System programming

Assignment 2-2 Advanced_ls

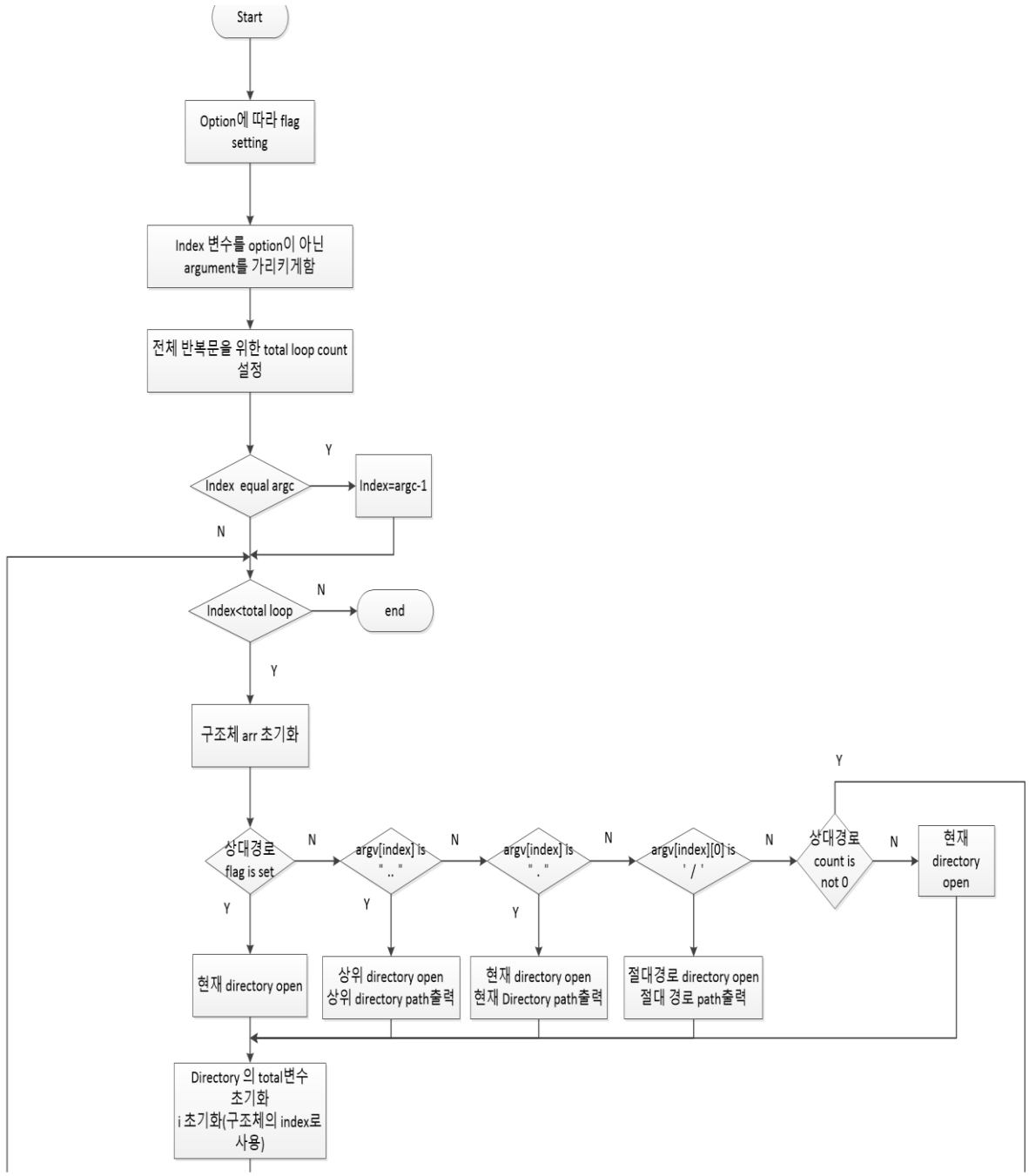
Professor	이기훈 교수님
Department	Computer engineering
Student ID	2014722046
Name	유지현
Class	설계(화6목5) /실습(목34)
Date	2016. 4. 8

