시스템프로그래밍

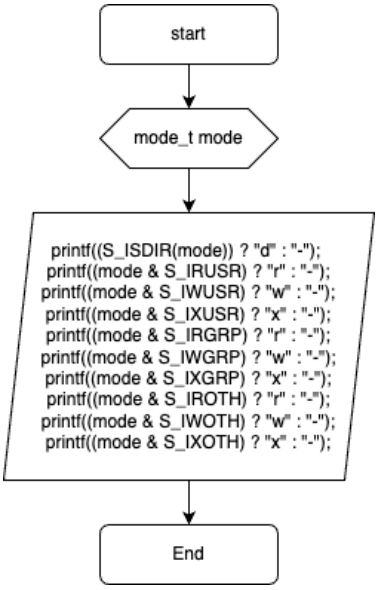
과제이름: Assignment 1-3

- 담당교수:김 태 석 교수님
- 학 과: 컴퓨터정보공학부
- 학 번: 2019202021
- 이 름: 정 성 엽
- 제출일: 2023/4/12

1. Introduction
   A. 과제 소개

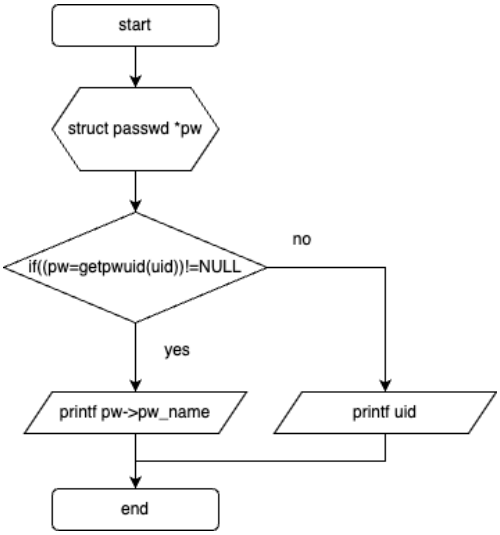
      이번 Assignment 1-3 과제는 이전 Assignment 1-2에서 만들었던 ls를 발전시켜 -h, -S, -r 옵션에 대한 기능을 추가하고, wild card '*', '?', '[seq]'를 입력 받았을 때 해당 패턴을 가졌거나 범위 안에 있는 폴더나 파일을 출력하도록 한다. 이 때 wild card를 통해 디렉토리를 찾을 경우 디렉토리 내용을 출력하고 wildcard는 인자 즉 경로 끝에서만 사용된다. 또한 wildcard는 따음표(' ') 내부에서만 사용하여 구분한다. 마지막으로 wildcard는 옵션과 함께 사용되지 않으며 패턴과 일치하는 것이 없으면 무시한다.

2. FlowChart
   A. print_permission



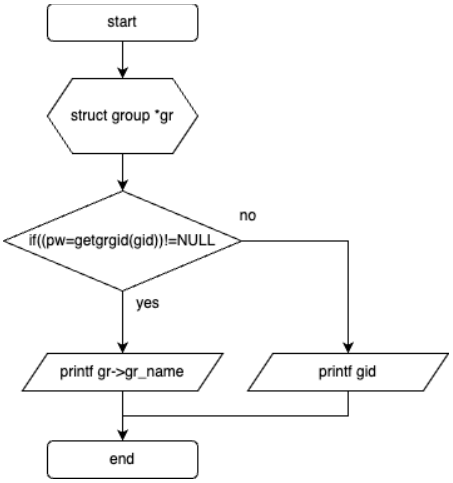
파일의 허용정보를 출력하는 함수이다.

   B. print_UID



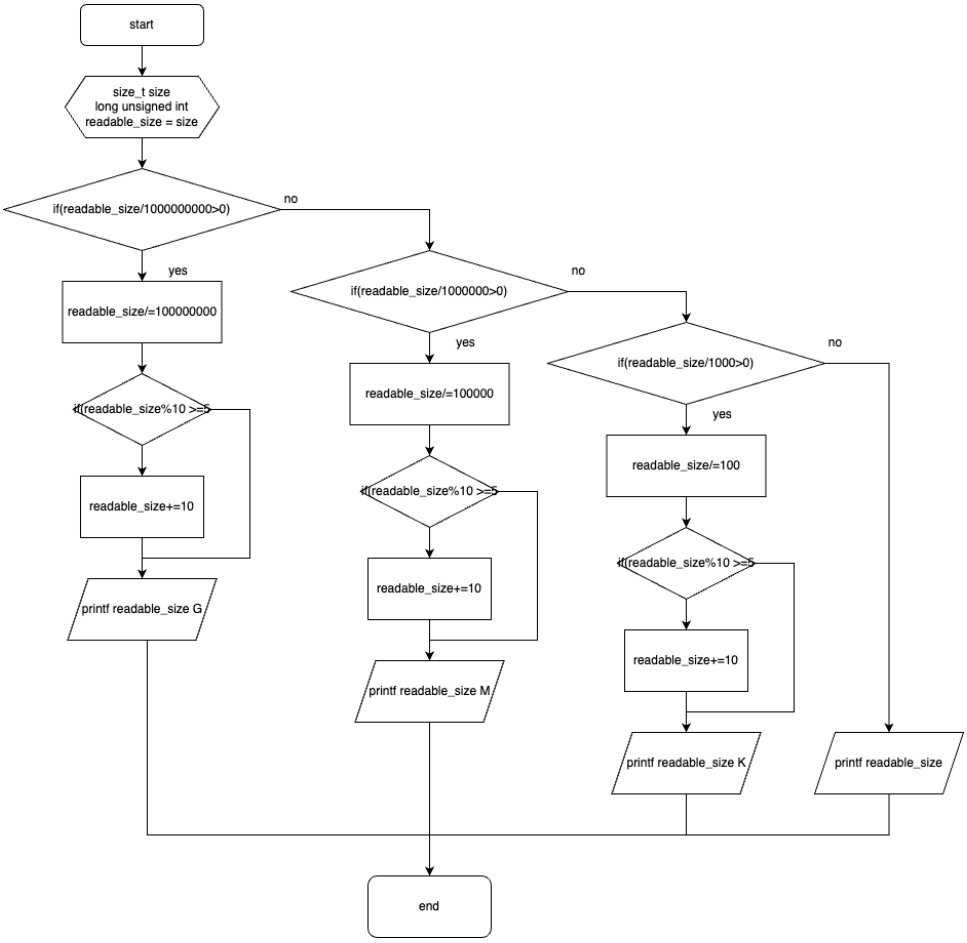
파일의 UID 정보를 출력하는 함수이다.

