

시스템프로그래밍실습

assignment 1-3

학 과: 컴퓨터정보공학부

담당교수: 이기훈 교수님

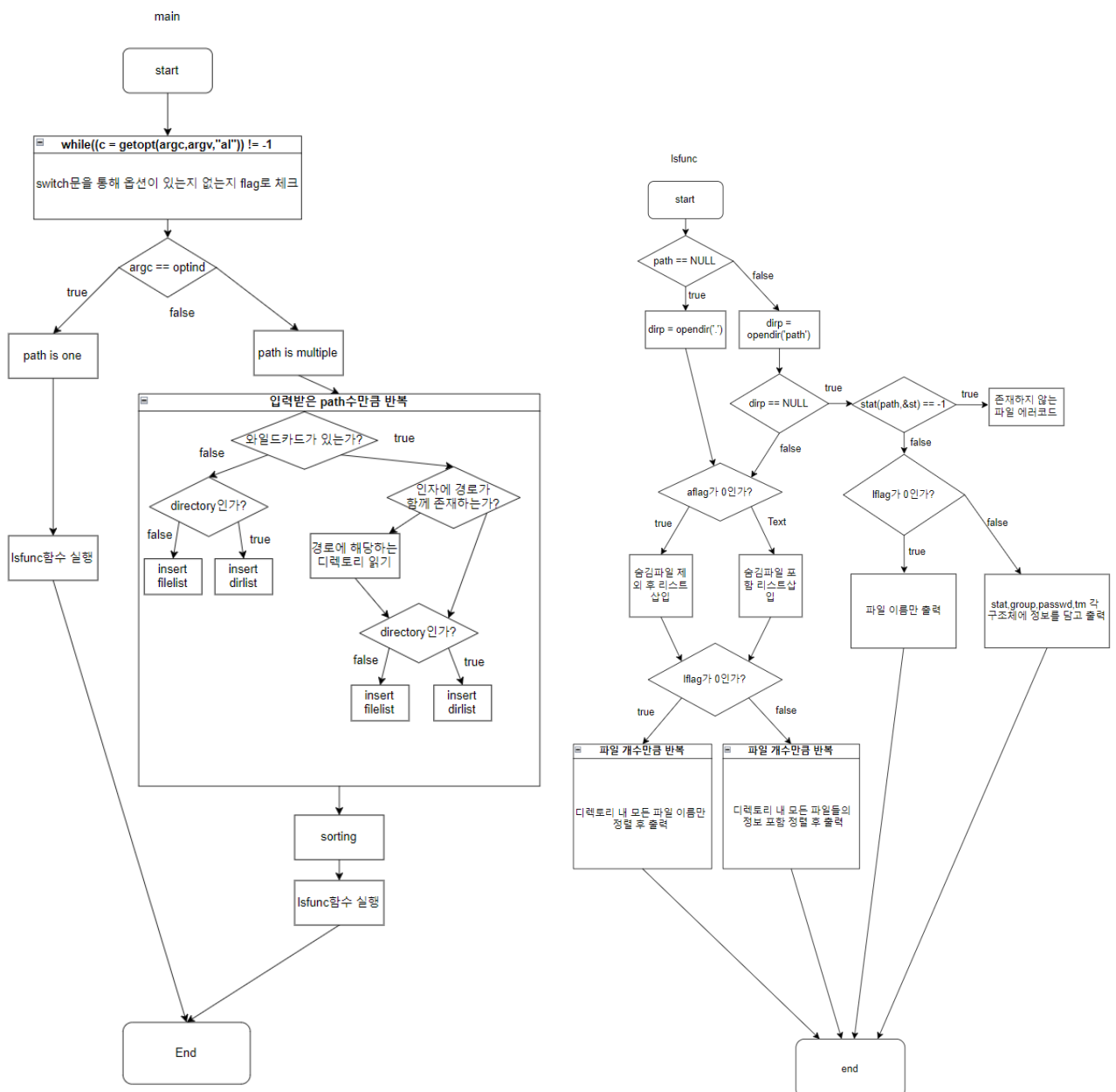
학 번: 2019202050

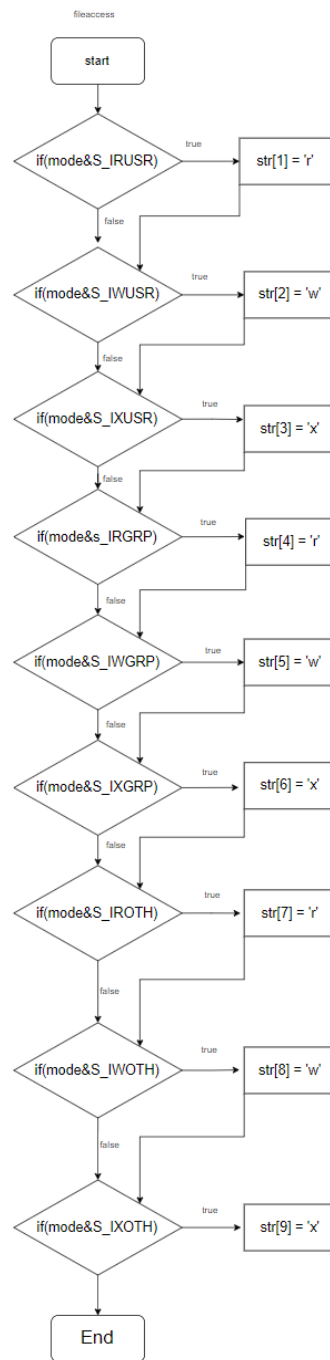
