

시스템프로그래밍

과제이름: Assignment 2-1

- 담당교수: 김 태 석 교수님
- 학 과: 컴퓨터정보공학부
- 학 번: 2019202021
- 이 름: 정 성 업
- 제출일: 2023/4/19

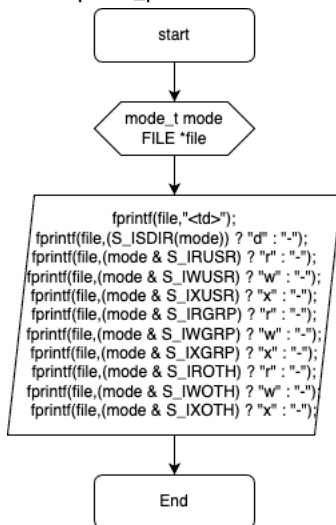
1. Introduction

A. 과제 소개

이번 Assignment 2-1 과제는 이전 Assignment 1-3에서 만들었던 ls를 활용하여 html파일을 제작한다. 이때 current directory는 title로 지정하고 command는 heading으로 출력한다. ls의 결과는 table을 사용해서 출력하며 각 file_name은 하이퍼링크로 해당 경로를 연결한다. table로 결과를 출력할 때는 directory는 파란색 링크 파일은 초록색, 나머지는 빨간색으로 출력한다.

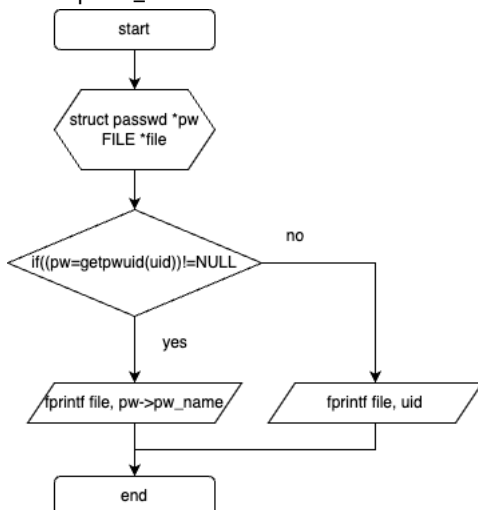
2. Flow chart

A. print_permission



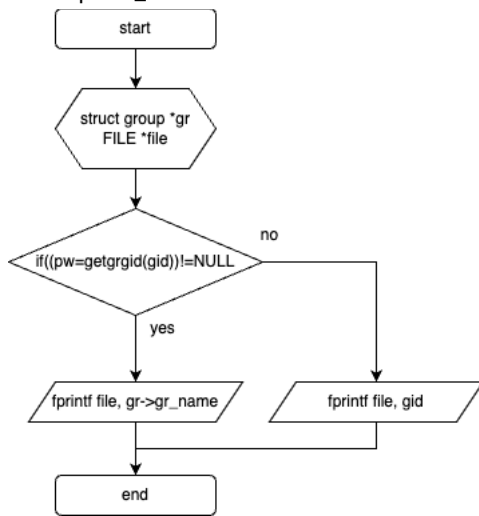
파일의 허용정보를 출력하는 함수이다.

B. print_UID



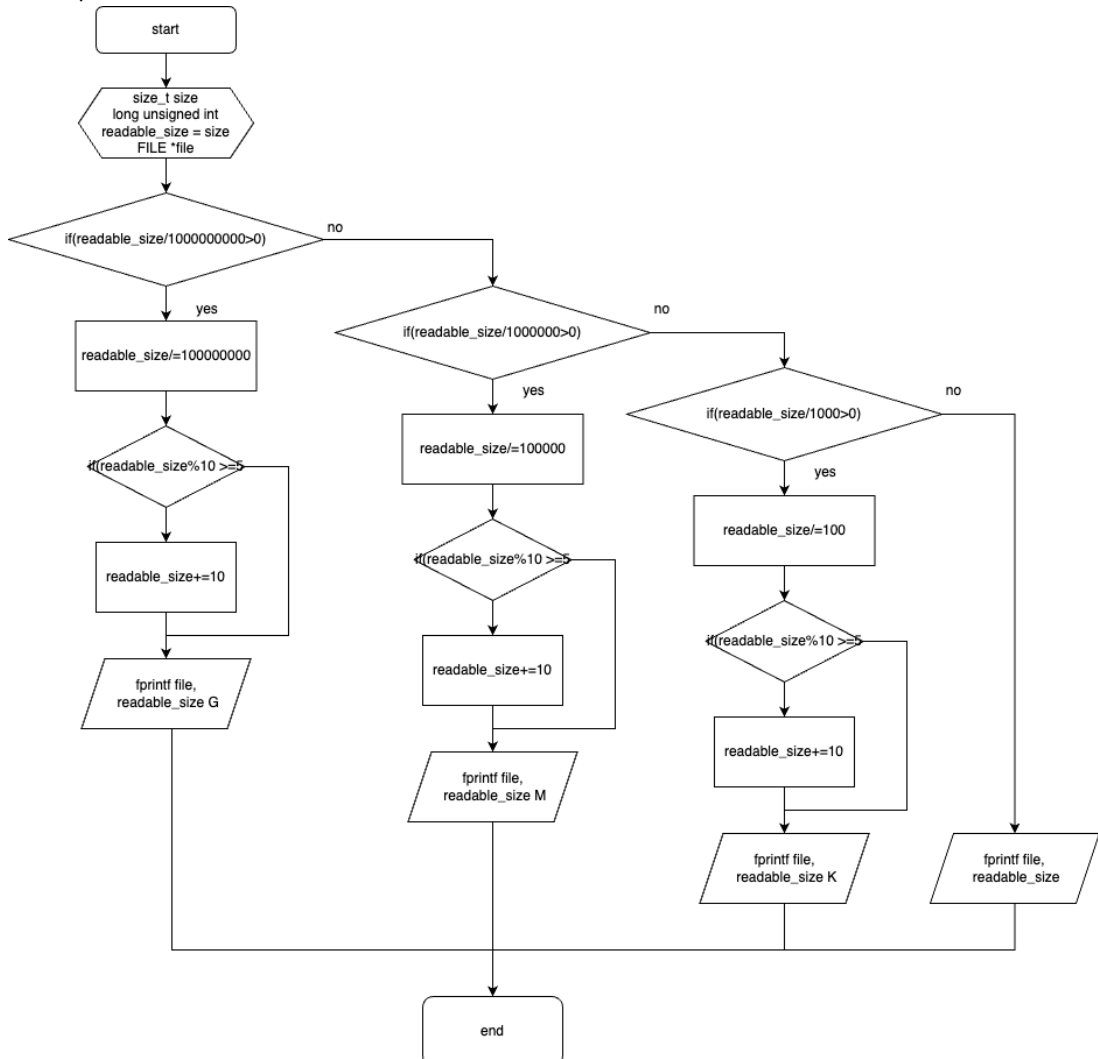
파일의 UID 정보를 출력하는 함수이다.

C. print_GID



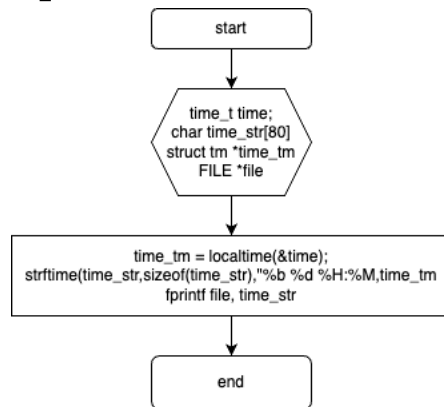
파일의 GID 정보를 출력하는 함수이다.

