

# 시스템프로그래밍

## assignment1-2

학번: 2021202045

이름: 김예은

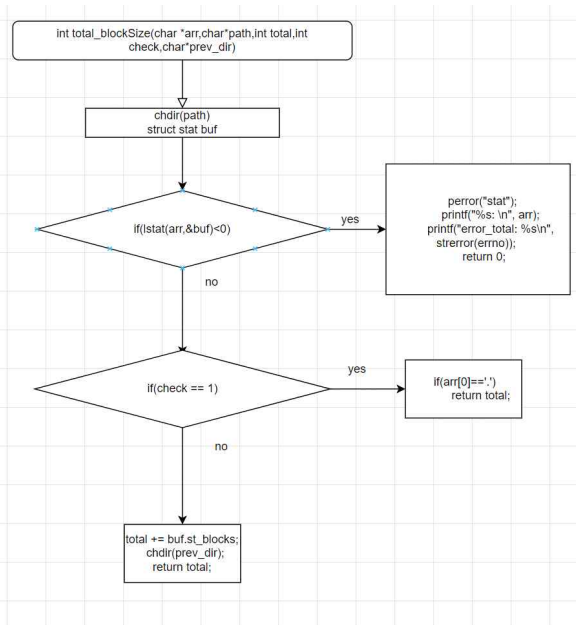
담당 교수님: 최상호 교수님

## I . Introduction

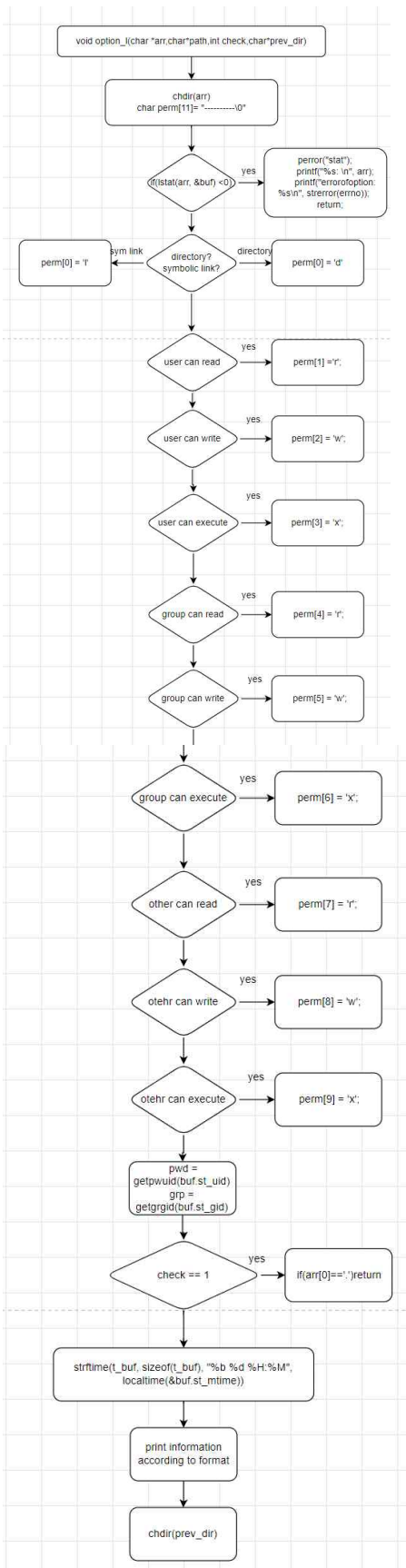
지난 과제에 이어 ls를 구현하되, '-a'/'-l'/'-la' 옵션이 추가된 ls를 구현하는 것이 이번 과제의 포인트이다. -a의 경우, hidden file을 포함하여 sorting 후에 출력하는 것이고, -l의 경우 인자로 준 파일의 정보(파일타입,접근권한,링크수,user name,group name,파일사이즈,수정시간및날짜,파일명)을 출력해주는 옵션이다. 인자 수의 제한은 없으며, 파일과 디렉토리가 입력될 때, 각각 따로 정렬되어 파일 먼저 출력해준다. 만약 없는 파일이라면 오류메세지 출력이후 후에 인자에 대해 ls기능을 해야한다. 실행파일명 이름은 ./spls\_advanced이며 코드파일명은 학번\_spls\_advanced.c이다.