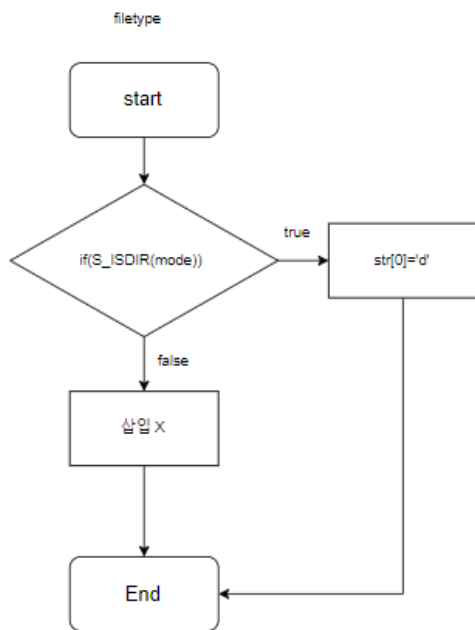
성 명: 이강현

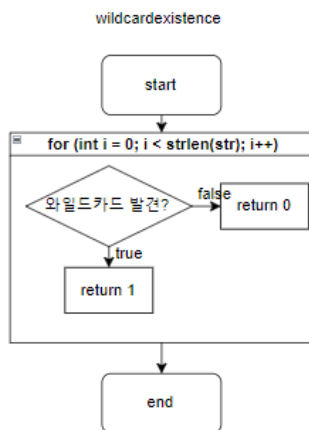
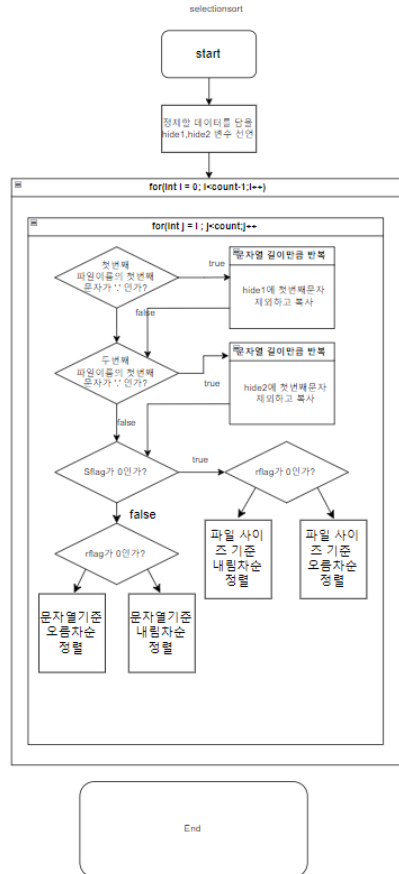
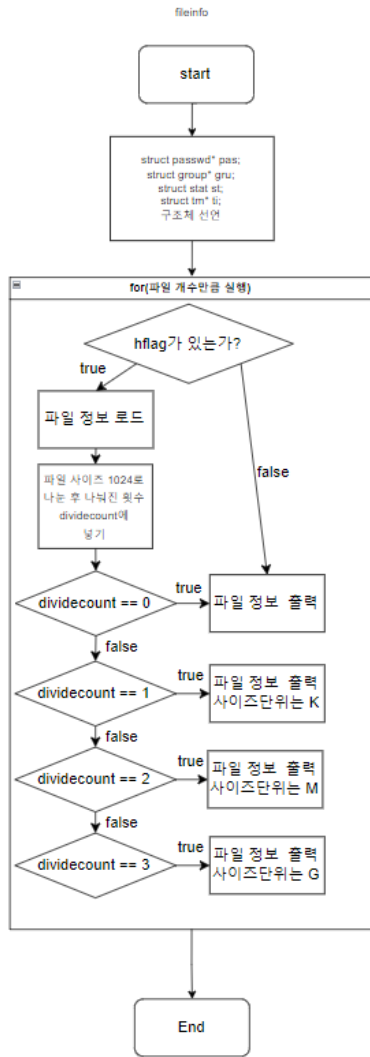
1. Introduction

