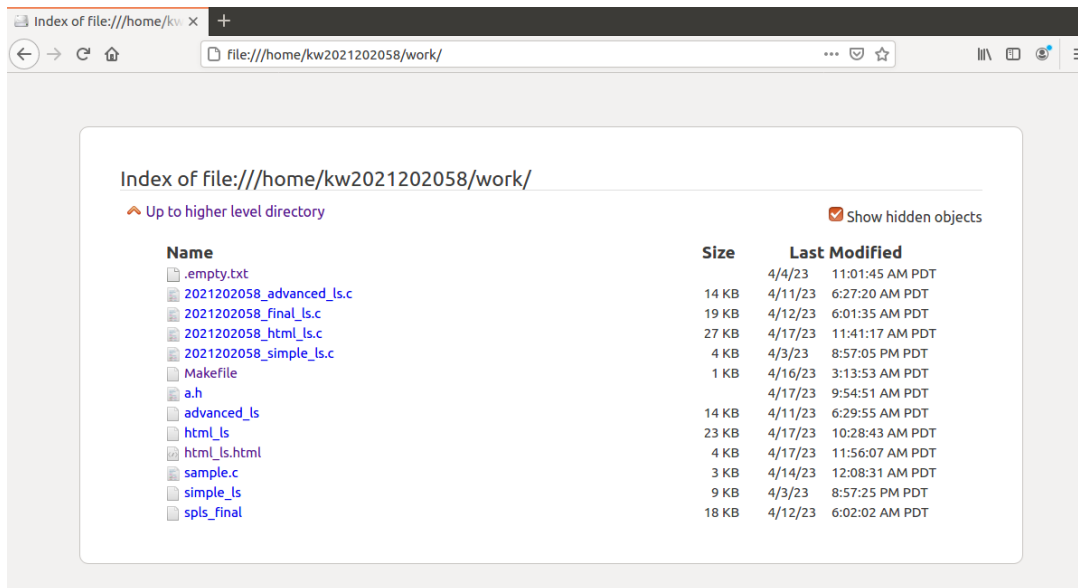
'l' 옵션일 때 permission 을 알 수 있으며 그 외의 경우 파일의 정보를 알 수 없으므로 색을 출력해주지 않았다.

```
home > kw2021202058 > work > <html_ls.html> <html>
1 |<html>
2 |<head>
3 |<title> /home/kw2021202058/work </title>
4 |</head>
5 |<body>
6 |<h1> ./html_ls -a</h1><br><b> Directory path : /home/kw2021202058/work</b>
7 |<table border= "1">
8 |<tr><th>Name</th></tr>
9 |<tr><td>.</td></tr><b><tr><td>..</td></tr><b><tr><td>2021202058_advanced_ls.c</td></tr><b><tr><td>2021202058_f
10 |</body>
11 |</html>
```

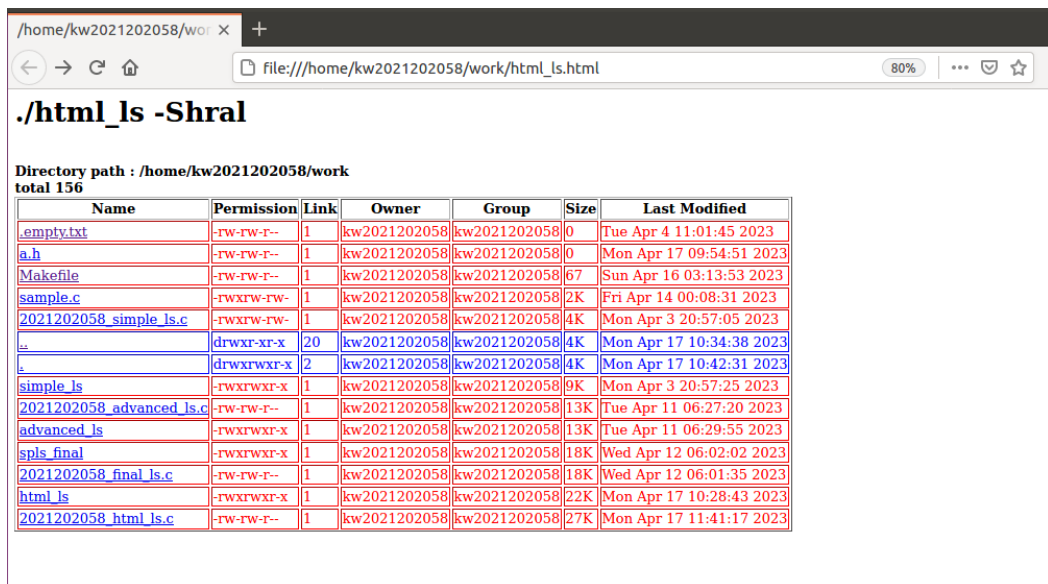
위의 사진의 html\_ls.html 코드로, html 출력 형식을 확인할 수 있다.