## II . flowchart

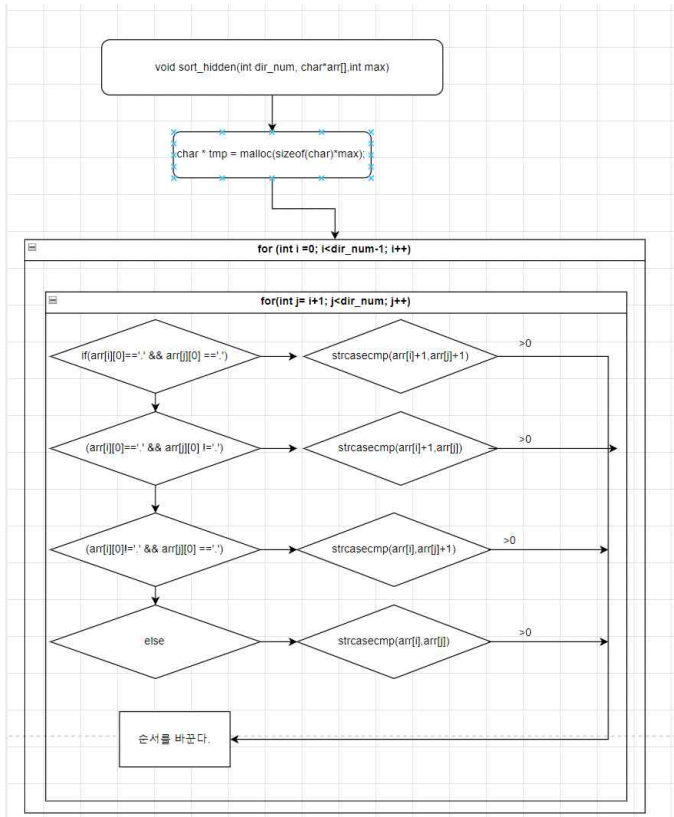
(1) total\_blockSize함수 -> int 반환형



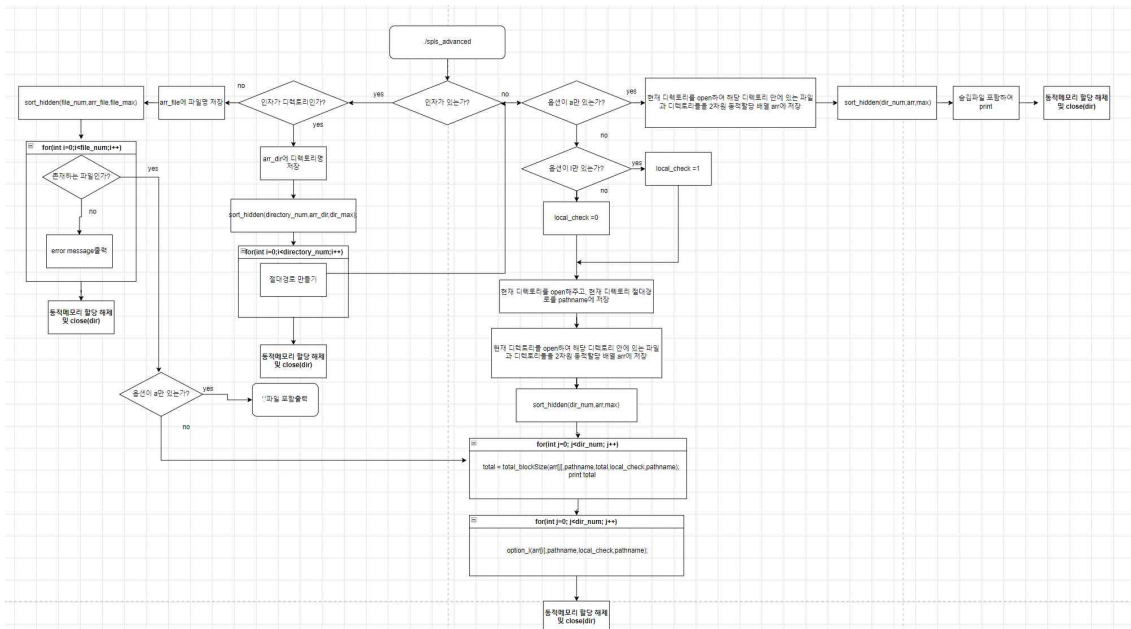
(2) option\_l함수 -> void형



### (3) sort\_hidden 함수 -> void형



### (4) main 함수



인자가 있는지 확인-> 옵션이 무엇인지 확인-> 인자가 디렉토리인지의 흐름에 따라 구현한다.

### III. Pseudo code

int total\_blockSize는 디렉토리나 파일의 총 블록 사이즈를 구하는 함수이다.

인자로 받은 path로 우선 이동한다. 그 다음, stat구조체 변수 buf를 선언해준다.

stat 인자로 받은 인자로 받은 디렉토리명과 buf를 넣어준다.

만약 lstat에 문제가 생길 시, perror와 errno를 통해 오류 메시지를 출력해준다.

함수 인자로 받은 check는 숨김 파일을 고려해야할지 안해야할지를 알려주는 signal로

만약 check == 1이라면,

숨김파일일 때, 블록 사이즈를 더하지 않고 함수를 끝낸다.

아니라면,

함수인자로 받은 기존 total를 업데이트 해준다.

이전 경로를 함수인자로 받았으므로 이전경로(prev\_dir)로 다시 나가주고, total을 반환하면서 함수를 끝낸다.

void option\_l함수는 인자로 받은 파일이나 디렉토리의 정보를 구한 후, 출력해준다.

permission에 대한 정보를 저장하기 위해 perm[11]을 선언 및 맨 처음엔 모두 '-'로 초기화 해준다.

구조체 stat 변수 buf를 정의한다.

user name, group name을 출력하기 위한 passwd, group 구조체 변수 pwd, grp도 각각 선언한다.

인자로 받은 path로 이동한다.