C. print_GID



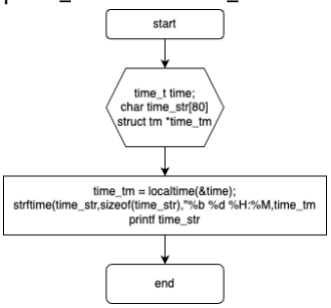
파일의 GID 정보를 출력하는 함수이다.

D. print_readableSIZE



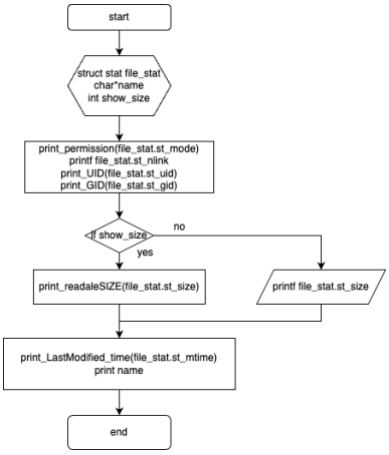
Option -h를 받은 경우 파일의 사이즈를 단위에 맞춰 출력하는 함수이다.

E. print_LastModified_time



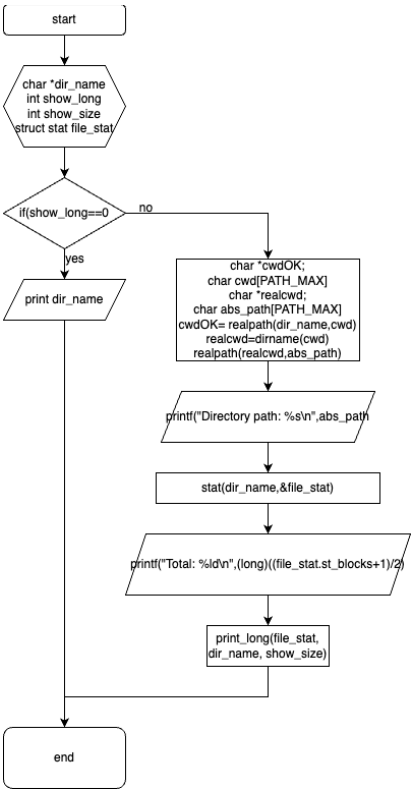
파일의 마지막 수정 일자와 시간을 출력하는 함수이다.

F. print_long



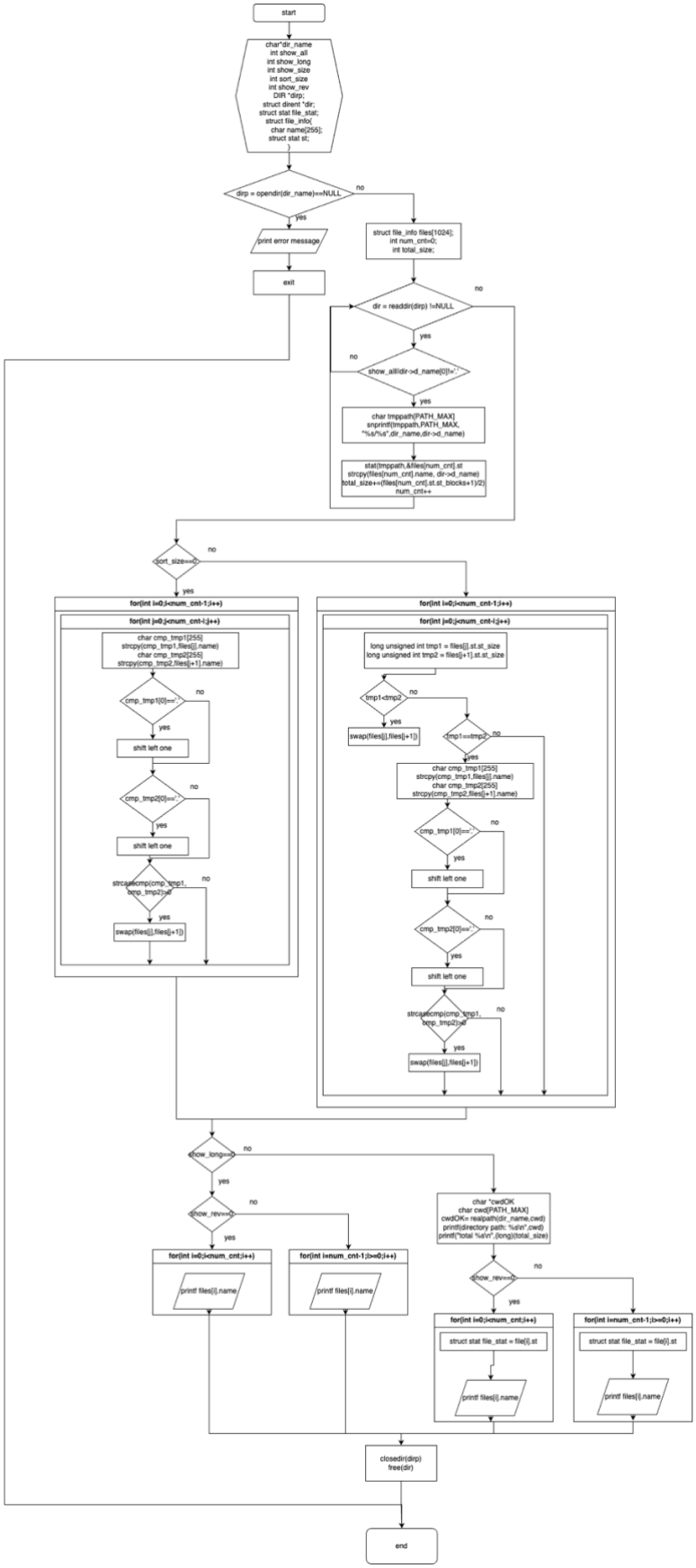
파일의 정보를 long format에 맞춰서 출력하는 함수이다.

