

시스템프로그래밍

assignment 2-1

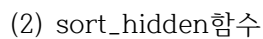
학번: 2021202045

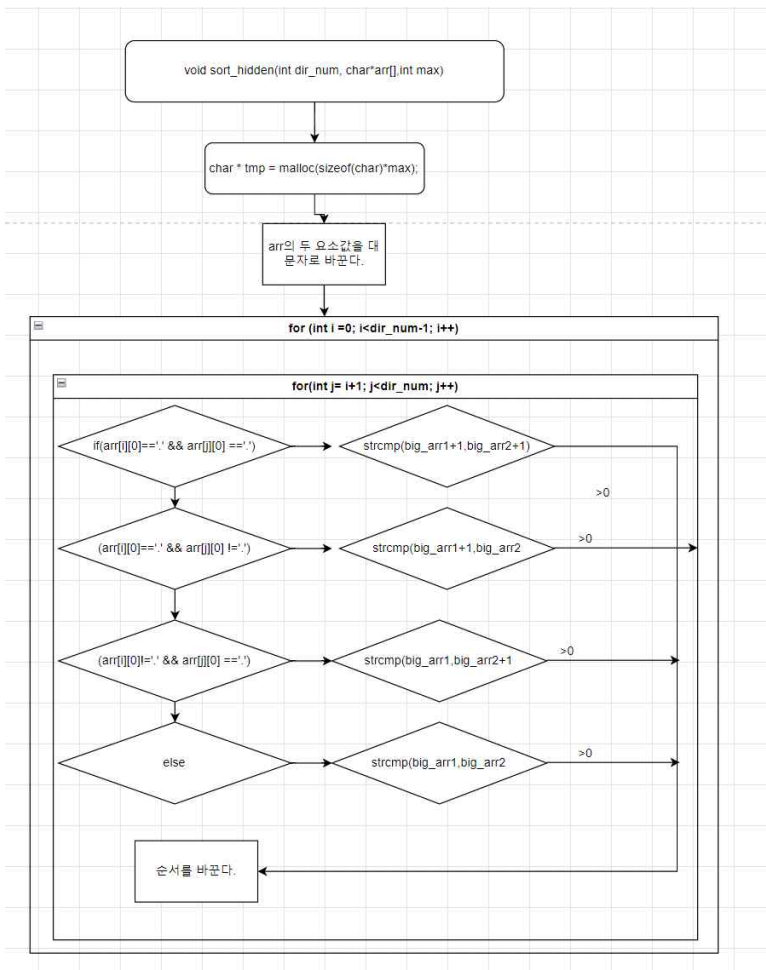
이름: 김예은

담당 교수님: 최상호 교수님

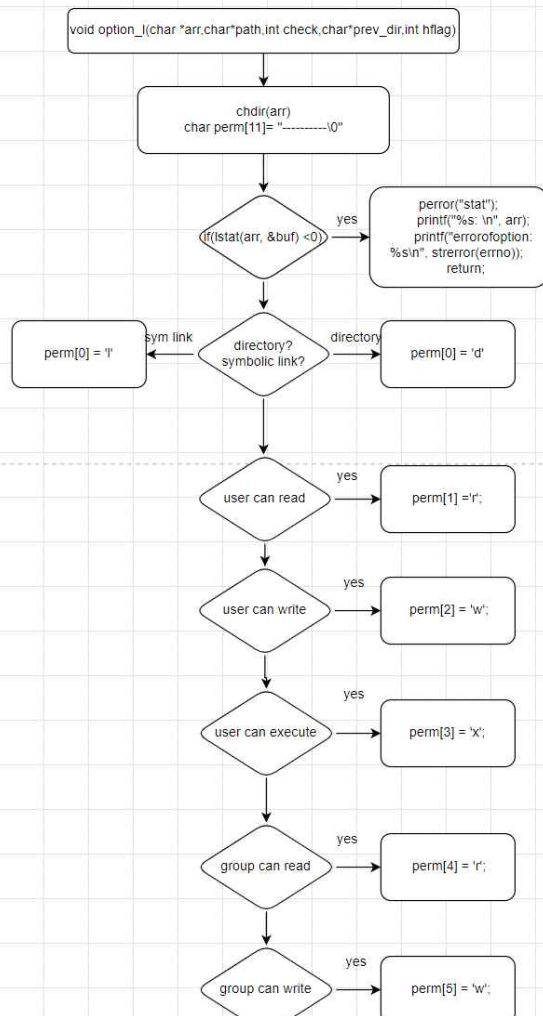
이번 과제는 여태까지 구현한 `final_ls` 코드를 `terminal`창이 아니라 `'html_ls.html'`파일에 `display`해주는 것이다. 이때, `command`창에 입력한 `command`와 현재 디렉토리 경로를 `head`부분에 출력해주고, 현재 경로를 `title`로 지정한다. `entry`들이 링크파일 일 때는 칸 및 글자 색을 초록색으로, 디렉토리일때는 파란색, 이 외 파일에는 빨간색으로 칠해준다. 각 파일의 이름에는 `hyper link`를 걸어주고, `..` 파일에는 부모 디렉토리에 해당하는 링크를 걸어준다. 이외 서식은 자유이다.