D. print_readableSIZE



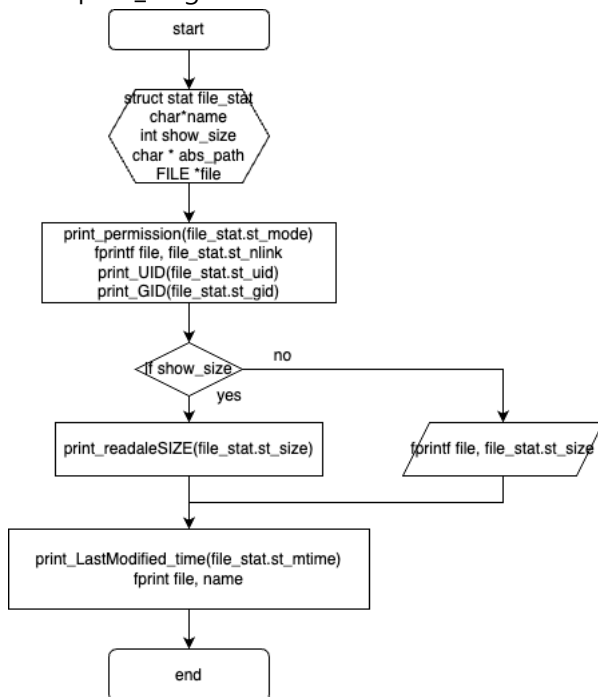
Option -h 를 받은 경우 파일의 사이즈를 단위에 맞춰 출력하는 함수이다.

E. print_Lastmodifiedtime



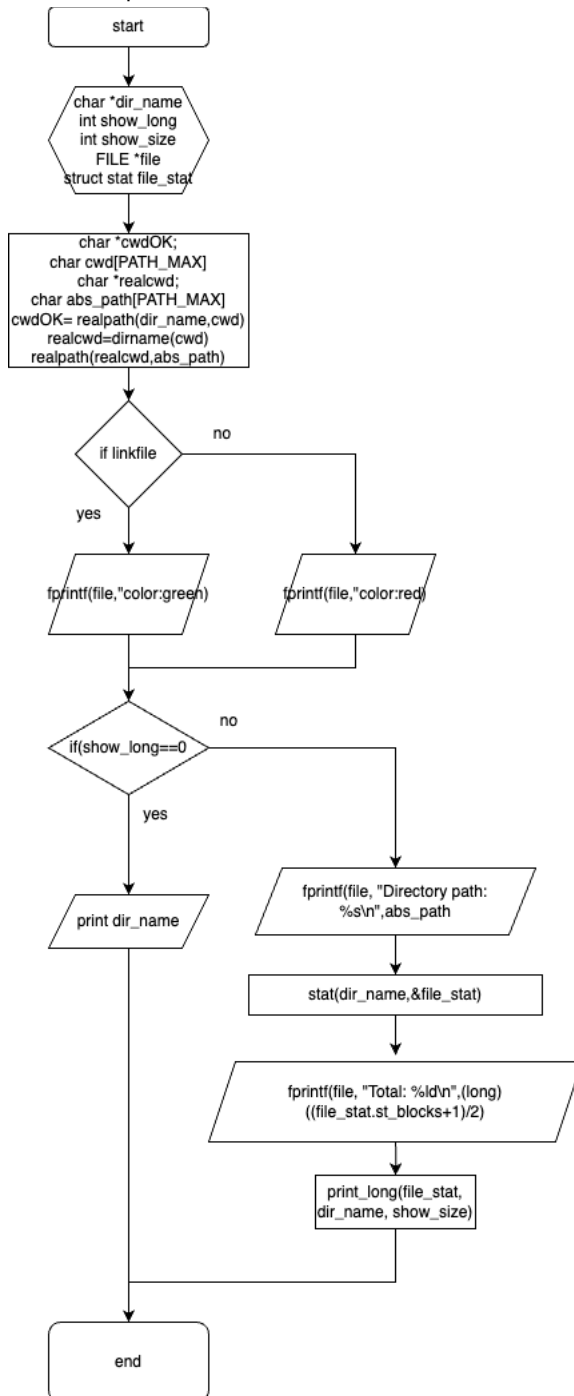
파일의 마지막 수정 일자와 시간을 출력하는 함수이다.

F. print_long



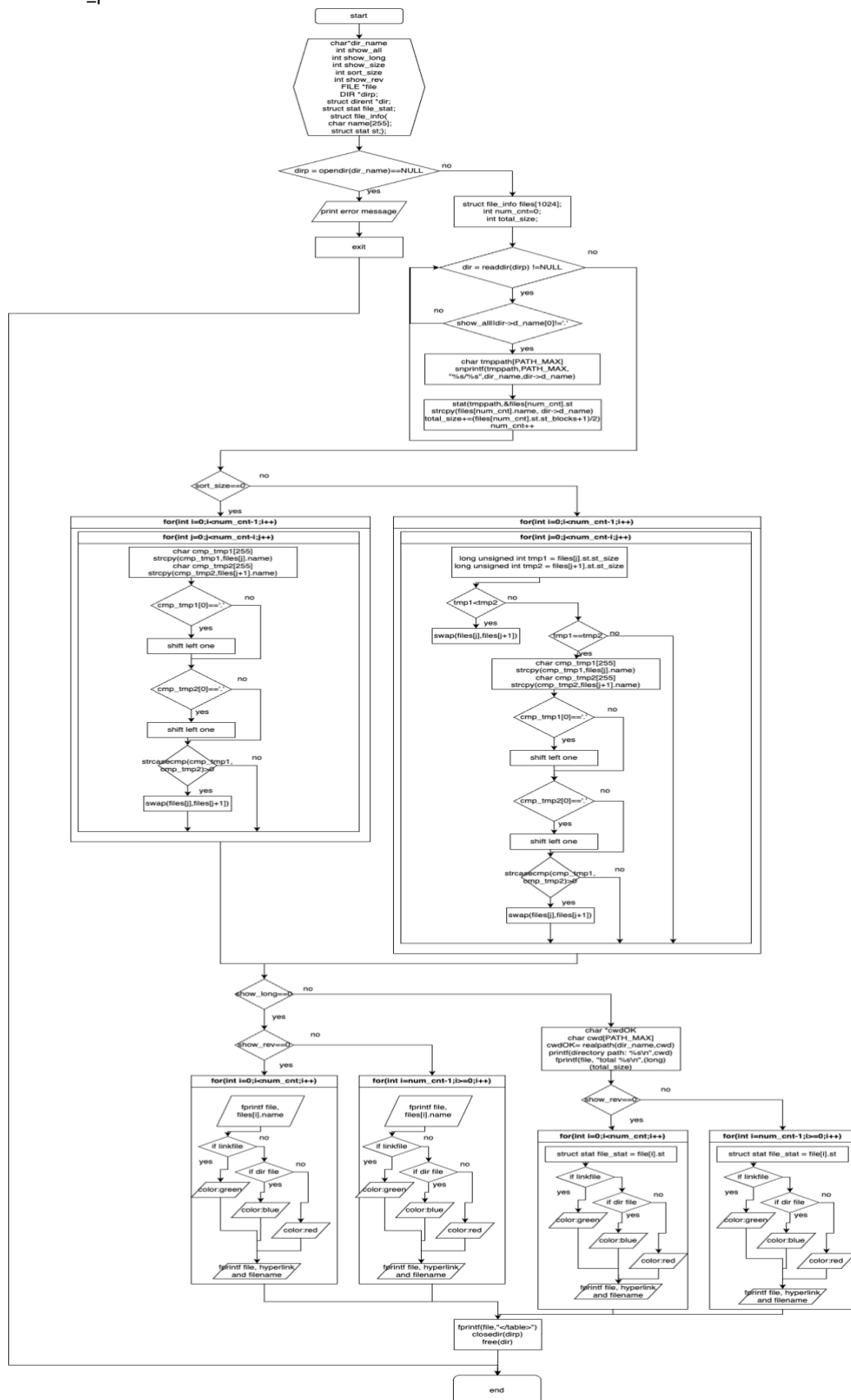
파일의 정보를 long format에 맞춰서 출력하는 함수이다.