이번 실습은 ls 명령어의 -h -S -r 옵션들을 추가적으로 구현해보고
추가적으로 wild card의 쓰임새와 이를 통해 ls 명령이 정상적으로
작동하도록 코드를 구현해보면서 패턴매칭 system call 함수인 fnmatch()의
사용법 또한 익힌다. 함수에서 플래그의 쓰임새를 익힌다.

2. Flow chart







모든 함수들의 흐름도이다. main 함수는 command 를 파싱하여 어떤 옵션이 들어왔는지 경로가 몇 개 들어왔는지를 체크하여 lsfunc 에 넘겨줄 인자와 lsfunc 이 실행되어야 하는 횟수를 결정한다.

이번 과제에서는 추가적으로 논옵션을 입력받았을 때 와일드카드가 있는지 없는지 체크한 후 있을 경우 또 경로가 포함되었을 때 아닐때로 나누고 디렉토리를 열었을 때 패턴이 매칭되면 저장 아니면 넘어가는 것으로 분기를 나누었다.

lsfunc 함수는 main 함수에서 넘겨받은 인자에 따라 파일인지 폴더인지를 체크한 후 분기를 나누고 각각 옵션에 따라 요구받은 행동을 취한다.

lsfunc 함수내에서 실행되는 fileaccess함수와 filetype함수는 l옵션에서 제일 먼저 출력되는 파일 타입과 접근권한을 숫자에서 문자데이터로 변환해주는 함수이다.

fileinfo 함수는 파일리스트와 파일 개수를 인자로 받아 파일 개수만큼 구조체에 파일정보를 저장하고 출력하는 것을 담당한다.

이번 과제에서는 h옵션이 추가되어 파일 사이즈를 단위를 붙여 출력하기 위한 분기가 추가되었다.

selectionsort함수는 선택정렬을 구현한 함수로 숨김파일을 구분하지 않고 정렬하기 위해 임시변수를 활용하여 구현하였다.

이번 과제에서는 사이즈기준으로 정렬하는 S옵션과 역순 출력을 나타내는 -r을 추가적으로 구현하여 파일 경로를 인자로 받아 파일 정보를 다시 로드하고 이를 통해 출력문을 구성하였다.