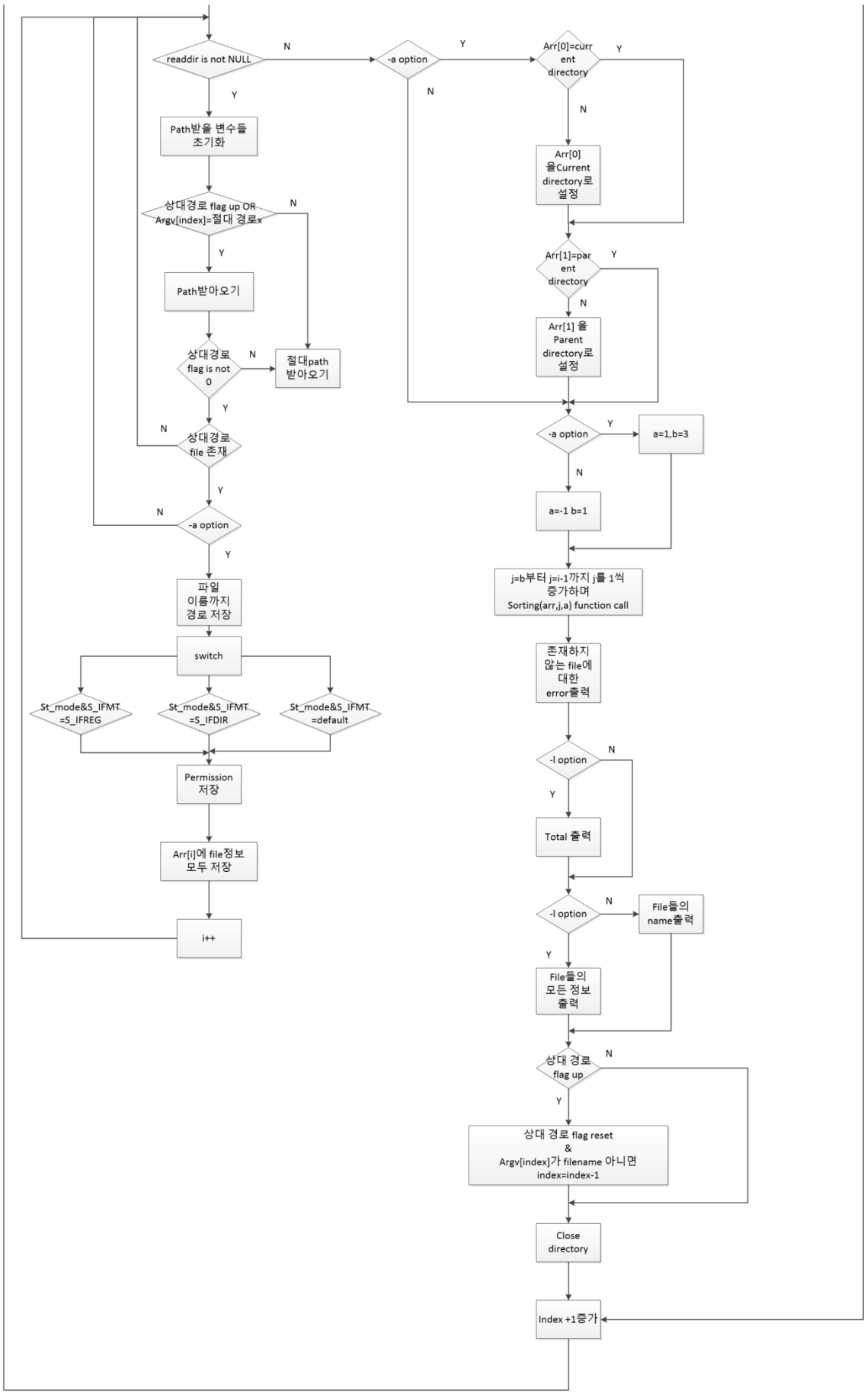
1.Instroduction

이번 과제는 지난 과제에서 만든 simple ls에 -a와 -l option 기능을 추가하는 것이다. 실제 ls 기능의 option처럼 -a일 때 hidden file을 출력시켜주고 -l일때 file 정보를 출력시켜준다. 그리고 절대 경로 혹은 상대 경로도 넣어주었을 때 그 경로에 대한 ls 혹은 ls option에 따라 구현해야 한다.