G. file_print



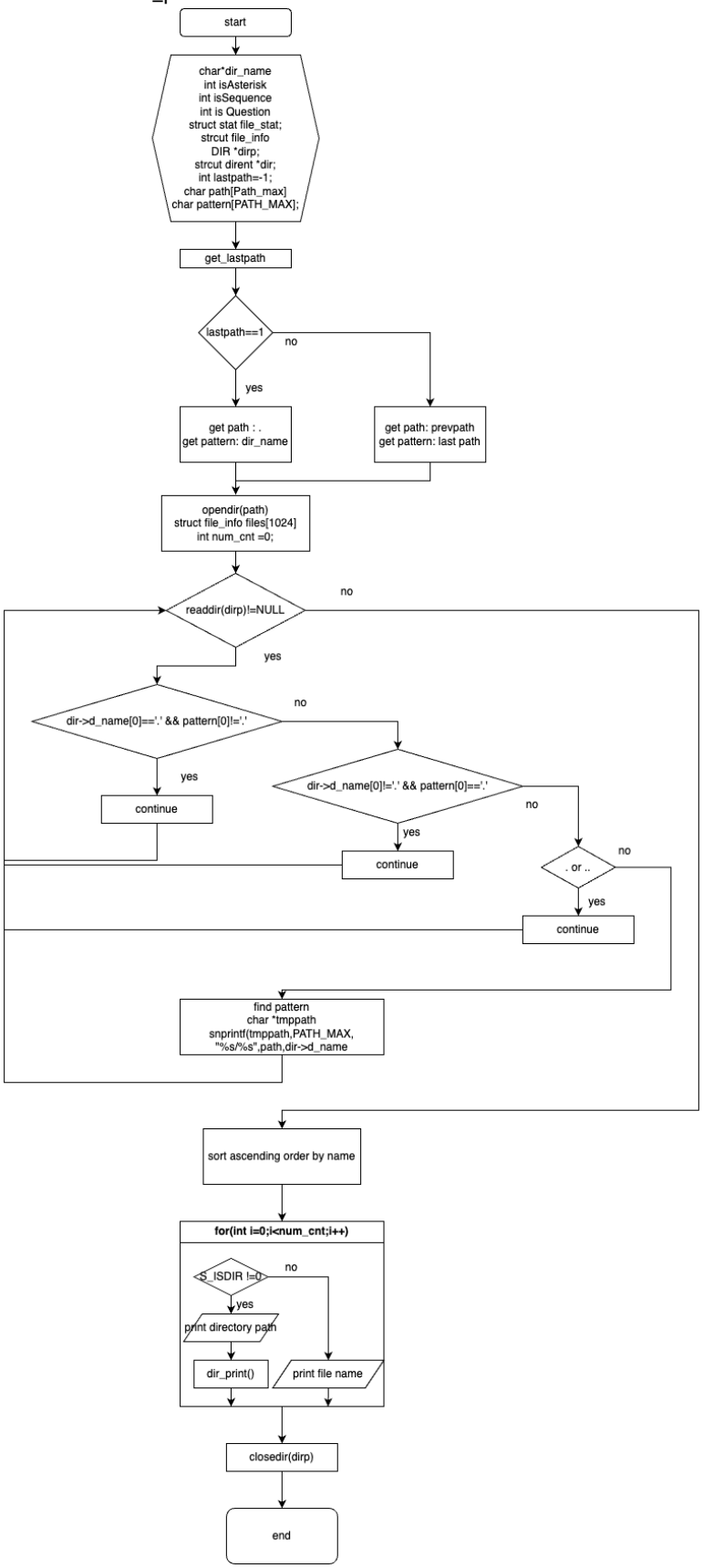
인자로 받은 것이 파일일 경우 진행하는 함수이다. -l과 -h 옵션만 고려한다.