3. Pseudo Code

```
selectionsort
파일 사이즈 변수선언
파일 경로 만들기 위한 변수선언
문자열 정제 후 저장할 변수 선언

만약 입력 경로가 NULL이 아니라면
    for(파일 개수만큼)
        경로 생성하여 파일 사이즈를 얻어내고 변수에 저장
for(파일개수 -1만큼)
    for(파일개수 -1만큼)
        만약 파일의 앞 문자가 '.'이라면
            앞문자 제외하여 변수에 저장
        아니라면
            그대로 저장

소문자 모두 대문자로 변경

if(!sflag)
    if(!rflag)
        문자열 기준 오름차순 정렬
    else
        문자열 기준 내림차순 정렬
else
    if(!rflag)
        파일사이즈 기준 내림차순 정렬
        사이즈가 같다면 문자열 기준 정렬
    else
        문자열 기준 오름차순 정렬
        사이즈가 같다면 문자열 기준 정렬
```

```
fileinfo
구조체 선언
입력받은 경로로 파일 정보 얻어와 구조체에 담기
파일 정보를 문자열로 변환
if(hflag)
    파일 정보중 파일 사이즈 부분을 나누어 단위를 붙여줌
파일 정보 출력

wildcardexistence
wildcard = "*"?"[]"
if(와일드 카드를 입력받은 문자열에서 찾으면)
    return 1
못찾으면
    return 0
```

```
filetype
입력받은 파일의 mode가 디렉토리라면
    입력받은 문자열의 맨 앞 글자를 d로 변경

fileaccess
입력받은 파일의 mode의 user access가 read가 가능하다면
    입력받은 문자열의 2번째를 r로 변경
입력받은 파일의 mode의 user access가 write가 가능하다면
    입력받은 문자열의 3번째를 w로 변경
입력받은 파일의 mode의 user access가 execution이 가능하다면
    입력받은 문자열의 4번째를 x로 변경

입력받은 파일의 mode의 group access가 read가 가능하다면
    입력받은 문자열의 5번째를 r로 변경
입력받은 파일의 mode의 group access가 write가 가능하다면
    입력받은 문자열의 6번째를 w로 변경
입력받은 파일의 mode의 group access가 execution이 가능하다면
    입력받은 문자열의 7번째를 x로 변경

입력받은 파일의 mode의 other access가 read가 가능하다면
    입력받은 문자열의 8번째를 r로 변경
입력받은 파일의 mode의 other access가 write가 가능하다면
    입력받은 문자열의 9번째를 w로 변경
입력받은 파일의 mode의 other access가 execution이 가능하다면
    입력받은 문자열의 10번째를 x로 변경
```

```
lsfunc
입력받은 경로로 디렉토리 열기
디렉토리가 null이라면
    if(!lflag)
        파일이름만 출력
    else
        파일 정보까지 출력(fileinfo함수)
if(!aflag)
    숨김파일 제외 파일 저장

else
    모두 읽기

if(!lflag)
    정렬 하고 파일들 이름만 출력
else
    정렬하고 파일들 정보까지 출력

main
command parsing
if(입력받은 논옵션부분이 없다면)
    현재디렉토리를 경로로하여 lsfunc 호출
else
    for(논옵션 인자개수만큼 반복)
        if(wildcard가 있다면)
            if(wildcard와 경로가 함께 존재한다면)
                경로로 디렉토리를 열고 디렉토리내부 파일들중
                와일드카드패턴과 일치하는 파일들을
                디렉토리는 dirlist에 파일은 filelist에 저장
            else
                현재경로로 디렉토리를 열고 내부 파일들을
                와일드카드패턴과 일치하는 파일들을
                디렉토리는 dirlist에 파일은 filelist에 저장
        else
            파일은 filelist에 디렉토리는 dirlist에 저장

filelist와 dirlist 각각 정렬
출력
```

4. 결과화면

```
kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ls
2019202050_final_ls.c bca.txt final.txt good.txt spls_final text4.txt Web1_2_D_2019202050.pdf
Abc.txt empty.txt good.txt Makefile text1.txt text9.txt zoom.txt
kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ls -a
. .aaa.txt bca.txt final.txt Makefile text4.txt zoom.txt
.. Abc.txt .ccc.txt good.txt spls_final text9.txt
2019202050_final_ls.c .bbb.txt empty.txt good.txt text1.txt Web1_2_D_2019202050.pdf
kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ./spls_final -al
Directory path: /home/kw2019202050/Desktop/work/Web1_2_D_2019202050
total: 1052
drwxrwxr-x 2 kw2019202050 kw2019202050 4096 Apr 8 23:31 .
drwxrwxr-x 3 kw2019202050 kw2019202050 4096 Apr 8 5:26 ..
-rwxrwxr-x 1 kw2019202050 kw2019202050 23860 Apr 8 22:50 2019202050_final_ls.c
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 7 6:24 .aaa.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:47 Abc.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 7 6:24 .bbb.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:47 bca.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 7 6:24 .ccc.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:50 empty.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:50 final.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:48 good.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 75 Apr 8 23:31 Makefile
-rwxrwxr-x 1 kw2019202050 kw2019202050 22440 Apr 8 23:31 spls_final
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 4:54 text1.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 4:54 text4.txt
-rwxrwxr-x 1 kw2019202050 kw2019202050 1014421 Apr 5 7:10 Web1_2_D_2019202050.pdf
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:50 zoom.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 5:18 _good.txt
kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ./spls_final -ls
Directory path: /home/kw2019202050/Desktop/work/Web1_2_D_2019202050
total: 1044
-rwxrwxr-x 1 kw2019202050 kw2019202050 1014421 Apr 5 7:10 Web1_2_D_2019202050.pdf
-rwxrwxr-x 1 kw2019202050 kw2019202050 23860 Apr 8 22:50 2019202050_final_ls.c
-rwxrwxr-x 1 kw2019202050 kw2019202050 22440 Apr 8 23:31 spls_final
-rw-rw-r-- 1 kw2019202050 kw2019202050 75 Apr 8 23:31 Makefile
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:47 Abc.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:47 bca.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:50 empty.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:50 final.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:48 good.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 4:54 text1.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 4:54 text4.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 4:54 text9.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:50 zoom.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 5:18 _good.txt
kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$
```