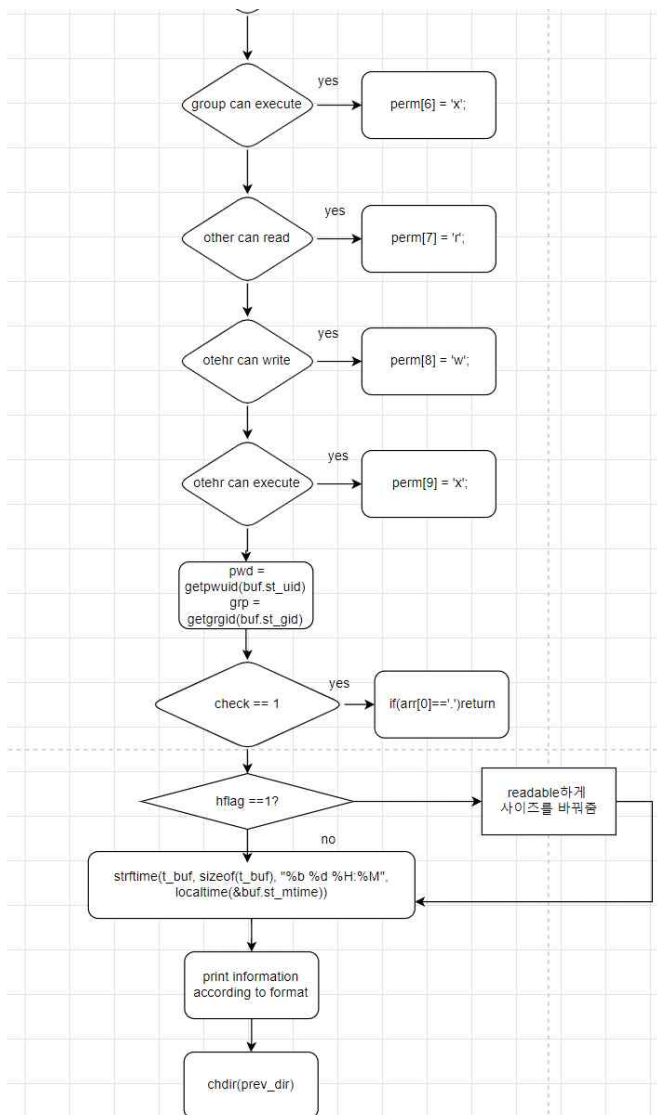
(1)ls_final함수에 대한 flow chart이다



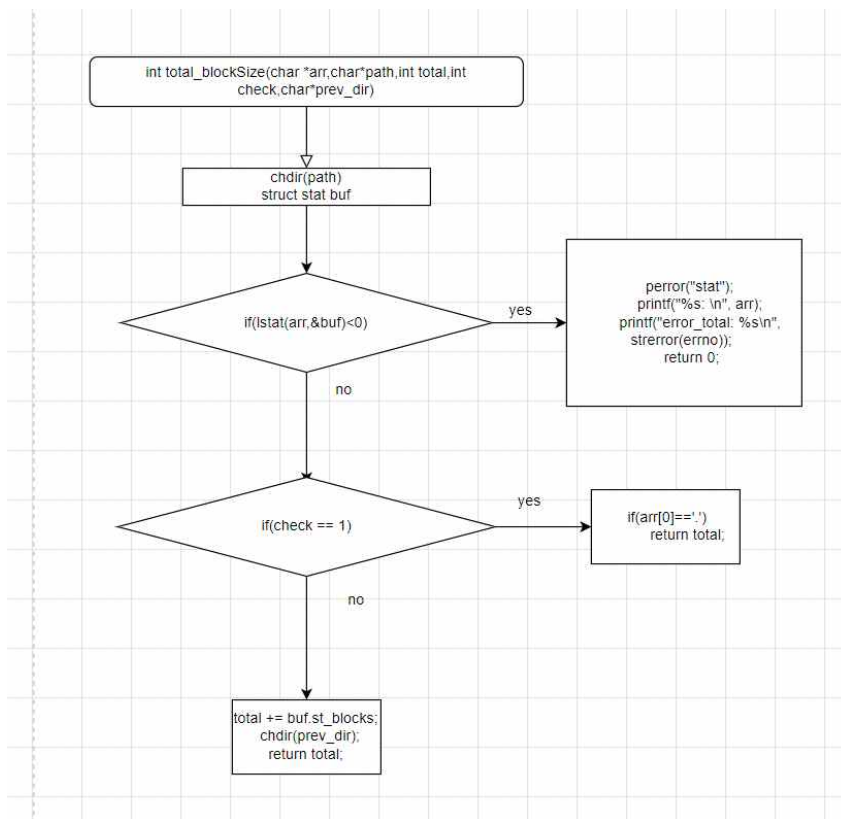


(3) option_l함수

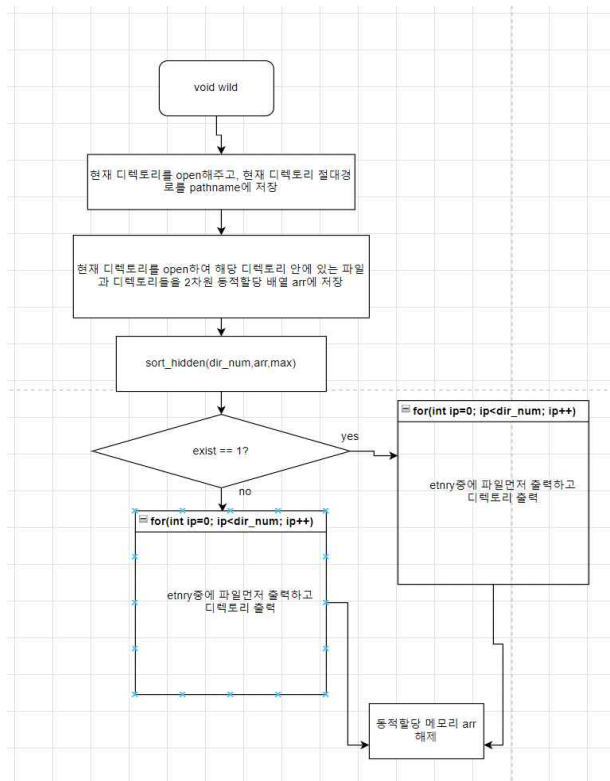




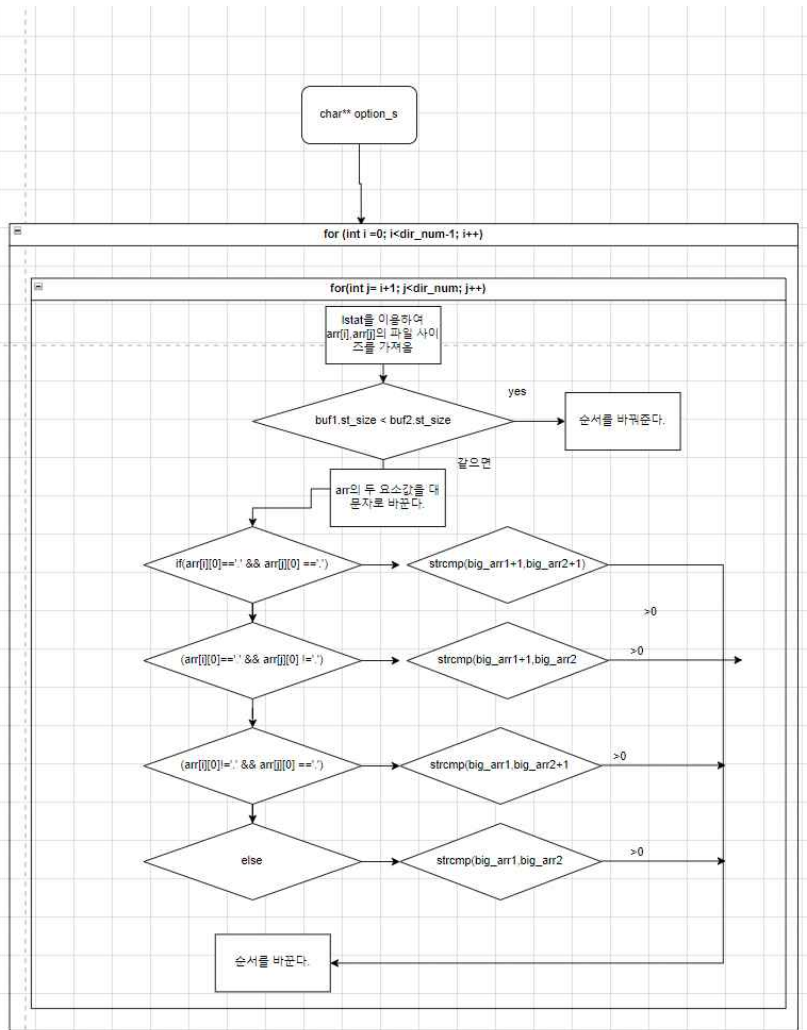