G. file_print



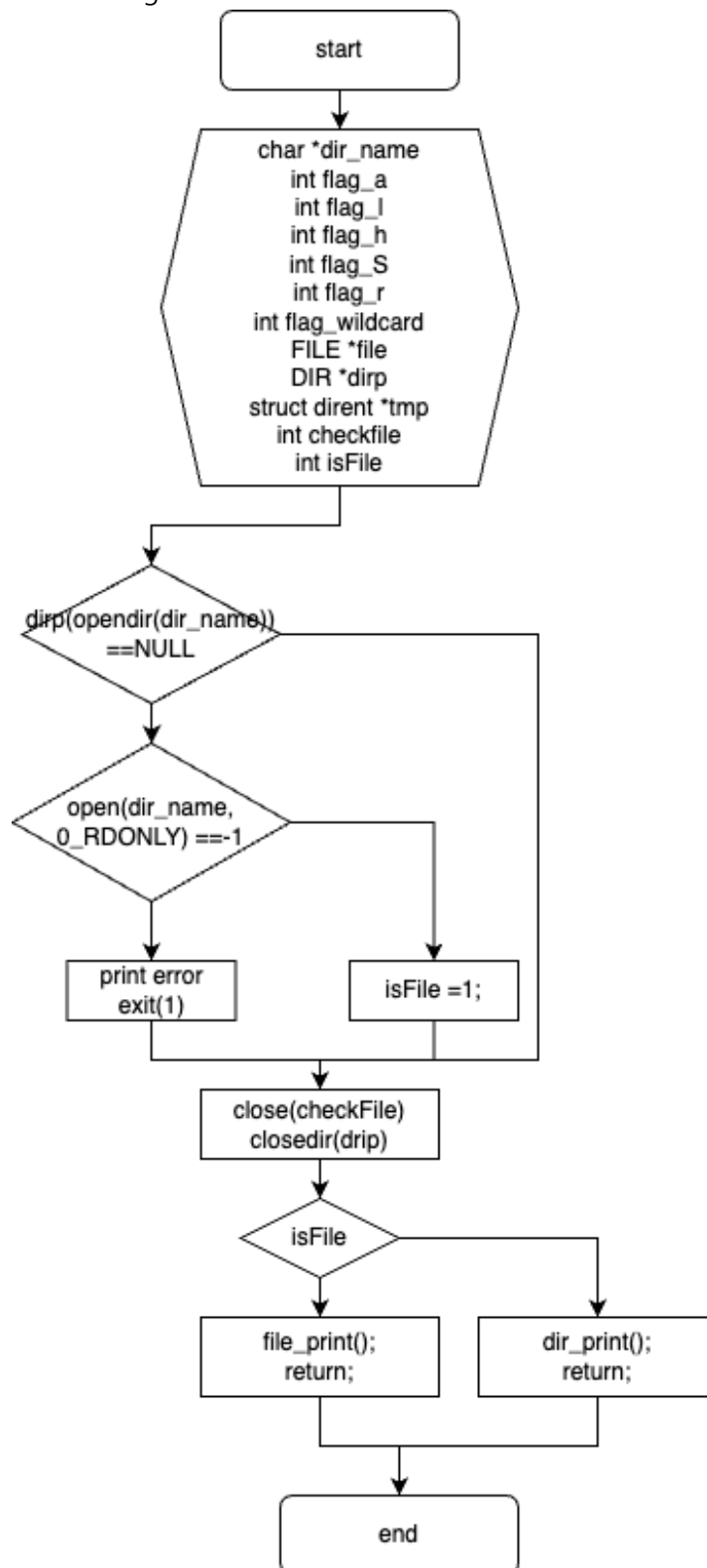
인자로 받은 것이 파일일 경우 진행하는 함수이다. -l과 -h 옵션만 고려한다.

H. dir_print



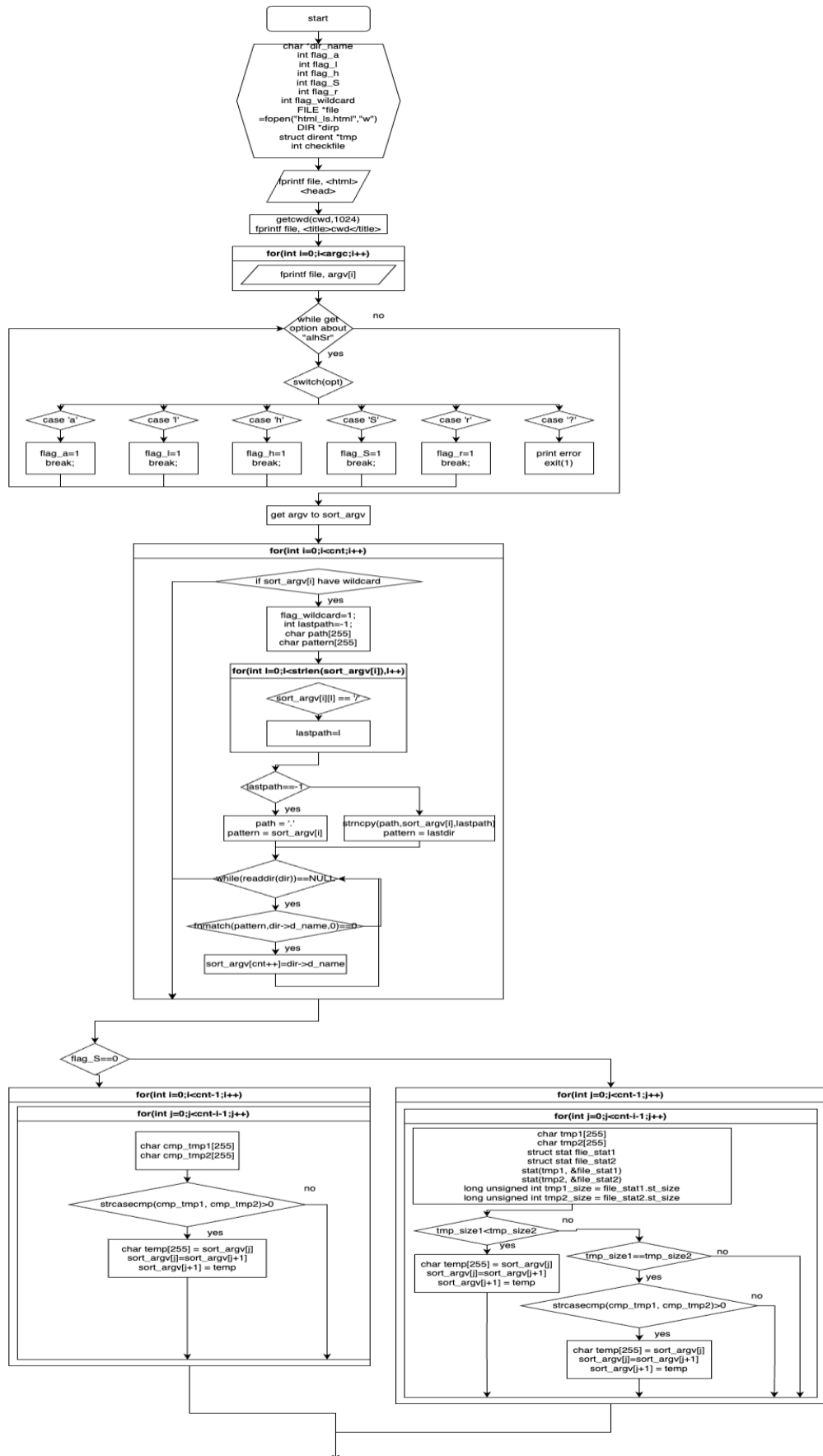
입력받은 인자가 경로일 경우 실행하는 함수이다. 모든 옵션을 고려한다.