만약 lstat에 대한 에러가 발생할 때, 에러 메시지를 출력해준다.

buf에 저장된 파일이나 디렉토리의 타입을 분석하여 perm[0]에 표시한다.

user, group, other에 대한 접근 권한 총 9가지에 대해 if문을 통해 분석하고,

해당 접근권한이 있을 시 각 자리에 맞게 r,w,x를 써준다.

pwd를 통해 user id를 가져오고, grp를 이용해 group id를 가져온다.

t\_buf에 strftime을 통해 time을 localtime으로 바꿔준 후, format형으로 저장해준다.

만약 함수인자로 받은 check가 1이라면 파일명이 '.'으로 시작하는 것은 출력을 무시한다.

접근권한,link수, user name, group name,size,수정시간,파일명을 '\t'로 구분하여 출력한 후,

prev\_dir로 되돌아간다.

void sort\_hidden함수는 '.'으로 시작되는 파일도 포함하여 파일명을 정렬해주는 함수이다.

bubble sorting을 써줄 것이기 때문에 임시 저장해줄 array tmp를 동적할당 해준다.

해당 array는

숨김 파일과 숨김 파일일 때,

숨김 파일과 숨김 파일이 아닐때,

숨김파일이 아니고 숨김파일일때,

둘다 숨김 파일이 아닐때

총 4개의 경우의 수를 통해 먼저 '.'을 빼고 비교해준 뒤, 다시 '.'을 포함하여 정렬해준다.

만약 숨김 파일이라면 arr[i][0] == '.'이므로 arr[i]+1을 해주면 '.'이후의 문자열을 가리키는 것이다.

만약 arr[0]와 arr[1]의 위치가 바뀌어야 한다면, tmp에 arrp[0]값을 저장  
arr[0]에 arr[1]값을 복사하고  
arr[1]에는 tmp값을 복사하여 저장한다.

main함수

getopt를 통해 옵션에 a가 있다면 aflag를 1로 set, l이 있다면 lflag를 1로 set한다.

aflag값과 argc값을 비교한다.