H. dir_print



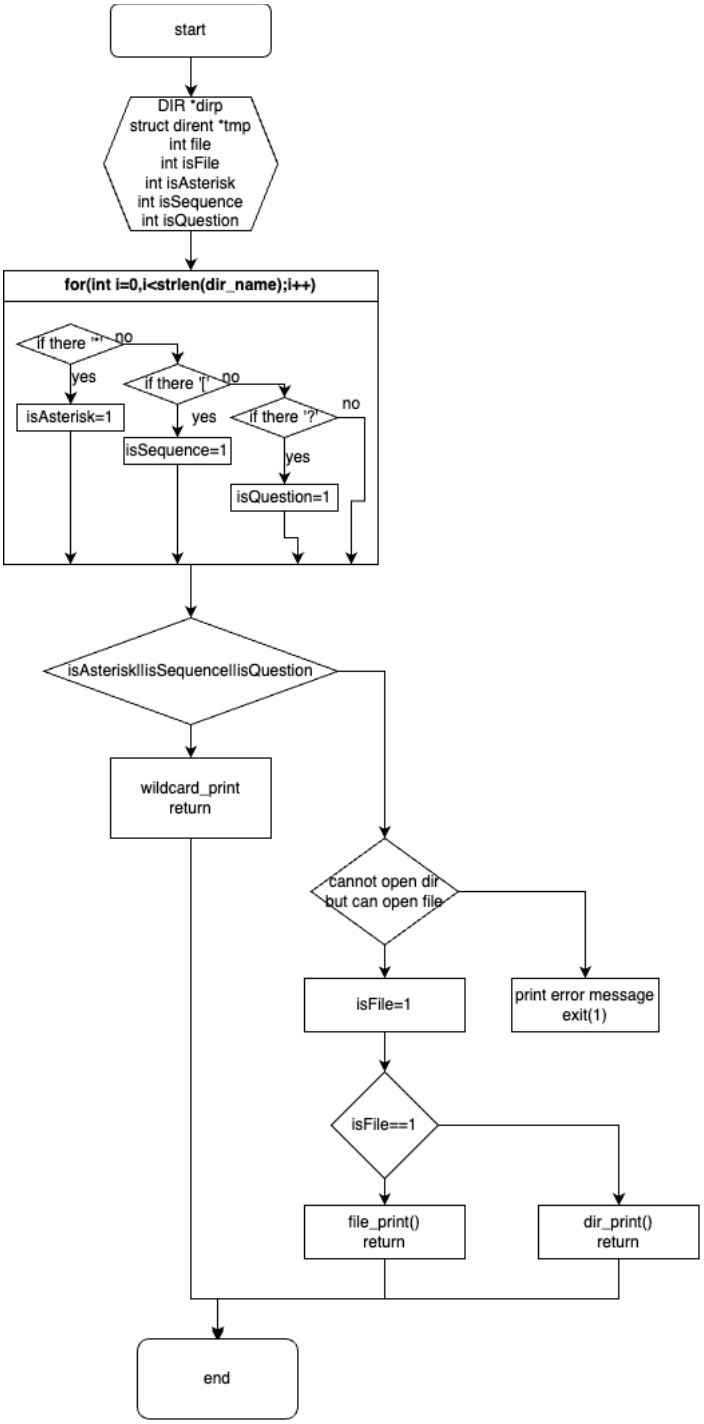
입력받은 인자가 경로일 경우 실행하는 함수이다. 모든 옵션을 고려한다.

I. wildcard_print



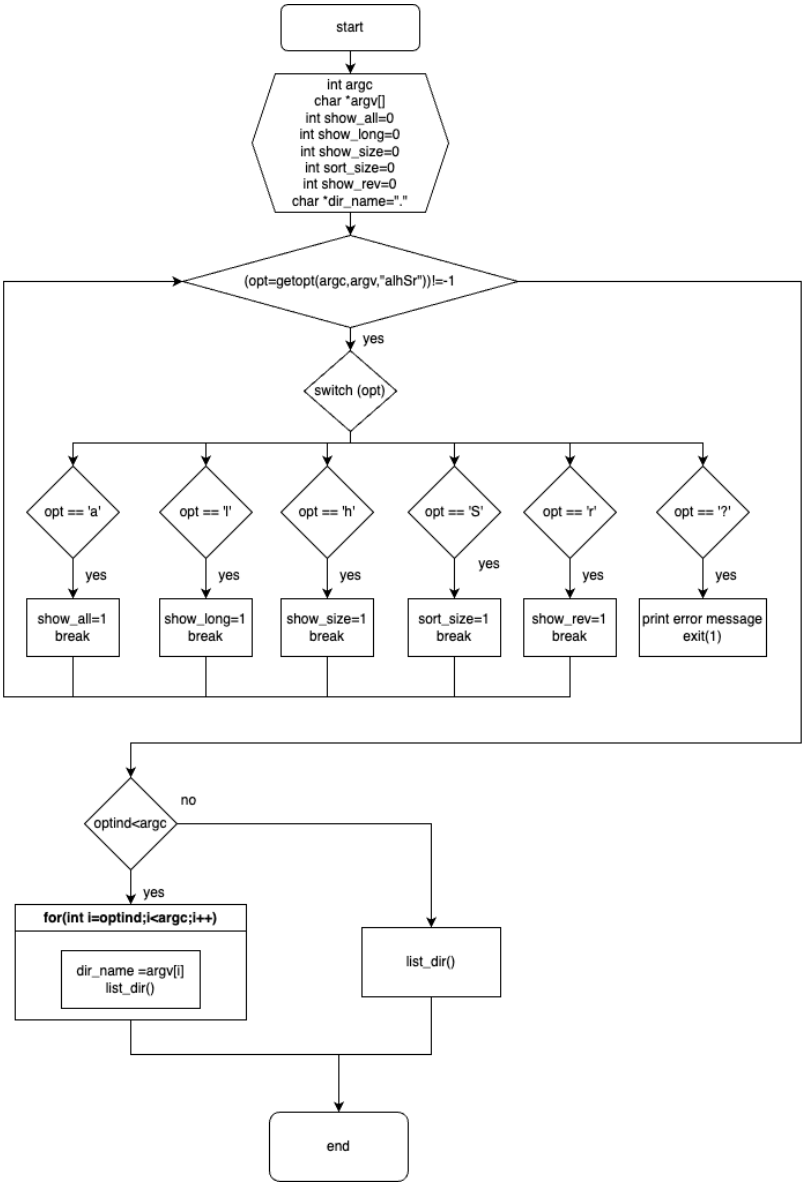
만약 인자에 wildcard가 존재할 경우 실행되는 함수이다. 옵션은 없다.