-al 옵션일 때의 출력결과로, directory 는 파란색으로, 나머지는 빨간색으로 색을 나타낸 것을 확인할 수 있다. 역시 html\_ls.html 파일은 출력되지 않는다.



''폴더를 클릭했을 때의 결과이다.



모든 옵션을 사용하여 출력한 결과를 보여주는 사진이며, 사이즈의 순서대로 정렬하여 출력하였고 size 의 크기가 같을 경우 역순으로 출력하는 것을 확인할 수 있다. file type 에 알맞게 색도 출력해주었고, 역시 html\_ls.html 파일이 출력되지 않는 것을 볼 수 있다.

file:///home/kw2021202058/work/html\_ls.html

## ./html\_ls -al /var

Directory path : /var  
total 56

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxr-xr-x	14	root	root	4096	Mon Jul 30 17:50:33 2018
..	drwxr-xr-x	24	root	root	4096	Tue Apr 4 06:40:37 2023
backups	drwxr-xr-x	2	root	root	4096	Wed Apr 12 05:59:33 2023
cache	drwxr-xr-x	16	root	root	4096	Fri Mar 31 00:07:15 2023
crash	drwxrwxrwx	2	root	whoopsie	4096	Mon Jul 30 17:40:49 2018
lib	drwxr-xr-x	66	root	root	4096	Fri Mar 31 00:12:07 2023
local	drwxrwxr-x	2	root	staff	4096	Tue Apr 12 13:14:23 2016
lock	lrwxrwxrwx	1	root	root	0	Fri Mar 31 00:03:20 2023
log	drwxrwxr-x	13	root	syslog	4096	Mon Apr 17 09:37:25 2023
mail	drwxrwxr-x	2	root	mail	4096	Mon Jul 30 17:30:31 2018
metrics	drwxrwxrwx	2	root	whoopsie	4096	Mon Jul 30 17:40:49 2018
opt	drwxr-xr-x	2	root	root	4096	Mon Jul 30 17:30:31 2018
run	lrwxrwxrwx	1	root	root	4	Fri Mar 31 00:03:20 2023
snap	drwxr-xr-x	2	root	root	4096	Thu Jul 19 03:48:02 2018
spool	drwxr-xr-x	7	root	root	4096	Mon Jul 30 17:37:04 2018
tmp	drwxrwxrwx	14	root	root	4096	Mon Apr 17 09:38:55 2023

link file 일 때 색이 잘 출력되는지 확인해 보았으며, 초록색으로 알맞게 뜨는 것을 볼 수 있었다.

file:///run/lock/

## Index of file:///run/lock/

Up to higher level directory

Name	Size	Last Modified
asound.state.lock	1 KB	4/17/23 9:37:23 AM PDT
subsys		4/17/23 9:37:17 AM PDT
whoopsie		4/17/23 9:37:31 AM PDT

lock 링크 파일을 클릭했을 때의 결과이다.

file:///home/kw2021202058/work/html\_ls.html

## ./html\_ls -al . Makefile

Directory path : /home/kw2021202058/work/Makefile  
total 4

Name	Permission	Link	Owner	Group	Size	Last Modified
Makefile	-rw-rw-r--	1	kw2021202058	kw2021202058	67	Sun Apr 16 03:13:53 2023

Directory path : /home/kw2021202058/work  
total 156

Name	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	2	kw2021202058	kw2021202058	4096	Mon Apr 17 10:42:31 2023
..	drwxr-xr-x	20	kw2021202058	kw2021202058	4096	Mon Apr 17 10:34:38 2023
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	-rw-rw-rw-	1	kw2021202058	kw2021202058	3788	Mon Apr 3 20:57:05 2023
a.h	-rw-rw-r--	1	kw2021202058	kw2021202058	0	Mon Apr 17 09:54:51 2023
advanced_ls	-rwxrwxr-x	1	kw2021202058	kw2021202058	13664	Tue Apr 11 06:29:55 2023
empty.txt	-rw-rw-r--	1	kw2021202058	kw2021202058	0	Tue Apr 4 11:01:45 2023
html_ls	-rwxrwxr-x	1	kw2021202058	kw2021202058	22584	Mon Apr 17 10:28:43 2023
Makefile	-rw-rw-r--	1	kw2021202058	kw2021202058	67	Sun Apr 16 03:13:53 2023
sample.c	-rwxrwr-rw-	1	kw2021202058	kw2021202058	2464	Fri Apr 14 00:08:31 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