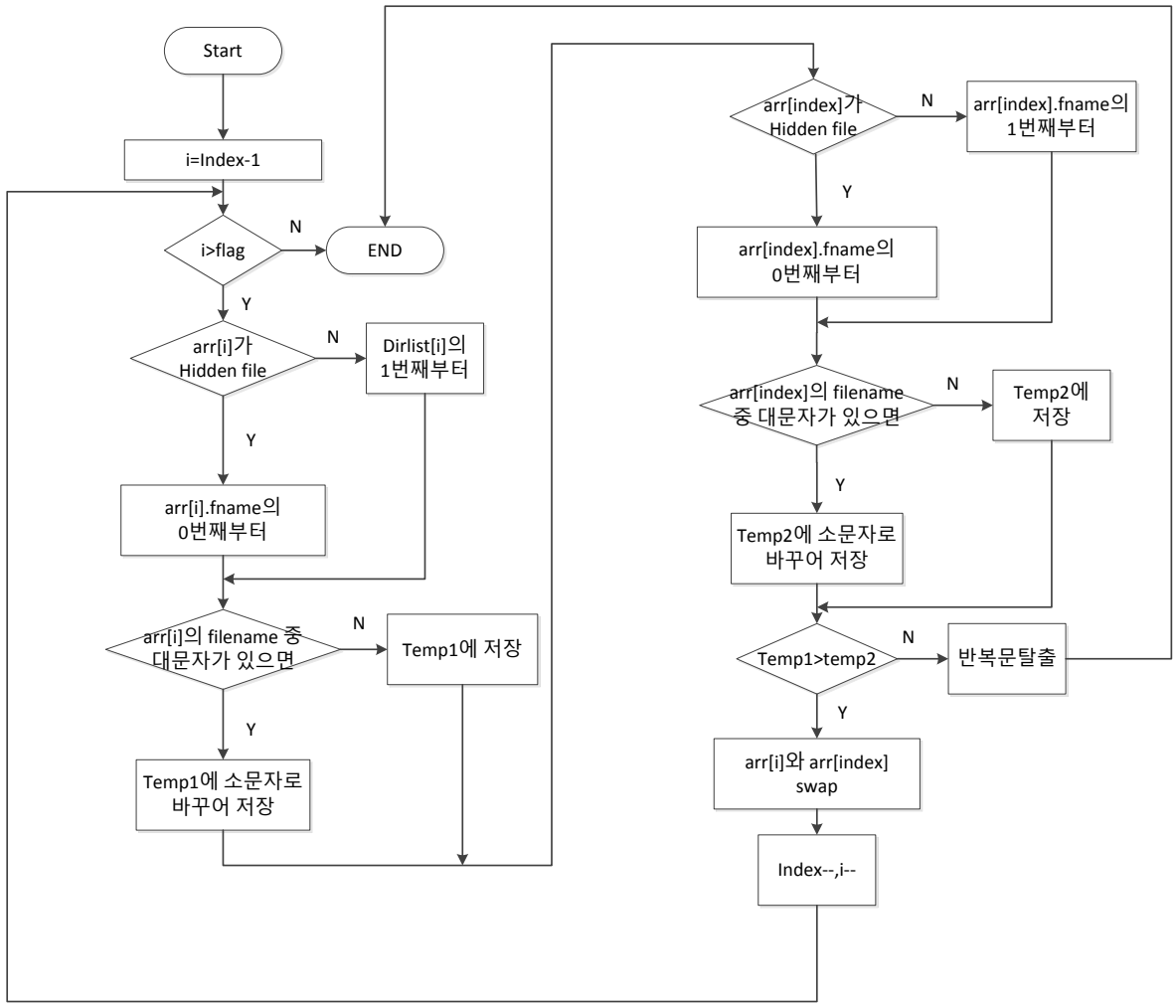
2.Flow chart

■main




readdir is not NULL
N
Y
-a option
Y
N
Arr[0]=current directory
Y
N
Path받을 변수들 초기화
Arr[0] 을Current directory로 설정
상대경로 flag up OR Argv[index]=절대 경로x
N
Y
Arr[1]=parent directory
Y
N
Path받아오기
Arr[1] 을 Parent directory로 설정
상대경로 flag is not 0
N
절대path 받아오기
Y
-a option
Y
a=1,b=3
N
상대경로 file 존재
N
Y
a=-1 b=1
-a option
N
Y
j=b부터 j=i-1까지 j를 1씩 증가하며 Sorting(arr,j,a) function call
파일 이름까지 경로 저장
switch
존재하지 않는 file에 대한 error출력
St_mode&S_IFMT =S_IFREG
St_mode&S_IFMT =S_IFDIR
St_mode&S_IFMT =default
-l option
N
Y
Permission 저장
Total 출력
Arr[i]에 file정보 모두 저장
-l option
N
File들의 name출력
Y
i++
File들의 모든 정보 출력
상대 경로 flag up
N
Y
상대 경로 flag reset
&
Argv[index]가 filename 아니면
index=index-1
Close directory
Index +1증가

■sorting



3.Pseudo code

■main function

it main(int argc, char ** argv)

index reset 0

pa_count reset 0;

while((c=getopt(argc,argv,"al"))! is not 1)

{

switch by c

{

if a option

aflag set;

exit

if l option

```
                    lflag set
                    exit
            }


}

for(index=1;index<argc; index++)
 {
            if path exist
            exit

 }


for( a=0,b=0; index<argc; index++)
{
            if absolute path
            {
            ab_flag set;
            path copy at ab_path[a]
            a is increased to a+1
            }
            if '.' / '..'
            pa_count   is increased to pa_count+1
            relative path
            {
            r_flag set;
            path copy at r_path[b]
            b is increased to b+1
            }
}
ab_count is set by a
r_count is set by b

i reset

for(index=1;index<argc; index++)
 {
```

```
            if argv is not "-a" or "-l" or "-al" or "-la"
            exit

 }
total_loop is set by index+ab_count+r_count+pa_count


if   not path, only include option in argv
{
            index is set by argc-1
}


for(; index<total_loop; index++)
{

for(a=0;a<1000; a++)                    //initialize
{

reset fname
reset permission
reset u_ID
reset g_ID
reset linkcounter
reset CAPACITY
reset month
reset day
reset hour
reset min

}


c reset
if relative path input exist
{
 open current dir
```

```
}
if parent directory
{
open parent directory
path receive current working directory
            for(a=0; path[a]!='₩0'; a++)
            {
            if path[a] is '/'
            i is set by a

            }
print "Directory path:"
print parent path



}
if current directory
{
open current directory
path receive current working directory
print   Directory path
}
if   absolute path
{
open absolute path directory
            if absolute path exist
            print Directory path
            if absolute path does not exist
            {
            error message print
            continue
            }
}
if already all relative path handled at first
 continue

if     any path does not exist
{
open current directory
```

```
}
i   reset
total reset


while((dir=readdir(dirp))!=NULL){


        c and a reset
        reset info
        reset path

        if not absolute path
        {