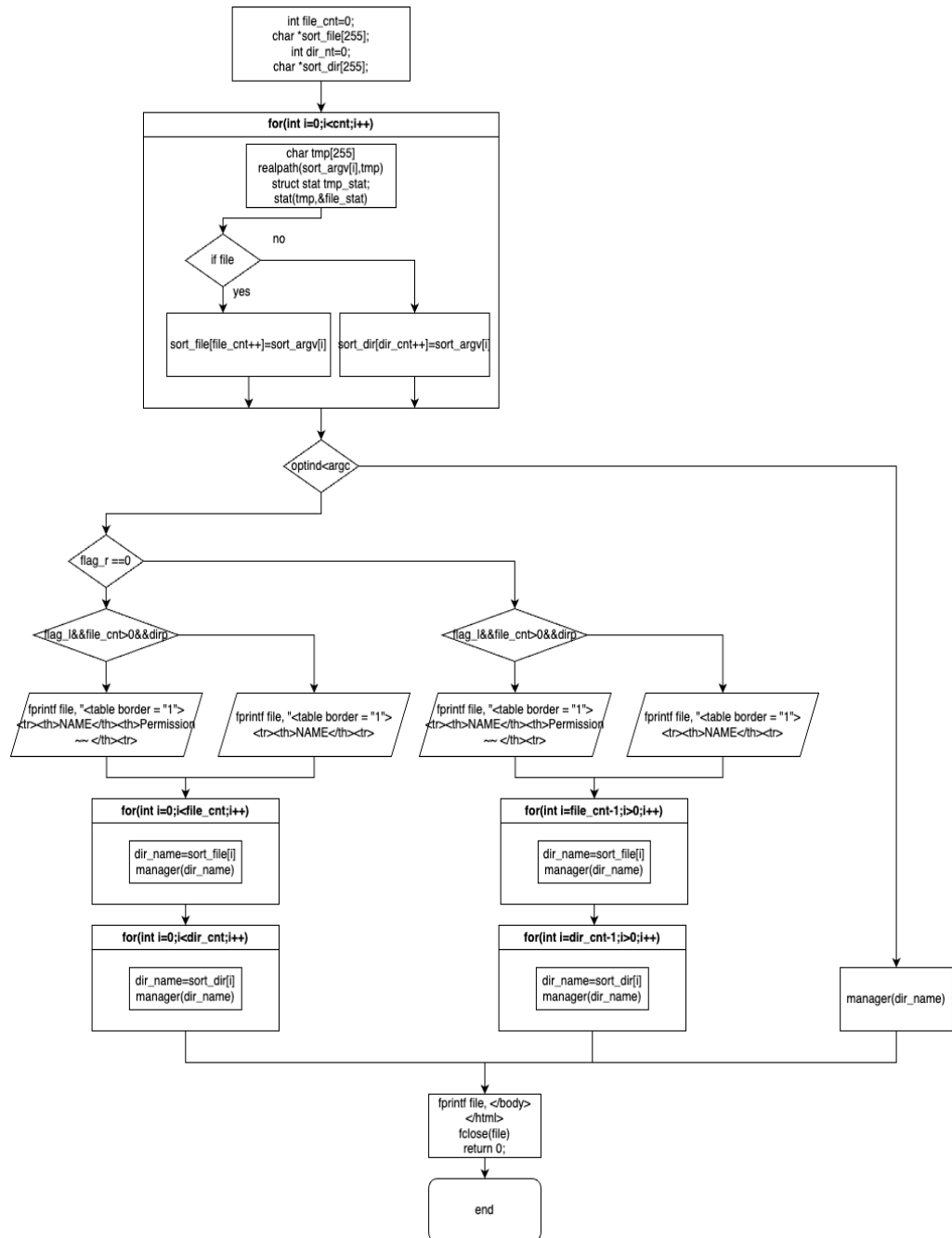
I. manager



옵션 및 인자에 따라 이동할 함수를 정해주는 매니저 역할을 하는 함수이다. wildcard와 아닌 것을 구분하여 보낸다.

J. main





main함수로 option 값에 따라 각 해당값을 set하고 optind<argc면 인자를 전달, 아니라면 인자 없이 옵션 또는 옵션 없이 list_dir()이란 매니저 함수로 간다. 기존에는 와일드카드 함수를 만들어서 manager에서 처리했으나 이번 과제에서는 와일드카드 처리를 메인에서 모두 하였다. 그리고 옵션에 따라 argv 의 위치도 조정한다.

3. Pseudo code

A. print_permission

```
fprint file, <td>
if file is dir, print "d" else "-"
if have read permission by user, fprint file, "r" else "-"
if have write permission by user, fprint file, "w" else "-"
if have execute permission by user, fprint file, "x" else "-"
if have read permission by group, fprint file, "r" else "-"
if have write permission by group, fprint file, "w" else "-"
if have execute permission by group, fprint file, "x" else "-"
if have read permission by other, fprint file, "r" else "-"
if have write permission by other, fprint file, "w" else "-"
if have execute permission by other, fprint file, "x" else "-"
```

B. print_UID

```
get struct passwd pointer pw
if pw=getpwuid(uid) is not NULL
    fprint file, pw->pw_name
else
    fprint file, uid
```

C. print_GID

```
get struct group pointer gr
if gr=getgrgid(gid) is not NULL
    fprint file, pw -> pw_name
else
    fprint file, gid
```

D. print_readableSIZE

```
get size number
if can divide by 10^9
    convert GIGA
else if can divide by 10^6
    convert MEGA
else if can divide by 10^3
    convert KILO
else
    just fprint file, size
```

E. print_LastModified_time

```
make array 80
get struct tm pointer time_tm
time_tm is localtime(&time)
    run strftime to get date and time format
    fprintf file, time
```

F. print_long

```
use print_permission function
fprintf file, file_stat.st_nlink
use print_UID function
use print_GID function
if -h option activated
    use print_readableSIZE function
else
    just fprintf file, file_stat.st_size
use print_LastModified_time function
fprintf file, file name
```

G. file_print

```
get struct stat file_stat

if S_ISLNK(file_stat.st_mode)
    color:green
else
    color:red

if -l option has deactivated
    just print dir_name
else
    get absolute path by realpath function
    print absolute path
    get stat about file
    print total blocks
    print long format using by print_long
```

H. dir_print

```
DIR pointer dirp
get struct dirent pointer dir
get struct stat file_stat
make struct file_info with name and stat

opendir(dir_name)
get struct file_info for 1024 array
while readdir(dirp) is not NULL
    if -a option activated or hidden file
        tmppath is dir_name/dir->d_name
        get tmppath's stat
        get name by dir->d_name
        sum total size about file
        next content

if -S option has not activated
    bubble sort except for first '.' ascending order by name
else
    bubble sort descending order by size
    if same size file
        bubble sort except for first '.' descending order by name
if -l option has not activated
    if -r option has not activated
        if S_ISLNK(file_stat.st_mode)
            color:green
        else if S_ISDIR(file_stat.st_mode)
            color:blue
        else
            color:red

        just fprint file, name step by step
    else
        if S_ISLNK(file_stat.st_mode)
            color:green
        else if S_ISDIR(file_stat.st_mode)
            color:blue
        else
            color:red
```

```

        fprintf file, name reverse
else
    get absolute_path by realpath function
    fprintf file, directory and total size of blocks
    if -r option has not activated
        if S_ISLNK(file_stat.st_mode)
            color:green
        else if S_ISDIR(file_stat.st_mode)
            color:blue
        else
            color:red
        just print name step by step
    else
        if S_ISLNK(file_stat.st_mode)
            color:green
        else if S_ISDIR(file_stat.st_mode)
            color:blue
        else
            color:red

        print name reverse
closedir about dirp

```

I. manager

```

DIR pointer dirp
get struct dirent pointer tmp
inspect dir_name if have wildcard like '*' , '[' , ']' or '?'

if opendir is NULL
    if open file is -1
        fprintf file, error
    else
        use file_print function to print
else
    if another argument have wildcard
        fprintf file, absolute path about file
    use dir_print function to print