위는 ls로 현재 있는 파일들을 확인하고 spls_final에 옵션을 넣어 잘 나오는지 확인해보는 코드이다. a와 l 옵션을 준 결과 숨김파일을 포함한 세부정보 목록들을 출력한다. l과 S 옵션을 주니 파일 세부정보를 파일 사이즈를 기준으로 정렬하여 보여주는 것을 확인할 수 있다.

```
kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ./spls_final -Shal
Directory path: /home/kw2019202050/Desktop/work/Web1_2_D_2019202050
total: 1052
-rwxrwxr-x 1 kw2019202050 kw2019202050 990.6K Apr 5 7:10 Web1_2_D_2019202050.pdf
-rwxrwxr-x 1 kw2019202050 kw2019202050 23.3K Apr 8 22:50 2019202050_final_ls.c
-rwxrwxr-x 1 kw2019202050 kw2019202050 21.9K Apr 8 23:31 spls_final
drwxrwxr-x 2 kw2019202050 kw2019202050 4.0K Apr 8 23:31 .
drwxrwxr-x 3 kw2019202050 kw2019202050 4.0K Apr 8 5:26 ..
-rw-rw-r-- 1 kw2019202050 kw2019202050 75 Apr 8 23:31 Makefile
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 7 6:24 .aaa.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:47 Abc.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 7 6:24 .bbb.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:47 bca.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 7 6:24 .ccc.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:50 empty.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:50 final.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:48 good.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 4:54 text1.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 4:54 text4.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 4:54 text9.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:50 zoom.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 5:18 _good.txt
```