만약 argc ==1이라면, 즉 옵션과 인자가 둘다 없다면

자동으로 현재 디렉토리를 open하여 해당 디렉토리 안에 있는 파일과 디렉토리들을 2차원 동적할당 배열 arr에 저장한다.

디렉토리내의 entry 수들을 dir\_num으로 정의하고, entry name중 가장 긴 이름을 max로 정의한다.

sort\_hidden함수를 call한다.

숨김파일이라면(파일 이름이 .으로 시작하면) print를 하지 않고, entry들을 정렬된 순서대로 출력한다.

동적할당해준 것을 모두 해제해준 다음, main함수를 끝낸다.

옵션은 a인데 인자가 없다면,

현재 디렉토리를 자동으로 열어주고, a옵션 없을 때랑 똑같은 알고리즘으로 수행한다.

단, print를 할 때, '.'으로 시작하는 entry도 출력해준다.

동적할당해준 것을 모두 해제해준 다음, main함수를 끝낸다.

옵션이 l인데 인자가 없다면 local\_check =1, 옵션이 la인데 인자가 없다면 local\_check =0으로 set한다.

자동으로 현재 디렉토리를 open해주고, 현재 디렉토리 절대경로를 pathname에 저장한다.

현재 디렉토리 안에 있는 entry들을 2d array에 저장하고 이를 차례로 total\_blockSize()에 넣어 블록 사이즈의 총합을 알아낸다.

그 다음 다시 반복문을 통해 모든 entry들을 option\_l함수에 넣는다.

동적할당해준 것을 모두 해제해준 다음, main함수를 끝낸다.

옵션이 있고, 인자가 있다면 우선 인자들을 파일인지 디렉토리인지 구분해서 opendir반환값에 따라(따로 정렬해주기위해) 각각

for문을 통해 옵션다음부터 인자값 끝까지 돌면서 arr\_file, arr\_dir에 저장해준다. 연 directory는 다시 close해준다.

arr\_file, arr\_dir은 각각 요소들을 정렬해줘야하기 때문에 sort\_hidden함수를 각각 호출하여 정렬해준다.

옵션이 a만 있다면,

file의 경우 arr\_file 처음부터 끝까지 for문을 돌면서 F\_OK를 통해 존재하는 파일인지 아닌지 판단한 다음 파일을 출력해준다.

directory의 경우 arr\_dir 처음부터 끝까지 for문을 돌면서 해당 디렉토리를 open및 read하

여 해당 디렉토리 내의 entry들을

arr이라는 2d array에 저장한다. arr의 요소들도 정렬해줘야하기 때문에, arr요소를 인자로 sort\_hidden함수를 call 한다.

a옵션은 히든파일까지 출력해주는 것이므로, arr의 처음부터 끝까지 모두 entry를 출력해준다음 메모리 할당 해제 및 디렉토리를 닫아준다.

옵션이 al 이라면,

file의 경우 arr\_file 처음부터 끝까지 반복하면서 파일이 없는 경우를 제외하고, 현재 디렉토리를 저장하여 출력한 다음, option\_l함수로 넘어간다.

디렉토리의 경우, p\_path, path를 동적할당 배열로 선언하여 절대경로를 만들어준다. 우선 현재 디렉토리를 prev\_dir에 저장해놓고, p\_path에 /을 저장한다.

p\_path에 인자로 줄 디렉토리를 복사하여 최종적으로 /arr\_dir[i]형식으로 만든다음 현재 디렉토리를 앞에 붙임으로써 절대경로를 완성시킨다.

arr\_dir 처음부터 끝까지 for문을 돌면서 해당 디렉토리를 open및 read하여 해당 디렉토리 내의 entry들을

arr이라는 2d array에 저장한다. arr의 요소들도 정렬해줘야하기 때문에, arr요소를 인자로 sort\_hidden함수를 call 한다.