(4)total_blockSize함수



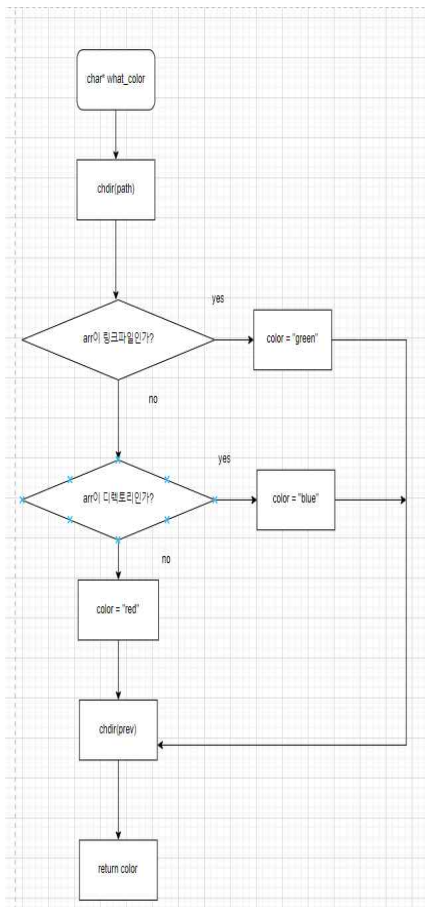
(5) wild 함수



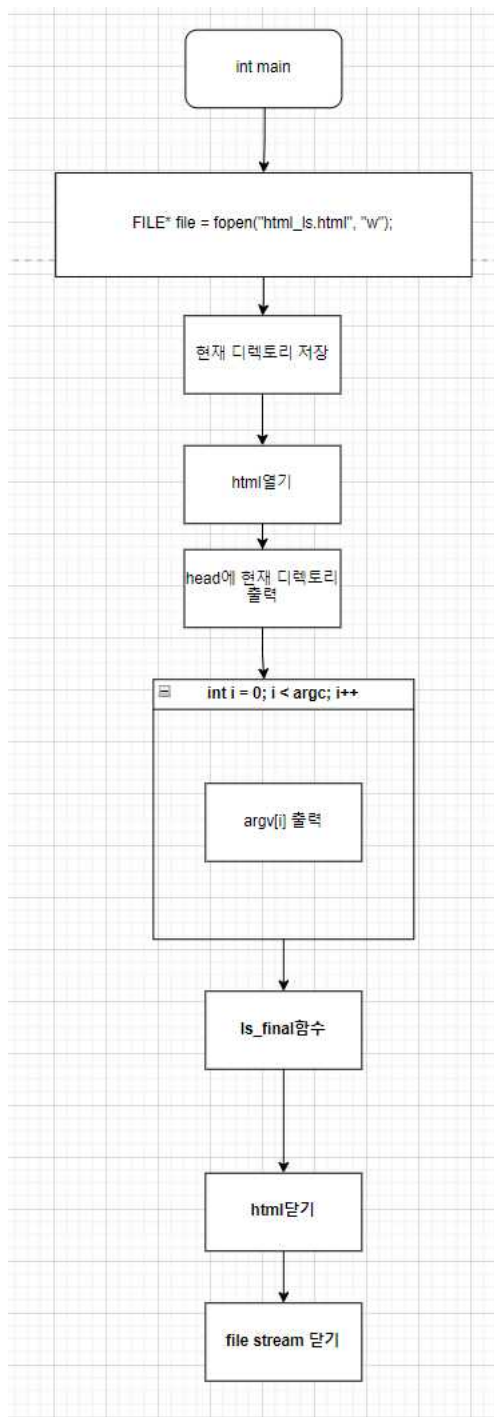
(6) option_s



(7) what_color함수



(8) main함수



III. Pseudo code

main함수

읽기 전용으로 "html_ls.html"파일을 연다.

현재 디렉토리의 path를 getcwd()

html을 열어준다.

current_path를 출력한다

title 또한 현재 디렉토리로 지정해준다.

i =0 부터 i<argc까지

argv[i]를 <h1>크기로 출력한다.

ls_final함수를 call한다.

html을 닫는다.

file stream을 닫는다.

what_color함수

인자로 받은 path로 이동

인자로 받은 arr에 대해 lstat구조체에 넣는다.