```

J. main

```
getoption with argc and argv and option is "alhSr" until is not -1
    if option a
        set show_all and break
    if option l
        set show_long and break
    if option h
        set show_size and break
    if option S
        set sort_size and break
    if option r
        set show_rev and break
get value argv to sort_argv
cnt is argc
find argv about wildcard '*', '?', '['
    if find wildcard set isWildcard
    open directory of path
    while (readdir(dirp)) is not NULL
    if fnmatch(pattern, dir->d_name,0)==-1
        sort_argv[cnt++] is dir->d_name
if flag_s set
    sort ascending order by name about sort_argv
else
    sort descending order by size of file about sort_argv
    if same size
        sort descending order by name of file about sort_argv

for int i=0 and i<cnt and i++
    if file
        sort_file[file_cnt++] is sort_argv[i]
    else
        sort_dir[dir_cnt++] is sort_argv[i]

if optind is smaller than argc
    if flag_r not set
        if flag_l and file_cnt >0 and dirp
            fprintf file, "<table border = \"1\">
            fprintf file, "<tr> <th>Name</th>
            <th>Permission</th> <th>~~~</tr>
```

```

else
    fprintf file, "<table border = \"1\">
    fprintf file, "<tr><th>Name</th></tr>

for int i=0 and i<file_cnt and i++
    use manager function about sort_file
for int i=0 and i<dir_cnt and i++
    use manager fuction about sort_dir

else
    if flag_l and file_cnt >0 and dirp
        fprintf file, "<table border = \"1\">
        fprintf file, "<tr> <th>Name</th>
        <th>Permission</th> <th>~~~</tr>

    else
        fprintf file, "<table border = \"1\">
        fprintf file, "<tr><th>Name</th></tr>

for int i=file_cnt-1 and i>0 and i++
    use manager function about sort_file
for int i=dir_cnt-1 and i>0 and i++
    use manager fuction about sort_dir

else
    use manager function about dir_name

```

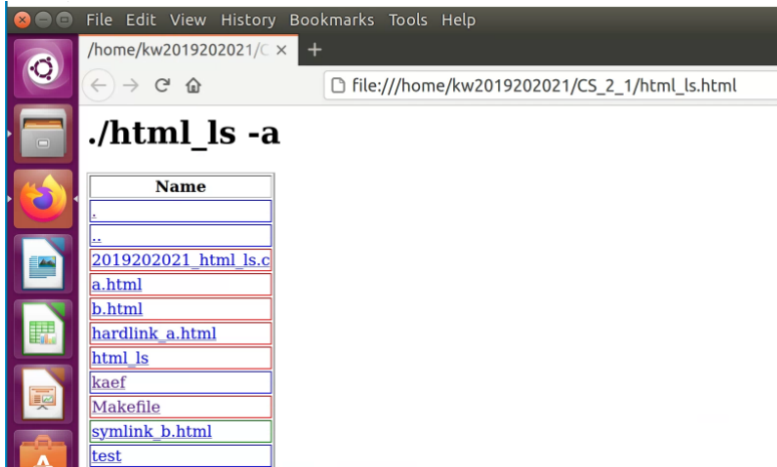
4. 결과화면

A. 기본 출력



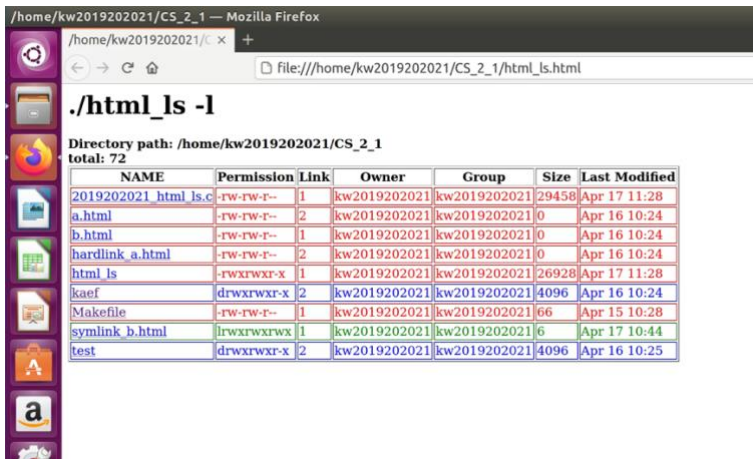
- 옵션 없이 출력했을 때 결과로 숨김 파일 없이 공개된 파일만 출력하였다. 각 파일 별 하이퍼 링크가 연결되어 있다.

B. -a 옵션



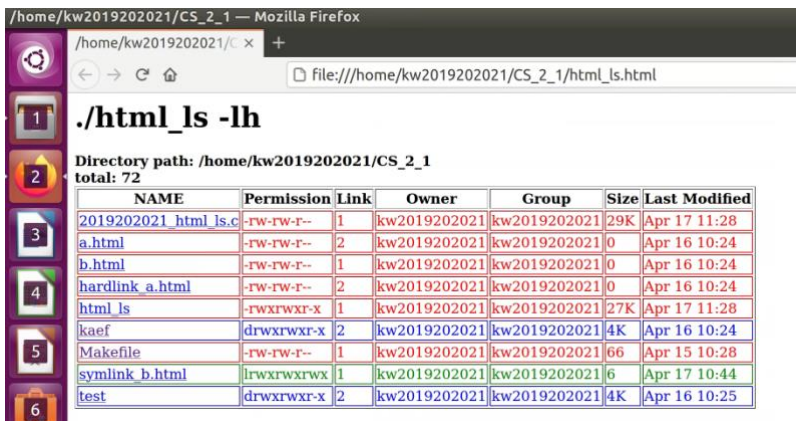
-옵션 -a를 하였을 때 결과로 숨김파일과 함께 출력되는 것을 확인할 수 있다.

C. -l 옵션



-옵션 -l을 하였을 때 결과이다. long format에 맞춰서 각 파일 별로 상세정보를 보여주고 있고 오름차순으로 출력하였다. -l 옵션에서는 각 파일 종류별로 색상을 따로 두어 링크 파일은 녹색, 폴더는 파란색, 나머지는 빨간 색이다. 이때 하드 링크는 나머지로 인식한다.

D. -lh 옵션



-옵션 -lh를 하였을 때 결과로 각 size를 kilo, mega, giga 단위로 출력됨을 확인할 수 있다. 또한 반올림 처리하여 26928 은 27K가 됨을 확인할 수 있다.

E. -ls 옵션

Directory path: /home/kw2019202021/CS_2_1
total: 72

NAME	Permission	Link	Owner	Group	Size	Last Modified
2019202021_html_ls.c	-rw-rw-r--	1	kw2019202021	kw2019202021	29458	Apr 17 11:28
html_ls	-rwxrwxr-x	1	kw2019202021	kw2019202021	26928	Apr 17 11:28
test	drwxrwxr-x	2	kw2019202021	kw2019202021	4096	Apr 16 10:25
kaef	drwxrwxr-x	2	kw2019202021	kw2019202021	4096	Apr 16 10:24
Makefile	-rw-rw-r--	1	kw2019202021	kw2019202021	66	Apr 15 10:28
symlink_b.html	lrwxrwxrwx	1	kw2019202021	kw2019202021	6	Apr 17 10:44
hardlink_a.html	-rw-rw-r--	2	kw2019202021	kw2019202021	0	Apr 16 10:24
b.html	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 16 10:24
a.html	-rw-rw-r--	2	kw2019202021	kw2019202021	0	Apr 16 10:24

-옵션 S를 사용하였을 때 결과로 size 크기로 내림차순 정렬을 하였다. 이 때 사이즈가 같다면 이름을 내림차순으로 정렬하여 출력한다. -l 옵션과 함께 사용하여 사이즈로 내림차순이 올바르게 되었는지 확인하였다.