그 다음 정렬된 entry들을 각각 total\_blockSize()에 넣는다.

정렬된 entry들을 각각 option\_l함수에 넣는다. 이때 옵션이 a가 있기 때문에 함수인자에 check부분에 0을 넣는다.

메모리 할당 해제 및 디렉토리를 닫아준다.

만약 옵션이 l만 있다면,

위와 동일한 알고리즘에 option\_l함수를 부를 때 check부분에 1을 넣는다.

메모리 할당 해제 및 디렉토리를 닫아준다.

옵션이 없고 인자만 있다면,

file의 경우 arr\_file 처음부터 끝까지 for문을 돌면서 F\_OK를 통해 존재하는 파일인지 아닌지 판단한 다음 파일을 출력해준다.

directory의 경우 arr\_dir 처음부터 끝까지 for문을 돌면서 해당 디렉토리를 open및 read하여 해당 디렉토리 내의 entry들을

arr이라는 2d array에 저장한다. arr의 요소들도 정렬해줘야하기 때문에, arr요소를 인자로 sort\_hidden함수를 call 한다.

arr의 처음부터 끝까지 '.'으로 시작되는 entry를 제외한 entry를 출력해준다음 메모리 할당 해제 및 디렉토리를 닫아준다.

#### IV. 결과화면

```
kw2021202045@ubuntu:~/kw_hw$ ./spls_advanced
2021202045_simple_ls.c
2021202045_spls_advanced.c
apple
Banana
link_apple
Makefile
spls_advanced
test.txt
test2
Web1_1
```

인자와 옵션이 없을 때 숨김파일 빼고 디렉토리 내 entry를 정렬하여 보여준다.

```
kw2021202045@ubuntu:~/kw_hw$ ./spls_advanced -la
Directory path: /home/kw2021202045/kw_hw
total : 84
drwxrwxr-x 6 kw2021202045 kw2021202045 4096 Apr 05 00:42 .
drwxr-xr-x 20 kw2021202045 kw2021202045 4096 Apr 04 07:00 ..
-rw-rw-r-- 1 kw2021202045 kw2021202045 3176 Mar 26 15:20 2021202045_simple_ls.c
-rw-rw-r-- 1 kw2021202045 kw2021202045 26408 Apr 05 00:11 2021202045_spls_advanced.c
drwxrwxr-x 3 kw2021202045 kw2021202045 4096 Apr 04 03:50 apple
drwxrwxr-x 2 kw2021202045 kw2021202045 4096 Mar 24 05:14 Banana
lrwxrwxrwx 1 kw2021202045 kw2021202045 5 Apr 04 07:01 link_apple
-rw-rw-r-- 1 kw2021202045 kw2021202045 56 Apr 04 07:00 Makefile
-rwxrwxr-x 1 kw2021202045 kw2021202045 22408 Apr 05 00:10 spls_advanced
-rw-rw-r-- 1 kw2021202045 kw2021202045 0 Mar 26 02:55 test.txt
-rw-rw-r-- 1 kw2021202045 kw2021202045 0 Mar 24 09:20 test2
drwxrwxr-x 2 kw2021202045 kw2021202045 4096 Mar 24 04:55 .vscode
drwxrwxr-x 2 kw2021202045 kw2021202045 4096 Apr 02 21:36 Web1_1
kw2021202045@ubuntu:~/kw_hw$ ls -la
total 84
drwxrwxr-x 6 kw2021202045 kw2021202045 4096 Apr 5 00:42 .
drwxr-xr-x 20 kw2021202045 kw2021202045 4096 Apr 4 07:00 ..
-rw-rw-r-- 1 kw2021202045 kw2021202045 3176 Mar 26 15:20 2021202045_simple_ls.c
-rw-rw-r-- 1 kw2021202045 kw2021202045 26408 Apr 5 00:11 2021202045_spls_advanced.c
drwxrwxr-x 3 kw2021202045 kw2021202045 4096 Apr 4 03:50 apple
drwxrwxr-x 2 kw2021202045 kw2021202045 4096 Mar 24 05:14 Banana
lrwxrwxrwx 1 kw2021202045 kw2021202045 5 Apr 4 07:01 link_apple -> apple
-rw-rw-r-- 1 kw2021202045 kw2021202045 56 Apr 4 07:00 Makefile
-rwxrwxr-x 1 kw2021202045 kw2021202045 22408 Apr 5 00:10 spls_advanced
-rw-rw-r-- 1 kw2021202045 kw2021202045 0 Mar 24 09:20 test2
-rw-rw-r-- 1 kw2021202045 kw2021202045 0 Mar 26 02:55 test.txt
drwxrwxr-x 2 kw2021202045 kw2021202045 4096 Mar 24 04:55 .vscode
drwxrwxr-x 2 kw2021202045 kw2021202045 4096 Apr 2 21:36 Web1_1
```