J. list_dir



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

K. int 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.

3. Pseudo Code

A. print_permission

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

```
        if have execute permission by other, print "x" else "-"
```

B. print_UID

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

C. print_GID

```
get strcut group pointer gr
if gr=getgrgid(gid) is not NULL
        print pw -> pw_name
else
        print 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 print 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
        print time
```

F. print_long

```
use print_permission function
print file_stat,st_nlink
use print_UID function
use print GID function
if -h option activated
        use print_readableSIZE function
else
        just print file_stat.st_size
use print_LastModified_time function
```

```
print file name
```

G. file_print

```
get strcut stat file_stat
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 strct 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
                just print name step by step
        else
                print name reverse
else
```

```
        get absolute_path by realpath function
        print directory and total size of blocks
        if -r option has not activated
                just print name step by step
        else
                print name reverse
closedir about dirp
```

I. wildcard_print

```
get struct stat file_stat
make file_info struct with name and stat
DIR pointer dirp
get struct dirent pointer dir
get lastpath by find '/'
if cannot find '/'
        path is current working directory
        pattern is dir_name
if can find '/'
        path is lastpath before finding '/'
        pattern is after finding '/'

opendir about path
get struct file_info files about 1024 array
while readdir about dirp is not NULL
        if hidden file when pattern[0] is not '.' continue
        else if not hidden file when pattern[0] is '.' continue
        if filename is "." or ".." continue
        if find pattern in dir->d_name
        set tmppath to get stat
        make path dir_name/dir->d_name
        get stat about tmppath to files struct
        get name about tmppath
sort by ascending order about name
if directory is on path
        print files in directory
else
        print files name

closedir about dirp
```

J. last_dir

```
DIR pointer dirp
get struct dirent pointer tmp
inspect dir_name if have wildcard like '*' ,'[','] or '?'
if find wildcard
        use wildcard_print function to print

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

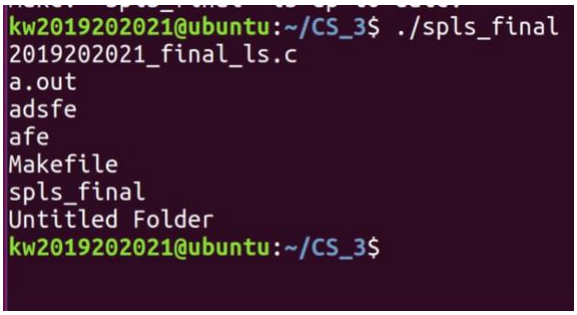
K. main

```
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
find argv about wildcard '*', '?', '['
        if find wildcard set isWildcard
sort ascending order by name about argv