F. -lr 옵션

Directory path: /home/kw2019202021/CS_2_1
total: 72

NAME	Permission	Link	Owner	Group	Size	Last Modified
test	drwxrwxr-x	2	kw2019202021	kw2019202021	4096	Apr 16 10:25
symlink_b.html	lrwxrwxrwx	1	kw2019202021	kw2019202021	6	Apr 17 10:44
Makefile	-rw-rw-r--	1	kw2019202021	kw2019202021	66	Apr 15 10:28
kaef	drwxrwxr-x	2	kw2019202021	kw2019202021	4096	Apr 16 10:24
html_ls	-rwxrwxr-x	1	kw2019202021	kw2019202021	26928	Apr 17 11:28
hardlink_a.html	-rw-rw-r--	2	kw2019202021	kw2019202021	0	Apr 16 10:24
b.html	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 16 10:24
a.html	-rw-rw-r--	2	kw2019202021	kw2019202021	0	Apr 16 10:24
2019202021_html_ls.c	-rw-rw-r--	1	kw2019202021	kw2019202021	29458	Apr 17 11:28

-옵션 r을 사용하여 역순 출력을 진행하였다. 이 때 아스키코드를 기준으로 하며 오름차순의 역순인 내림차순으로 출력됨을 확인할 수 있다.

G. -al 옵션

Directory path: /home/kw2019202021/CS_2_1
total: 80

NAME	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	4	kw2019202021	kw2019202021	4096	Apr 17 11:28
..	drwxr-xr-x	25	kw2019202021	kw2019202021	4096	Apr 15 10:26
2019202021_html_ls.c	-rw-rw-r--	1	kw2019202021	kw2019202021	29458	Apr 17 11:28
a.html	-rw-rw-r--	2	kw2019202021	kw2019202021	0	Apr 16 10:24
b.html	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 16 10:24
hardlink_a.html	-rw-rw-r--	2	kw2019202021	kw2019202021	0	Apr 16 10:24
html_ls	-rwxrwxr-x	1	kw2019202021	kw2019202021	26928	Apr 17 11:28
kaef	drwxrwxr-x	2	kw2019202021	kw2019202021	4096	Apr 16 10:24
Makefile	-rw-rw-r--	1	kw2019202021	kw2019202021	66	Apr 15 10:28
symlink_b.html	lrwxrwxrwx	1	kw2019202021	kw2019202021	6	Apr 17 10:44
test	drwxrwxr-x	2	kw2019202021	kw2019202021	4096	Apr 16 10:25

-a와 l을 함께 두었을 때 모든 파일이 long format에 맞춰 상세하게 출력하고 있다.

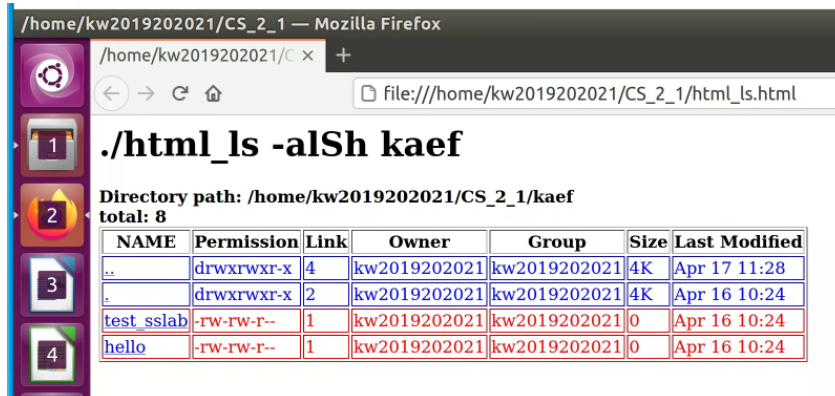
H. -alh 옵션



NAME	Permission	Link	Owner	Group	Size	Last Modified
.	drwxrwxr-x	2	kw2019202021	kw2019202021	4K	Apr 16 10:24
..	drwxrwxr-x	4	kw2019202021	kw2019202021	4K	Apr 17 11:28
hello	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 16 10:24
test_sslab	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 16 10:24

-alh 옵션을 같이 썼을 때 결과로 모든 파일을 Long format에 맞춰 상세하게 출력하되 사이즈는 readable size 단위로 맞춰 출력하도록 하였다.

I. -alSh 옵션



NAME	Permission	Link	Owner	Group	Size	Last Modified
..	drwxrwxr-x	4	kw2019202021	kw2019202021	4K	Apr 17 11:28
.	drwxrwxr-x	2	kw2019202021	kw2019202021	4K	Apr 16 10:24
test_sslab	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 16 10:24
hello	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 16 10:24

-H번 옵션에서 -S 옵션을 추가하여 사이즈별로 내림차순이 됨을 확인하고 같은 사이즈는 이름 순으로 내림차순 됨을 확인하였다.

J. -alhr 옵션



NAME	Permission	Link	Owner	Group	Size	Last Modified
test_sslab	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 16 10:24
hello	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 16 10:24
.	drwxrwxr-x	4	kw2019202021	kw2019202021	4K	Apr 17 11:28
..	drwxrwxr-x	2	kw2019202021	kw2019202021	4K	Apr 16 10:24

-H번 옵션에서 -r 옵션을 추가하여 이름으로 오름차순 정렬하였던 것이 내림차순 정렬로 변경된 것을 확인할 수 있다.

K. -alShr



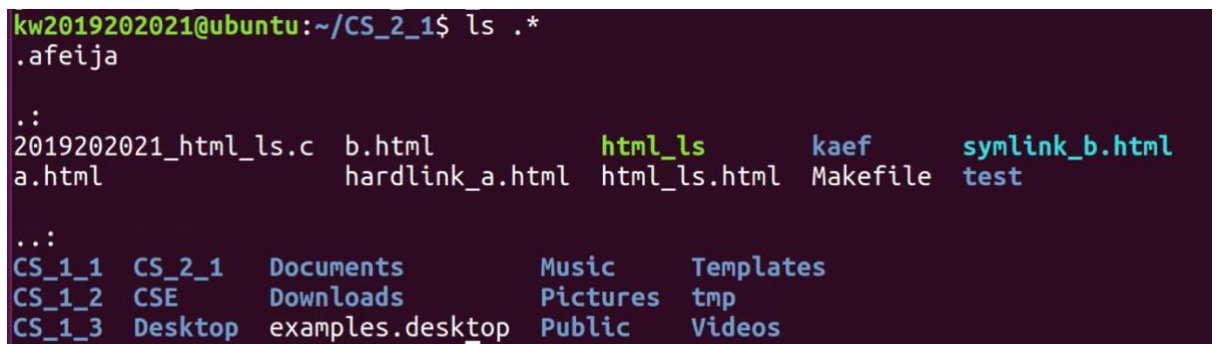
-l 옵션에서 -r 옵션을 추가한 것으로 사이즈별로 내림차순 되어있던 것이 오름차순 정렬로 변경되었으며 같은 사이즈는 오름차순 정렬이 되어있다.

L. wild card '*'



-'*' 와일드 카드를 사용했을 때 결과로 모든 파일을 출력하되 디렉토리가 포함된다면 해당 디렉토리의 경로 출력과 함께 하위 파일도 출력한다.