인자는 없고 옵션만 al일 때, 숨김파일과 entry에 대한 정보를 잘 출력하는 것을 볼 수 있다. 숨김파일의 경우 '.'을 포함하여 정렬된 것을 확인할 수 있고, symbolic link 또한 파일 타입이 잘 나타난 것을 볼 수 있다. 위 사진은 진짜 ls -al로과 비교할 수 있으며 ./spls\_advanced -al과 비교했을 때 동일하게 나오는 것을 볼 수 있다.



```
kw2021202045@ubuntu:~/kw_hw$ ./spls_advanced -a apple Web test2
test2
apple:
.
..
good
test
Web:
.
..
hello1.c
Makefile
test
Web1 1 B 2021202045.pdf
```

위 사진은 인자를 여러 개 주고 옵션이 -a일때이다. apple및 Web은 디렉토리이고, test2는 일반 파일이기 때문에 test2를 먼저 출력한 뒤, 디렉토리끼리 정렬하여 각 디렉토리 내부의 entry를 숨김파일 포함하여 출력해주는 것을 확인할 수 있다.

```
kw2021202045@ubuntu:~/kw_hw$ ./spls_advanced -al dkfjk apple Web
cannot access 'dkfjk' : No such directory
Directory path: /home/kw2021202045/kw_hw/apple
total : 12
drwxrwxr-x 3 kw2021202045 kw2021202045 4096 Apr 04 03:50 .
drwxrwxr-x 6 kw2021202045 kw2021202045 4096 Apr 04 18:49 ..
drwxrwxr-x 2 kw2021202045 kw2021202045 4096 Mar 24 09:20 good
-rw-rw-r-- 1 kw2021202045 kw2021202045 0 Mar 24 09:20 test
Directory path: /home/kw2021202045/kw_hw/Web
total : 208
drwxrwxr-x 2 kw2021202045 kw2021202045 4096 Apr 02 21:36 .
drwxrwxr-x 6 kw2021202045 kw2021202045 4096 Apr 04 18:49 ..
-rwxrwxrwx 1 kw2021202045 kw2021202045 832 Apr 02 21:36 hello1.c
-rw-rw-r-- 1 kw2021202045 kw2021202045 29 Mar 30 19:16 Makefile
-rwxrwxr-x 1 kw2021202045 kw2021202045 8744 Apr 02 21:36 test
-rwxrw-rw- 1 kw2021202045 kw2021202045 183053 Mar 28 04:22 Web1_1_B_2021202045.pdf
```