만약 arr이 링크파일이라면

color는 green

만약 arr이 directory라면

color는 blue

만약 그외라면

color는 red

인자로 받은 prev로 이동

color를 return

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 바꿔준후, format형으로 저장해준다.

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

hflag가 1이라면

파일 사이즈가 1024보다 작을 때,

파일 사이즈 출력

1024^2보다 작을 때

파일사이즈/1024.0을 올림하여 K붙여서 소수점 없이 출력

1024^3보다 작을 때

파일사이즈/1024.0^2을 올림하여 M붙여서 소수점 없이 출력

나머지

파일사이즈/1024.0^3을 올림하여 G붙여서 소수점 없이 출력

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

prev_dir로 되돌아간다.

option_s

인자로 받은 path로 이동한다.

i는 0부터 dir_num-1까지

j는 i+1부터 dir_num까지

arr[i]과 arr[j]의 파일 사이즈를 비교한다.

arr[i]의 사이즈 < arr[j] 사이즈

둘의 위치를 바꿔준다.

만약, 둘의 사이즈가 같다면

둘 모두 대문자로 바꿔준후,

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.

void wild

인자로 받은 path로 이동

해당 path를 opendir로 열어준 후, entry들을 arr에 저장한다.

check ==1이라면
모든 entry들을 반복문을 통해 돌며
만약 hidden file이라면 출력을 skip한다.
해당 entry가 opendir로 안열리면(directory가 아니라면)
출력한다.
해당 entry가 opendir로 열리면
출력한다.

check ==2이라면
모든 entry들을 반복문을 통해 돌며
만약 hidden file이라면 출력을 skip한다.
해당 entry가 opendir로 안열리면(directory가 아니라면)
출력한다.
해당 entry가 opendir로 열리면
출력한다.
arr 동적할당 해제
open한 directory닫기
이전 경로로 이동

void sort_hidden함수는 '.'으로 시작되는 파일도 포함하여 파일명을 정렬해주는 함수이다.
bubble sorting을 써줄 것이기 때문에 임시 저장해줄 array tmp를 동적할당 해준다.
해당 array는
array[i]와 array[j]를 모두 대문자로 바꿔준후 각각 big_arr1, big_arr2에 저장한다.
big_arr1[i]와 big_arr2[j]를 비교할 때
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.

ls_final함수
getopt를 통해 옵션에 a가 있다면 aflag를 1로 set, l이 있다면 lflag를 1로 set한다. S가 있
다면 sflag를, h가 있다면 hflag를, r이 있다면
rflag를 1로 바꾼다.
만약 argc ==1이거나, argc==2인데, lflag==0, aflag ==0일때,
argc가 1이 아니라면

인자에 wild card가 있는지 확인한다.

'/'로 시작하면 exist =1 , 현재디렉토리를 open한다.

아니라면 exist =2,

절대경로 중 wild card가 있는 부분 전까지 잘라서 path에 저장

나머지 뒷부분은 pattern array에 저장한다.

path를 open한다.

exist ==0이라면

인자값을 open한다. NULL값을 반환한다면,

현재 디렉토리를 연다.

open한 디렉토리에서 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(파일이름이 .으로 시작하면) print를 하지 않고, entry들을 정렬된 순서대로 출력한다.

exist가 1이라면,

arr을 처음부터 끝까지 돌면서 pattern과 매칭이 되는 entry라면

entry가 디렉토리가 아니라면

경로 출력과 wild함수로 간다.

arr을 처음부터 끝까지 돌면서 pattern과 매칭이 되는 entry라면

entry가 디렉토리라면

경로출력과 wild함수로 간다.

exist가 2라면,

arr을 처음부터 끝까지 돌면서 pattern과 매칭이 되는 entry라면

entry가 디렉토리가 아니라면

wild함수로 간다.

arr을 처음부터 끝까지 돌면서 pattern과 매칭이 되는 entry라면

entry가 디렉토리라면

wild함수로 간다.

동적할당해준 것을 모두 해제해준 다음, main함수를 끝낸다.

argc==2 && aflag ==1 && lflag ==0이라면,

open한 디렉토리에서 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.

만약 rflag ==1이라면,

sorting된 arr을 거꾸로 출력한다.

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

아니라면,

처음부터 끝까지 순서대로 arr을 출력한다.

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

동적할당해준 것을 모두 해제해준 다음, main함수를 끝낸다.

옵션이 1인데 argc==2일때 local_check =1, 옵션이 1a이면 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.

만약 hflag==0이라면,

total size를 출력한다.

만약 hflag ==1이라면,

total값이 1024보다 작으면

K와 함께 출력한다.

total값이 1024^2보다 작으면

total/1024.0을 소수점 없이 M과 출력

나머지라면

total/1024.0^2d을 소수점 없이 G와 출력

rflag가 1이라면

거꾸로 반복문을 돌며 모든 entry들을 option_l함수에 넣는다.