위는 S h a l 옵션을 사용한 것이다. S 옵션이므로 일단 사이즈를 기준으로 정렬하며 h이므로 사이즈를 읽기 쉽게 단위를 붙여주었으며 a이므로 숨김파일이 같이 출력되는 것을 볼 수 있고 마지막으로 l 옵션으로 세부정보를 보여준다.

```

kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ls
2019202050_final_ls.c bca.txt final.txt good.txt spls_final text4.txt Web1_2_D_2019202050.pdf
Abc.txt empty.txt _good.txt Makefile text1.txt text9.txt zoom.txt
kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ls -a
. .aaa.txt bca.txt final.txt Makefile text4.txt zoom.txt
.. Abc.txt .ccc.txt _good.txt spls_final text9.txt
2019202050_final_ls.c .bbb.txt empty.txt good.txt text1.txt Web1_2_D_2019202050.pdf
kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ./spls_final -aLr
Directory path: /home/kw2019202050/Desktop/work/Web1_2_D_2019202050
total: 1052
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 5:18 _good.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:50 zoom.txt
-rwxrwxr-- 1 kw2019202050 kw2019202050 1014421 Apr 5 7:10 Web1_2_D_2019202050.pdf
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 4:54 text9.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 4:54 text4.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 4:54 text1.txt
-rwxrwxr-- 1 kw2019202050 kw2019202050 22440 Apr 8 23:31 spls_final
-rw-rw-r-- 1 kw2019202050 kw2019202050 75 Apr 8 23:31 Makefile
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:48 good.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:50 final.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:50 empty.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 7 6:24 .ccc.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:47 bca.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 7 6:24 .bbb.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:47 Abc.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 7 6:24 .aaa.txt
-rwxrwxr-- 1 kw2019202050 kw2019202050 23860 Apr 8 22:50 2019202050_final_ls.c
drwxrwxr-- 3 kw2019202050 kw2019202050 4096 Apr 8 5:26 ..
drwxrwxr-- 2 kw2019202050 kw2019202050 4096 Apr 8 23:31 .
kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ./spls_final -aSLr
Directory path: /home/kw2019202050/Desktop/work/Web1_2_D_2019202050
total: 1052
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 5:18 _good.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:50 zoom.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 4:54 text9.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 4:54 text4.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 8 4:54 text1.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:48 good.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:50 final.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:50 empty.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 7 6:24 .ccc.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:47 bca.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 7 6:24 .bbb.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Mar 31 2:47 Abc.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 0 Apr 7 6:24 .aaa.txt
-rw-rw-r-- 1 kw2019202050 kw2019202050 75 Apr 8 23:31 Makefile
drwxrwxr-- 3 kw2019202050 kw2019202050 4096 Apr 8 5:26 ..
drwxrwxr-- 2 kw2019202050 kw2019202050 4096 Apr 8 23:31 .
-rwxrwxr-- 1 kw2019202050 kw2019202050 22440 Apr 8 23:31 spls_final
-rwxrwxr-- 1 kw2019202050 kw2019202050 23860 Apr 8 22:50 2019202050_final_ls.c
-rw-rw-r-- 1 kw2019202050 kw2019202050 1014421 Apr 5 7:10 Web1_2_D_2019202050.pdf
kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$

```

이번엔 a이므로 숨김파일을 포함하여 l이므로 세부정보 포함, r이므로 문자열을 역순으로 출력하게 된다. 아스키코드 기준 문자열을 대소문자 없이 비교이고 소문자라면 대문자를 기준으로 하여 정렬하므로 위에서부터 아스키코드 표 기준 내림차순으로 정렬된다. 이때 _는 대문자와 소문자 사이에 위치하나 대문자를 기준으로 정렬되었기 때문에 결과적으로 제일 마지막에 위치하는 것을 알 수 있다.

S 옵션이 추가적으로 사용된 경우 사이즈를 역순으로 출력하고 사이즈가 같은 경우 문자열을 역순으로 출력한 것을 확인할 수 있다.


```

kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ./spls_final '/home/kw2019202050/*'
/home/kw2019202050/examples.desktop
Directory path: /home/kw2019202050/Desktop
work

Directory path: /home/kw2019202050/Documents

Directory path: /home/kw2019202050/Downloads

Directory path: /home/kw2019202050/Music

Directory path: /home/kw2019202050/Pictures

Directory path: /home/kw2019202050/Public

Directory path: /home/kw2019202050/Templates

Directory path: /home/kw2019202050/Videos

Directory path: /home/kw2019202050/work

kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ls /home/kw2019202050/*
/home/kw2019202050/examples.desktop
/home/kw2019202050/Desktop:
work

/home/kw2019202050/Documents:

/home/kw2019202050/Downloads:

/home/kw2019202050/Music:

/home/kw2019202050/Pictures:

/home/kw2019202050/Public:

/home/kw2019202050/Templates:

/home/kw2019202050/Videos:

/home/kw2019202050/work:
kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$

```

이번엔 와일드카드 부분이다. /home/kw2019202050/이라는 경로 속 파일들을 모두 출력하였다. ls와 비교하였을 때 동일한 목록을 확인할 수 있다.

```

kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ls -a
.      .aaa.txt  bca.txt  final.txt  Makefile  text4.txt      zoom.txt
..     Abc.txt   .ccc.txt  _good.txt  spls_final text9.txt
2019202050_final_ls.c .bbb.txt  empty.txt  good.txt   text1.txt  Web1_2_D_2019202050.pdf
kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ./spls_final 'text[1-4].txt'
./text1.txt
./text4.txt
kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ls text[1-4].txt
text1.txt text4.txt
kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ./spls_final 't??t?.txt'
./text1.txt
./text4.txt
./text9.txt
kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ls t??t?.txt
text1.txt text4.txt text9.txt