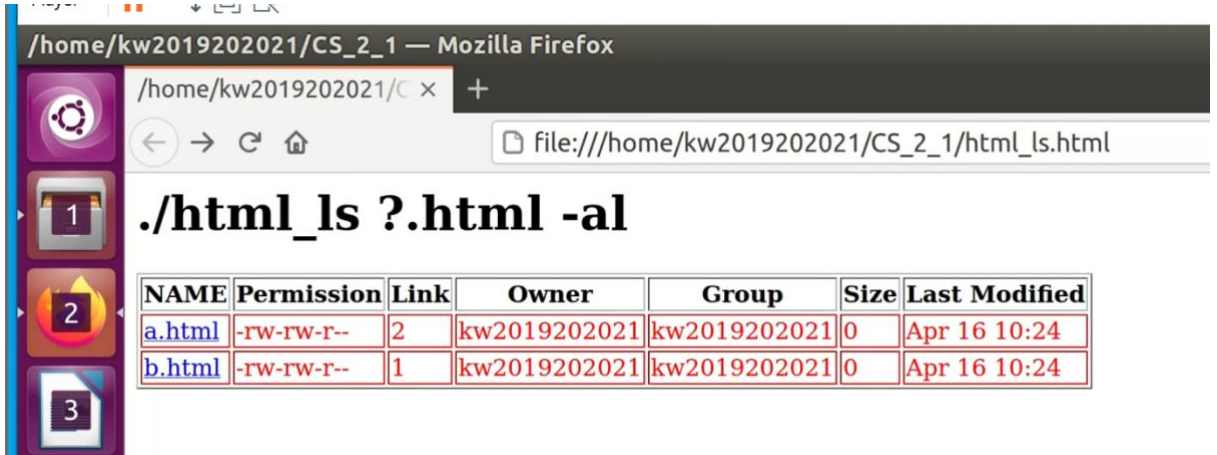
M. wild card '.*'





-'.' 와일드 카드를 사용하였을 때 숨김파일에 대해서만 파일과 폴더의 하위 파일까지 출력함을 확인하였다. 이 때 결과는 ls .*의 결과와 비교하여 확인하였다. 특이점으로 .과 ..도 인식하여 현재 디렉토리와 상위 디렉토리의 내부 파일도 출력하는 것을 확인하였다.

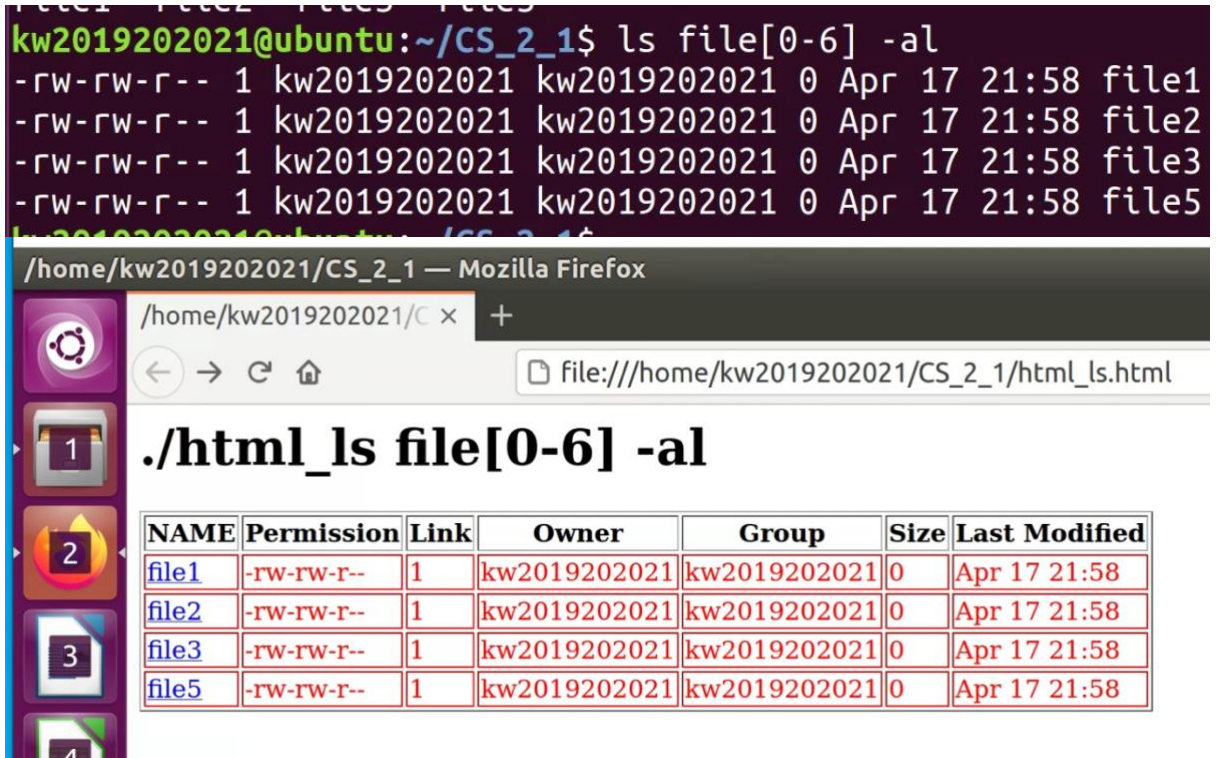
N. wild card '?'



NAME	Permission	Link	Owner	Group	Size	Last Modified
a.html	-rw-rw-r--	2	kw2019202021	kw2019202021	0	Apr 16 10:24
b.html	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 16 10:24

-.html 을 인자로 두어 .html 과 함께 한글자 이름을 가진 파일을 모두 출력하도록 하였다.

O. wild card '['



```
kw2019202021@ubuntu:~/CS_2_1$ ls file[0-6] -al
-rw-rw-r-- 1 kw2019202021 kw2019202021 0 Apr 17 21:58 file1
-rw-rw-r-- 1 kw2019202021 kw2019202021 0 Apr 17 21:58 file2
-rw-rw-r-- 1 kw2019202021 kw2019202021 0 Apr 17 21:58 file3
-rw-rw-r-- 1 kw2019202021 kw2019202021 0 Apr 17 21:58 file5
```

NAME	Permission	Link	Owner	Group	Size	Last Modified
file1	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 17 21:58
file2	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 17 21:58
file3	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 17 21:58
file5	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 17 21:58

-file[0-6]인자로 두었을 때 file이란 패턴과 0부터 6까지의 파일을 읽어와 출력하며 없다면 무시한다.

P. 폴더 '*'

/home/kw2019202021/CS_2_1 — Mozilla Firefox

/home/kw2019202021/C x /home/kw2019202021/C x +

file:///home/kw2019202021/CS_2_1/html_ls.htm

./html_ls /home/kw2019202021 * -l

NAME	Permission	Link	Owner	Group	Size	Last Modified
2019202021_html_ls.c	-rw-rw-r--	1	kw2019202021	kw2019202021	36159	Apr 18 06:12
a.html	-rw-rw-r--	2	kw2019202021	kw2019202021	4	Apr 18 05:09
b.html	-rw-rw-r--	1	kw2019202021	kw2019202021	6	Apr 18 05:09
file1	-rw-rw-r--	1	kw2019202021	kw2019202021	15	Apr 18 05:09
file2	-rw-rw-r--	1	kw2019202021	kw2019202021	9	Apr 18 05:09
file3	-rw-rw-r--	1	kw2019202021	kw2019202021	2	Apr 18 05:09
file5	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 17 21:58
file6	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 18 05:40
hardlink_a.html	-rw-rw-r--	2	kw2019202021	kw2019202021	4	Apr 18 05:09
html_ls	-rwxrwxr-x	1	kw2019202021	kw2019202021	31128	Apr 18 06:12
Makefile	-rw-rw-r--	1	kw2019202021	kw2019202021	66	Apr 15 10:28
symlink_b.html	lrwxrwxrwx	1	kw2019202021	kw2019202021	6	Apr 17 10:44

Directory path: /home/kw2019202021/CS_2_1/kaef
total: 0

NAME	Permission	Link	Owner	Group	Size	Last Modified
hello	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 16 10:24
test_sslab	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 16 10:24