아니라면

반복문을 통해 모든 entry들을 option_l함수에 넣는다.

동적할당해준 것을 모두 해제해준 다음, main함수를 끝낸다.

옵션이 있고파 디렉토리, 파일 이름이 인자로 있으면

인자들을 파일인지 디렉토리인지 패턴인지 구분해서 opendir반환값에 따라(따로 정렬해주기 위해) 각각

for문을 통해 옵션다음부터 인자값 끝까지 돌면서 arr_file, arr_dir, arr_pat에 저장해준다.

연 directory는 다시 close해준다.

arr_file, arr_dir, arr_pat은 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 read하여 해당 디렉토리 내의 entry들을

arr이라는 2d array에 저장한다. arr의 요소들도 정렬해줘야하기 때문에, arr요소를 인자로 sort_hidden함수를 call 한다.

만약 rflag ==1이라면,

sorting된 arr을 거꾸로 출력한다.

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

아니라면,

처음부터 끝까지 순서대로 arr을 출력한다.

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

메모리 할당 해제 및 디렉토리를 닫아준다.

lfag==1 && rflag==0 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.

만약 hflag ==1이라면,

total값이 1024보다 작으면

K와 함께 출력한다.

total값이 1024^2보다 작으면

total/1024.0을 소수점 없이 M과 출력

나머지라면

total/1024.0^2d을 소수점 없이 G와 출력

만약, sflag==1이라면,

option_s함수로 들어간다.

반복문을 통해 모든 entry들을 option_l함수에 넣는다.

정렬된 entry들을 각각 option_l함수에 넣는다. 이때 옵션이 a가 있기 때문에 함수인자에 check부분에 0을 넣는다.

메모리 할당 해제 및 디렉토리를 닫아준다.

lfag==1 && rflag==1 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.

만약 hflag ==1이라면,

total값이 1024보다 작으면

K와 함께 출력한다.

total값이 1024^2보다 작으면

total/1024.0을 소수점 없이 M과 출력

나머지라면

total/1024.0^2d을 소수점 없이 G와 출력

만약, sflag==1이라면,

option_s함수로 들어간다.

거꾸로 반복문을 돌며 모든 entry들을 option_l함수에 넣는다.

정렬된 entry들을 각각 option_l함수에 넣는다. 이때 옵션이 a가 있기 때문에 함수인자에 check부분에 0을 넣는다.

메모리 할당 해제 및 디렉토리를 닫아준다.

옵션이 없다면,

pattern이 있을 경우,

해당 패턴이 '/'로 시작하면 exist =1, path는 해당 패턴으로 업데이트

해당 패턴이 '/'로 시작하지않다면, exist =2, 현재 디렉토리가 path, prev_dir이 된다.

pattern을 저장한다.

마지막 '/'가 있을 때까지 num을 업데이트 해주다가, 마지막 '/'를 만나면 path[num]='\0'

pattern이 있는 부분만 pattern에 update한다.

path를 opendir한다.

해당 directory를 읽으면서 entry들을 arr에 저장한다.

exist가 1이라면,

arr을 처음부터 끝까지 돌면서 pattern과 매칭이 되는 entry라면

entry가 디렉토리가 아니라면

경로 출력과 wild함수로 간다.

arr을 처음부터 끝까지 돌면서 pattern과 매칭이 되는 entry라면

entry가 디렉토리라면

경로출력과 wild함수로 간다.

exist가 2라면,

arr을 처음부터 끝까지 돌면서 pattern과 매칭이 되는 entry라면

entry가 디렉토리가 아니라면

wild함수로 간다.

arr을 처음부터 끝까지 돌면서 pattern과 매칭이 되는 entry라면

entry가 디렉토리라면

wild함수로 간다.

file의 경우 arr_file 처음부터 끝까지 for문을 돌면서

F_OK를 통해 존재하는 파일인지 아닌지 판단

만약 존재한다면,

파일을 출력한다.

directory의 경우 arr_dir 처음부터 끝까지 for문을 돌면서

해당 디렉토리를 open 및 read하여 해당 디렉토리 내의 entry들을 arr이라는 2d array에 저장.

arr요소를 인자로 sort_hidden함수를 call.

arr의 처음부터 끝까지

'.'으로 시작되는 entry를 만나면 continue

entry를 출력.

메모리 할당 해제 및 디렉토리를 닫아준다.

IV. 결과화면

결과화면 캡처 중에 이미 선택한 파일에 대해 보라색으로 뜬다.