```

text[1-4].txt는 대괄호 속에 1부터 4까지의 범위내의 숫자가 오는 파일들을 출력하는 것인데 text1.txt와 text4.txt가 잘 출력되었다. t??t?.txt 는 물음표에 어떠한 하나의 문자가 대체되었을 때 일치하는 파일들을 출력하는데 text파일들이 모두 패턴에 매칭되어 출력되었다. ls와 같은 결과를 보이는 것을 알 수 있다.

```

kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ./spls_final ~ '*'
./2019202050_final_ls.c
./Abc.txt
./bca.txt
./empty.txt
./final.txt
./good.txt
./Makefile
./spls_final
./text1.txt
./text4.txt
./text9.txt
./Web1_2_D_2019202050.pdf
./zoom.txt
./_good.txt
Directory path: /home/kw2019202050
Desktop
Documents
Downloads
examples.desktop
Music
Pictures
Public
Templates
Videos
work

kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ls ~ *
2019202050_final_ls.c  bca.txt  final.txt  good.txt  spls_final  text4.txt  Web1_2_D_2019202050.pdf
Abc.txt               empty.txt  _good.txt  Makefile  text1.txt  text9.txt  zoom.txt

/home/kw2019202050:
Desktop Documents Downloads examples.desktop Music Pictures Public Templates Videos work
kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$

```

위는 ~ 홈 디렉토리와 현재디렉토리의 파일들이 출력되어야 한다. 정상적인 결과를 얻었다.

```

kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$ ./spls_final 'text.*' /home/kw2019202050/ wefs './*'
advance_ls: cannot access 'wefs' : No such directory or file
./2019202050_final_ls.c
./Abc.txt
./bca.txt
./empty.txt
./final.txt
./good.txt
./Makefile
./spls_final
./text1.txt
./text4.txt
./text9.txt
./Web1_2_D_2019202050.pdf
./zoom.txt
./_good.txt
Directory path: /home/kw2019202050
Desktop
Documents
Downloads
examples.desktop
Music
Pictures
Public
Templates
Videos
work

kw2019202050@ubuntu:~/Desktop/work/Web1_2_D_2019202050$

```

마지막으로 와일드카드가 파일이름에 있는 경우, 경로의 마지막에 온 경우, 존재하지 않는 파일, 와일드카드가 없는 경로를 섞어서 해본 결과이다. 파일들은 이렇게 경로가 많이 존재하는 경우 이름이 동일한 파일들을 구분하기 위해 경로와 파일을 이어 붙여 출력하였고 존재하지 않는 파일에 대한 출력문이 나타나는 것을 알 수 있다. 디렉토리가 출력되는 경우는 그 속에 있는 파일들도 잘 보여주는 것을 알 수 있다.

5. 고찰

이번 과제를 진행하면서 전반적인 코드의 흐름을 많이 바꾼 것 같다. 기존에 옵션을 인코딩하여 진행하였지만 이러한 방법은 옵션을 추가하게 되면 너무 많은 경우의 수가 나오게 되어 flag를 직접 넘겨주는 방식으로 바꾸었다. 이렇게 하니 정렬함수에서는 s 옵션, r 옵션의 유무가 중요하고 파일 정보를 출력하는 함수에서는 h 옵션이 중요한데 이를 넘겨주기에 용이했다. 특히나 어려웠던 점은 와일드카드의 구현이었는데 우선 와일드카드를 판단하는 것이 어려웠고 파일을 출력할 때 파일 먼저 출력하면서 모두 정렬되어야 한다는 점이 어려웠다. 인자를 넣어주는 순서에 상관없이 정렬되어야 한다는 조건이 기존에 구현해뒀던 코드와 맞지 않았기 때문이다. 또한 경로로 디렉토리를 열었을 때 해당 경로와 파일이름을 concatenate해주어서 stat의 경로에 넣어주어야 정상적으로 파일의 정보를 읽어온다는 것을 알았다. 단순히 이름만 넘겨주게 되면 파일정보를 온전히 읽지 못하였다. 이번 과제로 옵션을 추가하게 될 때 어떤 식으로 코드를 구현하여야 유지보수가 쉬운지를 좀 많이 배우고 더 깔끔하게 짜려고 노력했던 것 같다.

6. Reference

시스템프로그래밍실습 강의자료 참조