if optind is smaller than argc
        dir_name is argv
        use list_dir function to manage
else
        use list_dir function to manange
```

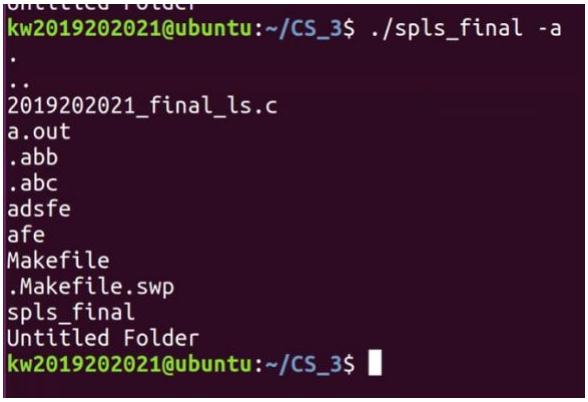
4. 결과화면

A. 기본출력



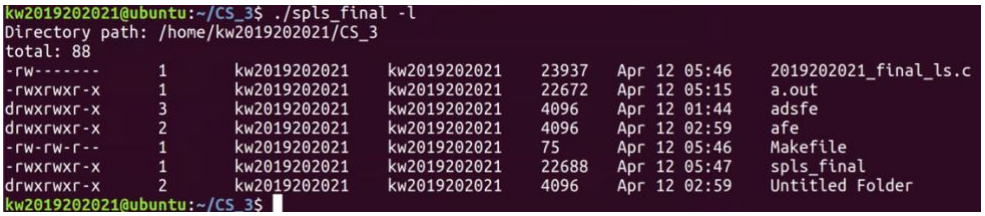
-옵션 없이 출력했을 때 결과이다. 숨김 파일 없이 공개된 파일만 출력하였다.

B. -a 옵션



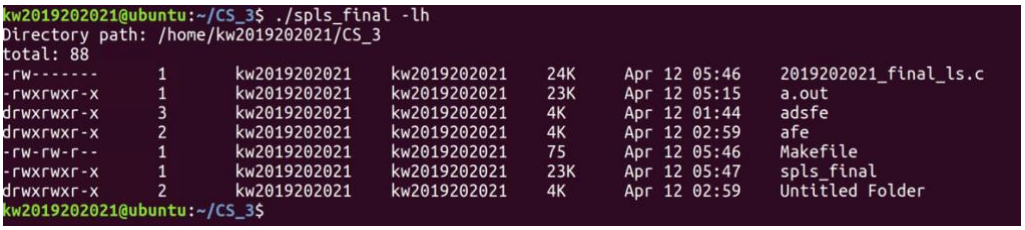
-옵션 -a를 하였을 때 결과이다. 숨김 파일까지 모두 출력되는 것을 확인할 수 있다.

C. -l 옵션



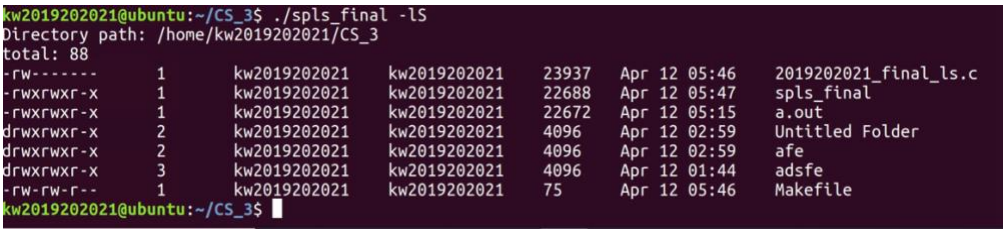
-옵션 -l을 하였을 때 결과이다. long format에 맞춰서 각 파일 별로 상세정보를 보여주고 있으며 파일이름을 오름차순으로 출력하였다.

D. -lh 옵션



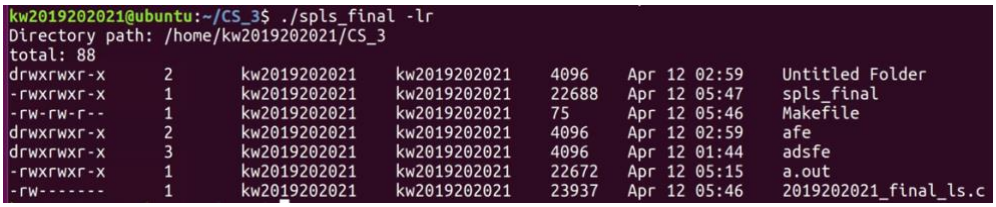
-옵션 -lh를 하였을 때 결과이다. -h 옵션은 size를 readable하도록 만드는 것인데 -l 없이 혼자 사용되지 못하므로 같이 진행하였다. 10^9, 10^6, 10^3을 나눴을 때 가능하다면 단위를 맞추도록 하였으며 반올림 처리하여 23937은 24K가 됨을 확인할 수 있다.

E. -lS 옵션



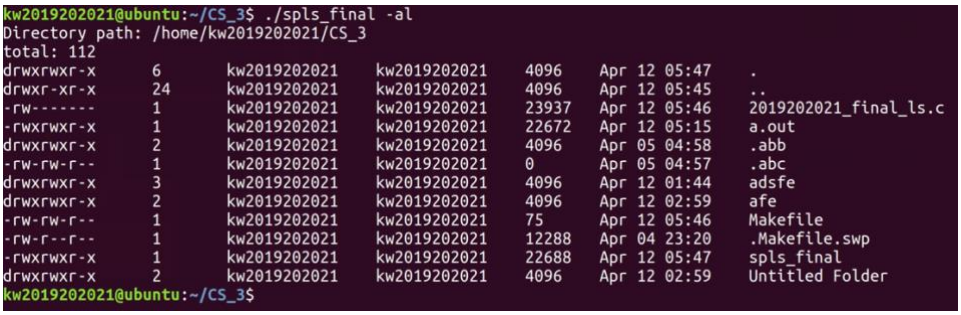
-Size 크기로 내림차순 정렬을 하는 옵션 -S와 -l을 사용했을 때 결과이다. size 가 큰 순서대로 출력이 됨을 확인 할 수 있고 사이즈가 같은 directory 같은 경우 이름을 내림차순으로 출력하였다.

F. -lr 옵션



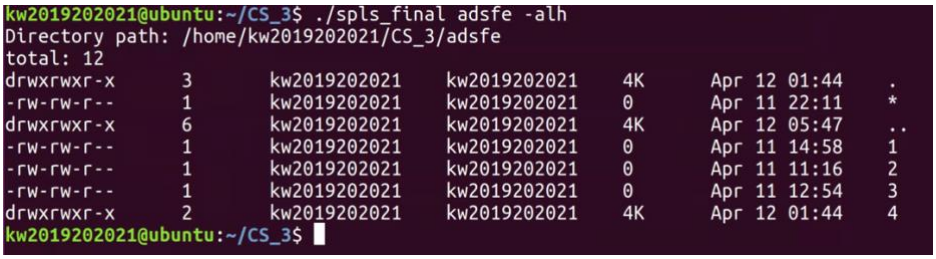
-r 옵션을 사용하였을 때 결과로 순서를 반대로 하는 옵션이다. 기존 -l 옵션만 할 때와 반대의 출력을 보여 이름으로 내림차순 출력을 하는 것을 볼 수 있다.

G. -al 옵션



-지난 시간에서 진행했던 옵션으로 모든 파일을 long format에 맞춰 상세하게 출력하고 있다.

H. -alh 옵션



-alh 옵션을 같이 썼을 때 결과로 모든 파일을 long format에 맞춰 상세하게 출력하되 사이즈는 readable size로 단위를 만들어서 출력하도록 만들었다.

I. -alSh 옵션