        path receive   current working directory
        store path string in info
        info is set by info + "/"

        if relative path first handle
        {
                for(a=0; a<r_count; a++)
                {
                        if check file exist
                        {
                          exit
                        }

                }

        }

        }

        if absolute path
        {

        store argv[index] string in path
        store path string in info
        info is set by info + "/"
```

```
}


if relative path file not exist
        continue
if -a option not exist
        continue
info is set by info + dir->d_name



store st_info in info

switch by (fstat.st_mode)&(S_IFMT)
{
if file
        arr[i].permission[c++] store   '-'
        exit

if directory
        arr[i].permission[c++] store 'd'
        exit
if default
        exit
}


for(k=0; k<3; k++)
{
        if read enable
        arr[i].permission[c++] store 'r';
        if read not enable
        arr[i].permission[c++] store '-';
        if write enable
        arr[i].permission[c++] store 'w';
        if write not enable
        arr[i].permission[c++] store '-';
        if execution enable
        arr[i].permission[c++] store 'x';
```

```
                    if execution not enable
                    arr[i].permission[c++] store '-';
            }
            arr[i].permission[c] store NULL
            arr[i].linkcounter store linkcounter
            arr[i].CAPACITY store st_size
            arr[i].u_ID store u_ID
            arr[i].g_ID store g_ID
            arr[i].mon store month
            arr[i].day store day
            arr[i].hour store hour
            arr[i].min store min
            total is increased to total +fstat.st_block

            for(k=0; dir->d_name[k]!='₩0'; k++)
              arr[i].fname[k] is set by dir->d_name[k]



            arr[i].fname[k] is set by NULL
            i is increased to i+1
}

b reset

for( j=1; aflag!=0 && j<i; j++)
{
    if '.' is not array first
       {
           flag up
           exit
       }
}


if flag up
{
```

```
store arr[j]'s all components at temp variables
store arr[0]'s all components at arr[j]'s all components
store temp variable at arr[0]'s all components


}


a and b reset

for(j=2; aflag!=0 && j<i; j++)
{
    if '..' is not array second
       {
           flag up
           exit
       }
}


if flag up
{
 store arr[j]'s all components at temp variables
 store arr[1]'s all components at arr[j]'s all components
 store temp variables at arr[1]'s all components

}

if -a option
{
   a is set by 1, b is set by 3
}
if not -a option
{
   a is set by -1 b is set by1;
}


for(j=b; j<i; j++)
```

```
sorting(arr,j,a) function

if -l option
printf total


for(   a=0; a<i; a++)                    {

if l flag up
{
print arr[a].permission,linkcounter,u_ID,g_ID,CAPACITY,month,day,hour,min

}
print arr[a].fname




}
if r_flag is set
        {

        r_flag reset
        if now first r_flag handle
                index is set by index-1;
        }

close directory

}
```