인자를 여러 개 주고 옵션이 -al일때이다. 없는 파일에 대해 먼저 없다고 출력해준뒤, 뒤에 인자로 넘어가서 각각의 디렉토리에 대해 -al옵션을 수행해준다.

```
kw2021202045@ubuntu:~/kw_hw$ ls -al dkfjk apple Web
ls: cannot access 'dkfjk': No such file or directory
apple:
total 12
drwxrwxr-x 3 kw2021202045 kw2021202045 4096 Apr 4 03:50 .
drwxrwxr-x 6 kw2021202045 kw2021202045 4096 Apr 4 18:49 ..
drwxrwxr-x 2 kw2021202045 kw2021202045 4096 Mar 24 09:20 good
-rw-rw-r-- 1 kw2021202045 kw2021202045 0 Mar 24 09:20 test

Web:
total 208
drwxrwxr-x 2 kw2021202045 kw2021202045 4096 Apr 2 21:36 .
drwxrwxr-x 6 kw2021202045 kw2021202045 4096 Apr 4 18:49 ..
-rwxrwxrwx 1 kw2021202045 kw2021202045 832 Apr 2 21:36 hello1.c
-rw-rw-r-- 1 kw2021202045 kw2021202045 29 Mar 30 19:16 Makefile
-rwxrwxr-x 1 kw2021202045 kw2021202045 8744 Apr 2 21:36 test
-rwxrw-rw- 1 kw2021202045 kw2021202045 183053 Mar 28 04:22 Web1_1_B_2021202045.pdf
```

위 사진은 실제 ls 출력 장면이다.

```

kw2021202045@ubuntu:~/kw_hw$ ./spls_advanced -al /home /var
Directory path: /home
total : 12
drwxr-xr-x    3      root      root    4096    Mar 16 03:24  .
drwxr-xr-x   24      root      root    4096    Mar 18 05:10  ..
drwxr-xr-x   20 kw2021202045  kw2021202045  4096    Apr 04 07:00  kw2021202045
Directory path: /var
total : 56
drwxr-xr-x   14      root      root    4096    Jul 30 17:50  .
drwxr-xr-x   24      root      root    4096    Mar 18 05:10  ..
drwxr-xr-x    2      root      root    4096    Mar 24 09:14  backups
drwxr-xr-x   16      root      root    4096    Mar 16 03:25  cache
drwxrwxrwx    2      root    whoopsie  4096    Jul 30 17:40  crash
drwxr-xr-x   66      root      root    4096    Mar 16 03:27  lib
drwxrwxr-x    2      root    staff    4096    Apr 12 13:14  local
lrwxrwxrwx    1      root      root     9    Mar 16 03:23  lock
drwxrwxr-x   13      root    syslog   4096    Apr 04 20:05  log
drwxrwxr-x    2      root    mail     4096    Jul 30 17:30  mail
drwxrwxrwx    2      root    whoopsie  4096    Jul 30 17:40  metrics
drwxr-xr-x    2      root      root    4096    Jul 30 17:30  opt
lrwxrwxrwx    1      root      root     4    Mar 16 03:23  run
drwxr-xr-x    2      root      root    4096    Jul 19 03:48  snap
drwxr-xr-x    7      root      root    4096    Jul 30 17:37  spool
drwxrwxrwx    6      root      root    4096    Apr 02 00:31  tmp

```

인자로 /home /var을 주었을 때 /home 다음에 /var에 대해 잘 출력한다. 경로도 잘 출력되는 것을 볼 수 있다.

## V. 고찰

같은 코드가 계속 반복되기 때문에 이를 함수화해서 뺐으면 메인함수가 더 깔끔할 거라고 생각이 된다. 하지만 기간 안에 구현해야한다는 조바심과 맨처음 잘못된 접근으로 중복되는 코드를 자꾸 메인함수에 넣어서 코드가 길어지고 지저분해졌다. 경우의 수를 다 나눠서 그때마다 동적할당을 해줬는데, for문을 도는 동안 변수들을 초기화해주지 않아서(dir\_num같은 경우) 쓰레기값이 들어가 출력부분이 이상하게 된 부분을 고쳤다. option\_l함수의 경우 entry를 하나 인자로 받으면 파일 정보를 파악 후에 바로 출력해주는 구조라서, total block size를 구하려면 다시 함수가 필요했다. 인자로 여러 개의 디렉토리를 받을 때, chdir을 통해 디렉토리 간 이동이 필요한데 이전 경로를 저장하지 않고 chdir("..")을 했을 때, 오류가 났다. 따라서 prev\_dir을 도입하여 이전 경로를 저장하고, 디렉토리를 이동하는 일이 생기면 이전 디렉토리로 안전하게 되돌아갈 수 있도록 했다.

## VI. reference

생략