```
kw2019202021@ubuntu:~/CS_3$ ./spls_final adsfe -alSh
Directory path: /home/kw2019202021/CS_3/adsfe
total: 12
drwxrwxr-x      2       kw2019202021    kw2019202021    4K      Apr 12 01:44    4
drwxrwxr-x      6       kw2019202021    kw2019202021    4K      Apr 12 05:47    ..
drwxrwxr-x      3       kw2019202021    kw2019202021    4K      Apr 12 01:44    .
-rw-rw-r--      1       kw2019202021    kw2019202021    0       Apr 11 12:54    3
-rw-rw-r--      1       kw2019202021    kw2019202021    0       Apr 11 11:16    2
-rw-rw-r--      1       kw2019202021    kw2019202021    0       Apr 11 14:58    1
-rw-rw-r--      1       kw2019202021    kw2019202021    0       Apr 11 22:11    *
kw2019202021@ubuntu:~/CS_3$
```

-H번 옵션에서 -S 옵션을 추가한 것으로 사이즈 별로 내림차순 정렬을 하였으며 같은 사이즈는 이름별로 내림차순이 된 것을 확인 할 수 있다.

J. -alhr 옵션

```
kw2019202021@ubuntu:~/CS_3$ ./spls_final adsfe -alhr
Directory path: /home/kw2019202021/CS_3/adsfe
total: 12
drwxrwxr-x      2       kw2019202021    kw2019202021    4K      Apr 12 01:44    4
-rw-rw-r--      1       kw2019202021    kw2019202021    0       Apr 11 12:54    3
-rw-rw-r--      1       kw2019202021    kw2019202021    0       Apr 11 11:16    2
-rw-rw-r--      1       kw2019202021    kw2019202021    0       Apr 11 14:58    1
drwxrwxr-x      6       kw2019202021    kw2019202021    4K      Apr 12 05:47    ..
-rw-rw-r--      1       kw2019202021    kw2019202021    0       Apr 11 22:11    *
drwxrwxr-x      3       kw2019202021    kw2019202021    4K      Apr 12 01:44    .
kw2019202021@ubuntu:~/CS_3$
```

- H번 옵션에서 -r 옵션을 추가하여 이름으로 오름차순을 하였던 것이 내림차순으로 변경된것을 확인할 수 있다.

K. -alShr 옵션

```
kw2019202021@ubuntu:~/CS_3$ ./spls_final adsfe -alShr
Directory path: /home/kw2019202021/CS_3/adsfe
total: 12
-rw-rw-r--      1       kw2019202021    kw2019202021    0       Apr 11 22:11    *
-rw-rw-r--      1       kw2019202021    kw2019202021    0       Apr 11 14:58    1
-rw-rw-r--      1       kw2019202021    kw2019202021    0       Apr 11 11:16    2
-rw-rw-r--      1       kw2019202021    kw2019202021    0       Apr 11 12:54    3
drwxrwxr-x      3       kw2019202021    kw2019202021    4K      Apr 12 01:44    .
drwxrwxr-x      6       kw2019202021    kw2019202021    4K      Apr 12 05:47    ..
drwxrwxr-x      2       kw2019202021    kw2019202021    4K      Apr 12 01:44    4
kw2019202021@ubuntu:~/CS_3$
```

- I번 옵션에서 -r 옵션을 진행한 것으로 사이즈로 내림차순 되어있던 것이 오름차순이 되었으며 마찬가지로 같은 사이즈인 경우에는 이름으로 오름차순으로 변경되었다.

```
kw2019202021@ubuntu:~/CS_3$ ./spls_final -alShr
Directory path: /home/kw2019202021/CS_3
total: 112
-rw-rw-r--      1       kw2019202021    kw2019202021    0       Apr 05 04:57    .abc
-rw-rw-r--      1       kw2019202021    kw2019202021    75      Apr 12 05:46    Makefile
drwxrwxr-x      6       kw2019202021    kw2019202021    4K      Apr 12 05:47    .
drwxr-xr-x      24      kw2019202021    kw2019202021    4K      Apr 12 05:45    ..
drwxrwxr-x      2       kw2019202021    kw2019202021    4K      Apr 05 04:58    .abb
drwxrwxr-x      3       kw2019202021    kw2019202021    4K      Apr 12 01:44    adsfe
drwxrwxr-x      2       kw2019202021    kw2019202021    4K      Apr 12 02:59    afe
drwxrwxr-x      2       kw2019202021    kw2019202021    4K      Apr 12 02:59    Untitled Folder
-rw-r--r--      1       kw2019202021    kw2019202021    12K     Apr 04 23:20    .Makefile.swp
-rwxrwxr-x      1       kw2019202021    kw2019202021    23K     Apr 12 05:15    a.out
-rwxrwxr-x      1       kw2019202021    kw2019202021    23K     Apr 12 05:47    spls_final
-rw-------      1       kw2019202021    kw2019202021    24K     Apr 12 05:46    2019202021_final_ls.c
kw2019202021@ubuntu:~/CS_3$
```

-current working directory에서 한 결과로 마찬가지로 사이즈로 오름차순하고 있고 같은 사이즈는 이름으로 오름차순한다.

L. wildcard '*'

```
kw2019202021@ubuntu:~/CS_3$ ./spls_final '*'
2019202021_final_ls.c
a.out
Directory path: /home/kw2019202021/CS_3/adsfe
*
1
2
3
4

Directory path: /home/kw2019202021/CS_3/afe
Untitled Document

Makefile
spls_final
Directory path: /home/kw2019202021/CS_3/Untitled Folder

kw2019202021@ubuntu:~/CS_3$
```

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

```
kw2019202021@ubuntu:~/CS_3$ ./spls_final 'a*'
a.out
Directory path: /home/kw2019202021/CS_3/adsfe
*
1
2
3
4

Directory path: /home/kw2019202021/CS_3/afe
Untitled Document
```

-'a*' 와일드 카드를 사용하여 a로 시작하는 패턴을 찾아 출력하여 a.out, asdsfe, afe가 출력되었으며 디렉토리는 경로와 함께 하위 파일도 출력하였다.