■sort function(Insertion sort)

```
void sorting function( struct mynode arr[1000], int idx, int flag)
for(i=idx-1; i>flag ;)
        {
         k, j variable initialize 0
        if arr[i] is hidden file
           start_point is set by 1;
        if arr[i] isn't hidden file
```

```
   start_point is set by 0;

for(k=start_point; dirlist[i][k]!='₩0'; k++)
{
   if arr[i].fname[k] is capital
   arr[i].fname[k] change to small letter for comparing at temp1[j]
   if arr[i].fname[k] is not capital
     arr[i].fname[k] is stored at temp1[j]
   j is increased to j+1
}

 if arr[index] is hidden file
    start_point is set by1;
 if arr[index] is not hidden file
    start_point is set by 0;

j is set by 0

for(k=start_point; arr[idx][k]!='₩0'; k++)
{
 if arr[idx][k] is capital
   arr[idx][k] change to small letter for comparing at temp2[j]
 if arr[idx][k] is not capital
   arr[idx][k] is stored at temp2[j]
 j is increased to j+1
}



if temp1 is larger than temp2
{

   store arr[idx]'s all components at temp variables

k is set by idx
 store arr[k-1]'s all components at arr[k]

store temp variables at arr[k-1]'s all components

idx is decreased by idx-1
```

```
i is decreased by i-1
}

if temp1 is not larger than temp2
  exit loop
}
```

4.Conclusion

이번 과제를 구현하면서 어려웠던 점은 상대경로랑 절대경로를 섞어서 출력하는 것이었습니다. 상대경로를 넣고 그 다음에 절대경로를 넣으면 절대경로의 정보를 제대로 받아오지 못했습니다. 알고 보니 이 문제는 상대경로를 이미 다 출력했는데 또 상대경로를 잘못 읽어와 아예 절대경로의 정보를 받아오지 못했던 것이어서 조건문을 추가하여 해결하였습니다. 그리고 잘못된 파일 경로가 들어왔을 때 처음엔 segmentation fault가 떠서 이는 dir이 NULL일 경우 예외처리를 해서 해결해주었습니다.