(1)

```
./html_ls -lShra Web1_1 apple /home
```

Directory path: /home/kw2021202045/kw_hw/Web1_1

total :192K

Name	Permission	Link	Owner	Group	Size	Last Modified
2021202045_simple_ls.c	-rw-rw-r--	1	kw2021202045	kw2021202045	4K	Mar 26 15:20
..	drwxrwxr-x	8	kw2021202045	kw2021202045	4K	Apr 18 16:50
.	drwxrwxr-x	2	kw2021202045	kw2021202045	4K	Apr 07 07:00
Web1_1_B_2021202045.pdf	-rwxrw-rw-	1	kw2021202045	kw2021202045	179K	Mar 28 04:22

Directory path: /home/kw2021202045/kw_hw/apple

total :12K

Name	Permission	Link	Owner	Group	Size	Last Modified
test	-rw-rw-r--	1	kw2021202045	kw2021202045	0	Mar 24 09:20
good	drwxrwxr-x	2	kw2021202045	kw2021202045	4K	Mar 24 09:20
..	drwxrwxr-x	8	kw2021202045	kw2021202045	4K	Apr 18 16:50
.	drwxrwxr-x	3	kw2021202045	kw2021202045	4K	Apr 04 03:50

Directory path: /home

total :12K

Name	Permission	Link	Owner	Group	Size	Last Modified
kw2021202045	drwxr-xr-x	21	kw2021202045	kw2021202045	4K	Apr 18 16:45
..	drwxr-xr-x	24	root	root	4K	Mar 18 05:10
.	drwxr-xr-x	3	root	root	4K	Mar 16 03:24

(2)

./html_ls ?????? /home/kw2021202045/*

Directory path: /home/kw2021202045/examples.desktop

Directory path: /home/kw2021202045/Desktop

Name
Basic B 2021202045.pdf
Basic B 2021202045.tar.gz
kw_hello.c

Directory path: /home/kw2021202045/Documents

Name

Directory path: /home/kw2021202045/Downloads

Name
code 1.76.2-1678817801 amd64.deb

Directory path: /home/kw2021202045/kw_hw

Name
2021202045_html_ls.c
apple
Banana
c.html
html_ls
link_apple
Makefile
test2
Web1_1
Web1_3
zzz.txt
_pple

Directory path: /home/kw2021202045/Music

Name

Directory path: /home/kw2021202045/Pictures

Name

Directory path: /home/kw2021202045/Public

Name

Directory path: /home/kw2021202045/Templates

Name

Directory path: /home/kw2021202045/Videos

Name

Directory path: /home/kw2021202045/Videos

Name

Directory path: /home/kw2021202045/work

Name
2021202045
file1.txt
file2.txt
file3.txt
fileA.txt
fileC.txt
hello.txt
SP_lab

Directory path: /home/kw2021202045/kw_hw/test2

Directory path: /home/kw2021202045/kw_hw/apple

Name
good
test

Directory path: /home/kw2021202045/kw_hw/_pple

Name
bad
match.c
test4
test5




(3)

./html_ls -a

Directory path: /home/kw2021202045/kw_hw

Name
.
..
2021202045_html_ls.c
apple
Banana
c.html
html_ls
link_apple
Makefile
test2
.vscode
Web1_1
Web1_3
zzz.txt
_pple

(4) apple을 눌렀을 때, 해당 디렉토리로 link

Index of file:///home/kw2021202045/kw_hw/apple			
 Up to higher level directory			
Name	Size	Last Modified	
 good		3/24/23	9:20:39 AM PDT
 test		3/24/23	9:20:26 AM PDT

(5) ..을 눌렀을 때 부모 디렉토리로 이동

Index of file:///home/kw2021202045/

Up to higher level directory

Show hidden objects

Name	Size	Last Modified
.ICEauthority	1 KB	4/13/23 9:28:29 PM PDT
.Xauthority	1 KB	4/13/23 9:28:23 PM PDT
.bash_history	32 KB	4/13/23 9:25:19 PM PDT
.bash_logout	1 KB	3/16/23 3:24:38 AM PDT
.bashrc	4 KB	3/16/23 3:24:38 AM PDT
.cache		4/13/23 9:18:36 PM PDT
.config		4/11/23 11:52:28 PM PDT
.dmrc	1 KB	3/16/23 3:27:03 AM PDT
.dotnet		3/23/23 11:30:05 PM PDT
.gconf		3/16/23 3:27:32 AM PDT
.gnupg		4/13/23 9:28:26 PM PDT
.local		3/16/23 3:27:03 AM PDT
.mozilla		4/13/23 9:18:43 PM PDT
.pki		3/23/23 11:26:05 PM PDT
.profile	1 KB	3/16/23 3:24:38 AM PDT
.sudo_as_admin_successful		3/16/23 3:33:31 AM PDT
.viminfo	6 KB	4/18/23 4:45:07 PM PDT
.vscode		3/23/23 11:26:04 PM PDT
.xsession-errors	1 KB	4/13/23 9:28:23 PM PDT
.xsession-errors.old	2 KB	4/13/23 9:25:20 PM PDT
Desktop		4/15/23 3:17:10 AM PDT
Documents		3/16/23 3:27:03 AM PDT
Downloads		3/23/23 11:23:27 PM PDT
Music		3/16/23 3:27:03 AM PDT
Pictures		3/16/23 3:27:03 AM PDT
Public		3/16/23 3:27:03 AM PDT
Templates		3/16/23 3:27:03 AM PDT
Videos		3/16/23 3:27:03 AM PDT
examples.desktop	9 KB	3/16/23 3:24:38 AM PDT
kw_hw		4/18/23 4:50:13 PM PDT
work		3/19/23 10:50:58 PM PDT