```
kw2019202021@ubuntu:~/CS_3$ ./spls_final ~ '*'
2019202021_final_ls.c
a.out
Directory path: /home/kw2019202021/CS_3/adsfe
*
1
2
3
4

Directory path: /home/kw2019202021/CS_3/afe
Untitled Document

Makefile
spls_final
Directory path: /home/kw2019202021/CS_3/Untitled Folder

Directory path: /home/kw2019202021
CS_1
CS_2
CS_3
CSE
Desktop
Documents
Downloads
examples.desktop
Music
Pictures
Public
Templates
tmp
Videos
kw2019202021@ubuntu:~/CS_3$
```

-제안서에 있던 예시로 홈디렉토리를 의미하는 ~와 와일드카드 '*'를 사용했을 때 결과로 ~가 먼저 사용되었지만 이름 순으로 '*'가 먼저 출력되었다. 그 후 홈 디렉토리에 대한 절대 경로와 파일 이름들이 출력되었다.

M. wildcard '.*'

```
kw2019202021@ubuntu:~/CS_3$ ./spls_final '.*'
Directory path: /home/kw2019202021/CS_3/.abb

.abc
.Makefile.swp
kw2019202021@ubuntu:~/CS_3$
```

-숨김파일에 대한 와일드카드 예시이다. '.*'을 한 경우에는 숨김파일에 대해서만 출력을 할 수 있다.

N. wildcard '?'

```
kw2019202021@ubuntu:~/CS_3$ ./spls_final 'a????'
a.out
Directory path: /home/kw2019202021/CS_3/adsfe
*
1
2
3
4

kw2019202021@ubuntu:~/CS_3$
```

-a로 시작하고 나머지 4글자는 상관 없이 찾는 즉 a로 시작하는 5글자 파일을 찾는 것으로 a.out과 adsfe에 대해 출력이 되었고 디렉토리는 하위 파일도 같이 출력하였다.

O. wildcard '[]'

```
kw2019202021@ubuntu:~/CS_3$ ./spls_final 'adsfe/[0-4]'
1
2
3
Directory path: /home/kw2019202021/CS_3/adsfe/4
5

kw2019202021@ubuntu:~/CS_3$
```

와일드카드 '[seq]'에 대한 결과로 0부터 4까지 출력하도록 하였지만 0은 없으니 무시하고 1,2,3,4가 출력되었고 4는 디렉토리이므로 하위 파일도 출력되었다.

P. spls_final 파일 '*'

```
kw2019202021@ubuntu:~/CS_3$ ./spls_final adsfe/1 '*'
2019202021_final_ls.c
a.out
Directory path: /home/kw2019202021/CS_3/adsfe
*
1
2
3
4

Directory path: /home/kw2019202021/CS_3/afe
Untitled Document

Makefile
spls_final
Directory path: /home/kw2019202021/CS_3/Untitled Folder

adsfe/1
kw2019202021@ubuntu:~/CS_3$
```

경로 뿐만 아니라 파일을 와일드카드와 같이 사용해도 문제 없이 정상 출력하는 것을 확인할 수 있다.

5. 고찰

Assignment 1-1, 1-2, 1-3에 걸쳐 ls 명령어를 구현하여 option을 받아 option 별로 다르게 출력하고 wildcard에 따라 해당 파일을 출력하도록 설계하였다. Assignment 1-2에서 코드가 복잡하고 뒤죽박죽이라 많은 부분을 함수로 만들어 공통적인 부분은 각 함수로 모두 처리할 수 있도록 설계를 진행하였으나 시간 부족으로 인해 이번 Assignment 1-3에서 새롭게 설계한 부분은 함수로 따로 뺄 수 없고 크게 잡아서 만들었다.

또한 이번에 과제를 하며 느낀 점이 있다면, 이미 library에서 주어지는 함수를 얼만큼 알고 어떻게 적절히 쓰는 것에 따라 코드 구현 난이도가 달라졌다. 와일드카드를 처음 구현할 때는 패턴을 읽는 함수를 직접 제작하려 했으나 너무 복잡해져서 인터넷 서핑을 통해 terminal에서 사용하는 grep 처럼 패턴을 읽는 것 뿐만 아니라 이번 과제에서 주어진 wildcard를 빼고 패턴을 찾아주는 함수임을 발견하여 사용하였다. 다음 과제에서는 http 관련 과제라고 알고 있으므로 해당 관련 함수를 미리 숙지하고 필요할 때 적재적소에 넣을 수 있도록 할 예정이다.

마지막으로 linux에서 구현하고 실행 또한 터미널에서 해야하다보니 이전까지 코딩할 때 디버그를 사용했던 것을 사용하지 못하니 에러가 발생하면 코드에 임의로 printf를 넣어 오류가 발생하는 부분을 찾았다. 이전 데이터구조 설계에서는 에러에 대해서는 txt 파일로 만들도록 하였지만 이번 과제에서 txt 파일 생성해도 된다는 말이 없어 사용하지 않았는데 남은 6개의 과제는 사용하여 디버그의 기능을 비슷하게 만들어야겠다는 생각을 하였다.

6. Reference

1) fnmatch/ https://www.ibm.com/docs/ko/aix/7.3?topic=f-fnmatch-subroutine
2) 시스템프로그래밍 강의자료/광운대학교/컴퓨터정보공학부/김태석 교수님/2023
3) 시스템프로그래밍실습 Assignment 1-3/광운대학교/컴퓨터정보공학부/김태석 교수님/2023