두 개의 인자를 줬을 때를 확인해 본 것으로, 각각의 결과를 출력하는 것을 알 수 있다.

```
kw2021202058@ubuntu:~/work$ ./html_ls -al '*.c'
Directory path: /home/kw2021202058/work/2021202058_advanced_ls.c
2021202058_advanced_ls.c
total: 16
-rw-rw-r-- 1 kw2021202058 kw2021202058 13354 Tue Apr 11 06:27:20 2023 2021202058_advanced_ls.c
Directory path: /home/kw2021202058/work/2021202058_final_ls.c
2021202058_final_ls.c
total: 20
-rw-rw-r-- 1 kw2021202058 kw2021202058 18722 Wed Apr 12 06:01:35 2023 2021202058_final_ls.c
Directory path: /home/kw2021202058/work/2021202058_html_ls.c
2021202058_html_ls.c
total: 28
-rw-rw-r-- 1 kw2021202058 kw2021202058 27211 Mon Apr 17 11:41:17 2023 2021202058_html_ls.c
Directory path: /home/kw2021202058/work/2021202058_simple_ls.c
2021202058_simple_ls.c
total: 4
-rwxrwx-rw- 1 kw2021202058 kw2021202058 3788 Mon Apr 3 20:57:05 2023 2021202058_simple_ls.c
sample.c
total: 4
-rwxrwx-rw- 1 kw2021202058 kw2021202058 2464 Fri Apr 14 00:08:31 2023 sample.c
kw2021202058@ubuntu:~/work$
```

마지막으로 wild card pattern 을 넣었을 때의 결과를 확인해보았다.

/home/kw2021202058/work

file:///home/kw2021202058/work/html\_ls.html

# ./html\_ls -al \*.c

Directory path : /home/kw2021202058/work/2021202058\_advanced\_ls.c  
total 16

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

Directory path : /home/kw2021202058/work/2021202058\_final\_ls.c  
total 20

Name	Permission	Link	Owner	Group	Size	Last Modified
2021202058_final_ls.c	-rw-rw-r--	1	kw2021202058	kw2021202058	18722	Wed Apr 12 06:01:35 2023

Directory path : /home/kw2021202058/work/2021202058\_html\_ls.c  
total 28

Name	Permission	Link	Owner	Group	Size	Last Modified
2021202058_html_ls.c	-rw-rw-r--	1	kw2021202058	kw2021202058	27211	Mon Apr 17 11:41:17 2023

Directory path : /home/kw2021202058/work/2021202058\_simple\_ls.c  
total 4

Name	Permission	Link	Owner	Group	Size	Last Modified
2021202058_simple_ls.c	-rwxrwx-rw-	1	kw2021202058	kw2021202058	3788	Mon Apr 3 20:57:05 2023

Directory path : /home/kw2021202058/work/sample.c  
total 4

Name	Permission	Link	Owner	Group	Size	Last Modified
sample.c	-rwxrwx-rw-	1	kw2021202058	kw2021202058	2464	Fri Apr 14 00:08:31 2023

다음과 같이 .c 로 끝나는 파일들만 출력되는 것을 확인할 수 있다.

## 5. 고찰

1-3 과제를 바탕으로 이번 과제를 진행했기 때문에, 코드를 새롭게 구현하는 점 보다는, html 과 html tag 를 사용하는 방법을 알아보고 적응하고 익숙해지는데 시간이 걸렸던 것 같다. 1-3 과제를 수행하면서 기존의 코드를 조금 수정해주었다. 파일 정보 출력 함수에서, 파일의 정보를 stat 으로 읽었는데, lstat 으로 읽게끔 바꿔주었다. stat 의 경우 파일의 경로를 입력 받아 파일의 모든 정보를 반환하지만 lstat 은 링크 파일의 경우 해당 파일이 가리키는 원본 파일의 정보를 반환한다. 따라서 링크 파일의 경우도 가능하게 해주기 위해 바꿔주었다. 또한 파일 경로가 없는 경우 stat 으로 읽어오지 않아 seg 오류가 났었는데, 그 부분을 해결해주었다. 또한 기존 코드는 디렉토리일 경우와 아닐 때로 구분했지만, 이번 과제에서는 링크 파일일때를 구분하기 위해서 파일 유형에 따라 permission 배열의 0 번째 자리를 결정하게끔 코드를 수정했다.



## 6. Reference

시스템프로그래밍실습 강의자료