(6)

./html_ls -l /home apple

Directory path: /home

total : 4

Name	Permission	Link	Owner	Group	Size	Last Modified
kw2021202045	drwxr-xr-x	21	kw2021202045	kw2021202045	4096	Apr 18 16:45

Directory path: /home/kw2021202045/kw_hw/apple

total : 4

Name	Permission	Link	Owner	Group	Size	Last Modified
good	drwxrwxr-x	2	kw2021202045	kw2021202045	4096	Mar 24 09:20
test	-rw-rw-r--	1	kw2021202045	kw2021202045	0	Mar 24 09:20

(7)kw2021202045를 눌렀을 때 해당 디렉토리의 entry들을 display

Index of file:///home/kw2021202045

Up to higher level directory
Show hidden objects

Name	Size	Last Modified
.ICEauthority	1 KB	4/13/23 9:28:29 PM PDT
.Xauthority	1 KB	4/13/23 9:28:23 PM PDT
.bash_history	32 KB	4/13/23 9:25:19 PM PDT
.bash_logout	1 KB	3/16/23 3:24:38 AM PDT
.bashrc	4 KB	3/16/23 3:24:38 AM PDT
.cache		4/13/23 9:18:36 PM PDT
.config		4/11/23 11:52:28 PM PDT
.dmrc	1 KB	3/16/23 3:27:03 AM PDT
.dotnet		3/23/23 11:30:05 PM PDT
.gconf		3/16/23 3:27:32 AM PDT
.gnupg		4/13/23 9:28:26 PM PDT
.local		3/16/23 3:27:03 AM PDT
.mozilla		4/13/23 9:18:43 PM PDT
.pki		3/23/23 11:26:05 PM PDT
.profile	1 KB	3/16/23 3:24:38 AM PDT
.sudo_as_admin_successful		3/16/23 3:33:31 AM PDT
.viminfo	6 KB	4/18/23 4:45:07 PM PDT
.vscode		3/23/23 11:26:04 PM PDT
.xsession-errors	1 KB	4/13/23 9:28:23 PM PDT
.xsession-errors.old	2 KB	4/13/23 9:25:20 PM PDT
Desktop		4/15/23 3:17:10 AM PDT
Documents		3/16/23 3:27:03 AM PDT
Downloads		3/23/23 11:23:27 PM PDT
Music		3/16/23 3:27:03 AM PDT
Pictures		3/16/23 3:27:03 AM PDT
Public		3/16/23 3:27:03 AM PDT
Templates		3/16/23 3:27:03 AM PDT
Videos		3/16/23 3:27:03 AM PDT
examples.desktop	9 KB	3/16/23 3:24:38 AM PDT
kw_hw		4/18/23 4:50:13 PM PDT
work		3/19/23 10:50:58 PM PDT

V. 고찰

assignment2-1을 구현하면서 저번 과제 구현에서 미흡했던 부분을 보충하였고, 각 파일들을 `lsstat`을 통해 색을 지정해줘야하므로 `what_color`라는 함수를 만들어줬다. 이번 과제에서 `html_ls.html`파일은 어느 옵션에서든 출력되지 않게 해주었다. `html_ls`를 실행할 때 해당 `html`파일이 새로 만들어지거나 새로 덮어씌워지는데 이때 `file size`나 `block`을 0으로 읽기 때문에 `total`계산이 `ls`와는 다르게 나온다. 출력에서 `html_ls.html`파일을 출력안해주므로, `total`에서도 `html`파일을 세주지 않는 코드로 짰다. `html`파일을 다 안 세주나 `check`해주기 위해, `c.html`을 만들어줬고 해당 파일은 정상적으로 파일사이즈나 블록사이즈를 측정해주는 것을 확인할 수 있었다.