Directory path: /home/kw2019202021
total: 68

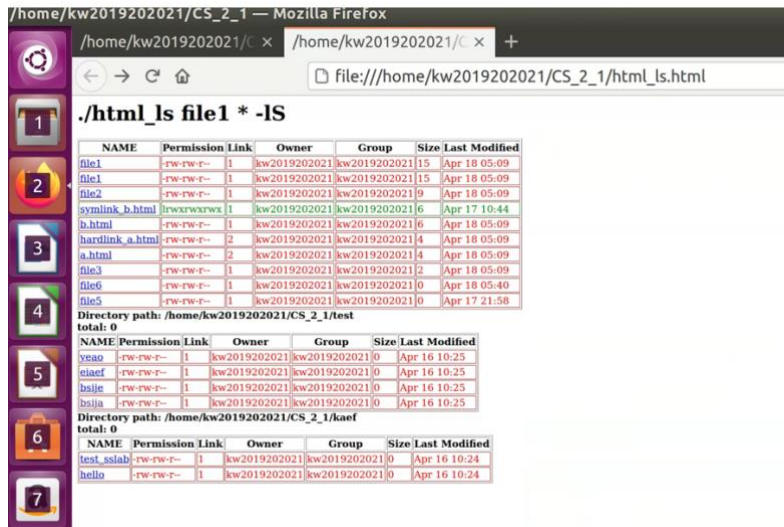
NAME	Permission	Link	Owner	Group	Size	Last Modified
CS_1_1	drwxrwxr-x	5	kw2019202021	kw2019202021	4096	Apr 16 00:58
CS_1_2	drwxrwxr-x	5	kw2019202021	kw2019202021	4096	Apr 11 09:52
CS_1_3	drwxrwxr-x	6	kw2019202021	kw2019202021	4096	Apr 14 10:38
CS_2_1	drwxrwxr-x	4	kw2019202021	kw2019202021	4096	Apr 18 06:12
CSE	drwxrwxr-x	4	kw2019202021	kw2019202021	4096	Apr 14 02:23
Desktop	drwxr-xr-x	2	kw2019202021	kw2019202021	4096	Mar 09 20:44
Documents	drwxr-xr-x	2	kw2019202021	kw2019202021	4096	Mar 09 20:44
Downloads	drwxr-xr-x	3	kw2019202021	kw2019202021	4096	Apr 11 14:49
examples.desktop	-rw-r--r--	1	kw2019202021	kw2019202021	8980	Mar 09 20:38
Music	drwxr-xr-x	2	kw2019202021	kw2019202021	4096	Mar 09 20:44
Pictures	drwxr-xr-x	2	kw2019202021	kw2019202021	4096	Mar 09 20:44
Public	drwxr-xr-x	2	kw2019202021	kw2019202021	4096	Mar 09 20:44
Templates	drwxr-xr-x	2	kw2019202021	kw2019202021	4096	Mar 09 20:44
tmp	drwxrwxr-x	2	kw2019202021	kw2019202021	4096	Mar 22 04:50
Videos	drwxr-xr-x	2	kw2019202021	kw2019202021	4096	Mar 09 20:44

Directory path: /home/kw2019202021/CS_2_1/test
total: 0

NAME	Permission	Link	Owner	Group	Size	Last Modified
bsija	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 16 10:25
bsije	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 16 10:25
eiaef	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 16 10:25
yeao	-rw-rw-r--	1	kw2019202021	kw2019202021	0	Apr 16 10:25

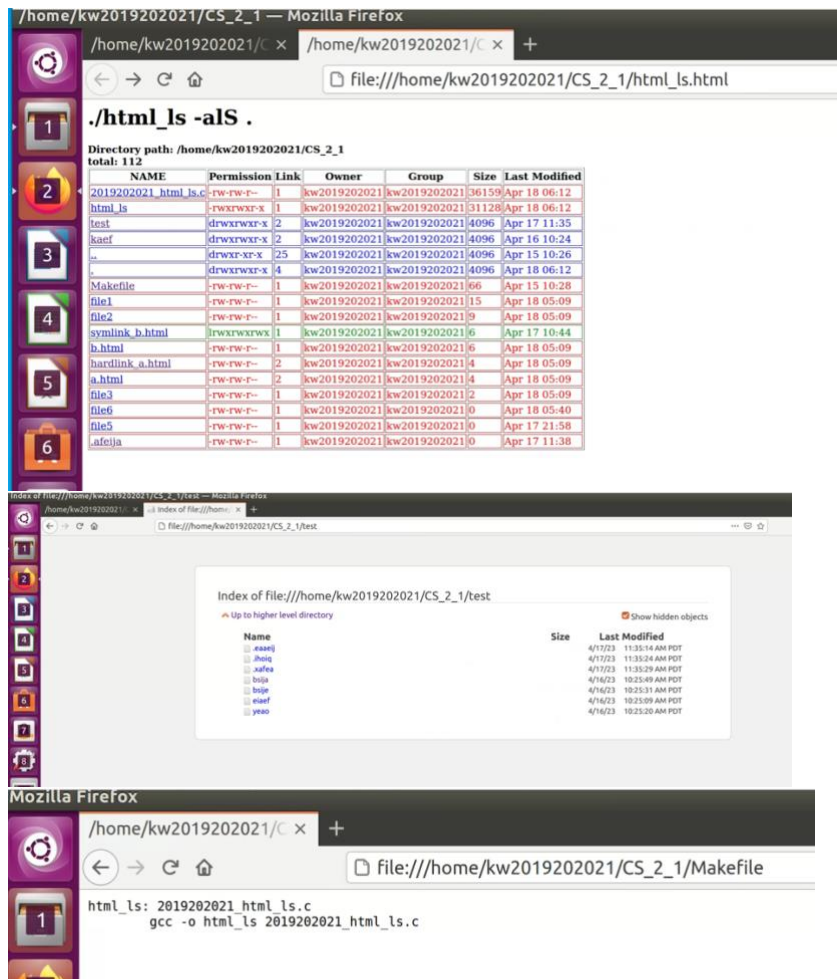
-인자로 ~와 '*'을 두었을 때 결과로 원래 옵션과 와일드카드를 동시에 사용하는 경우는 없지만 코드를 변경하여 -i 옵션과 같이 사용할 수 있도록 하여 상세 내용을 다시 확인하였다. 파일별로 먼저 출력하고 나머지 디렉토리를 차례로 출력함을 확인할 수 있다.

Q. 파일 '*'



file1과 '*'를 인자로 두었을 때 '*'만 인자로 두었을 때 결과에서 file1이 하나 더 추가되었음을 확인할 수 있다.

R. 하이퍼링크 작동



각 파일은 하이퍼링크로 연결되어있으며 디렉토리를 선택하면 디렉토리를 검색할 수 있고 일반 파일을 선택하면 다운로드 혹은 읽는다.

5. 고찰

Assignment 2-1은 Assignment 1-3을 html 파일로 출력하되 table로 결과를 나열하도록 하는 과제로 지난 Assignment 1-3의 과제를 진행할 때 확인하지 못한 예외사항이 있어 다시 처음부터 수정해야했고 이 과정에서 와일드카드를 " 없이 사용하면 알아서 해당 파일과 디렉토리의 이름을 가져오는 것을 확인하였다. 그래서 manager에서 와일드카드 유무를 확인했던 것을 main으로 가져와서 새로운 argv에 경로와 함께 저장하였다.

이번 과제에서 흥미로웠던 점은 html 파일을 c언어로 코딩하여 결과를 만들어낼 수 있음이었다. 다만 조금 힘들었던 점은 html 파일을 직접 제작하는 것이 아닌 코드를 fprintf 함수를 통해 하나하나 입력하다 보니 해당 html 파일의 순서가 헷갈려 table을 마무리 못하는 경우도 있었고 table을 여러개 입력하는 경우도 있었다.

그리고 link file을 구분하기 위해 stat을 쓰면 안되고 lstat을 사용해야함을 알고 S_ISLNK 함수를 사용할 때 hardlink는 구분하지 못하고 symbolic link만 구분할 수 있음을 확인하였다.

마지막으로 기존 구현했던 코드를 다시 사용하는 것이다 보니 자신의 코드의 잘못된 점을 다시 확인할 수 있음이 추후 코딩할 때 자신의 문제점을 인지할 수 있음이 좋다고 생각하였다.

6. Reference

- 1) 시스템프로그래밍실습 Assignment 2-1/광운대학교/컴퓨터정보공학부/김태석교수님 /2023
- 2) 시스템프로그래밍 강의자료/광운대학교/ 컴퓨터정보공학부/김태석교수님/2023