

시스템프로그래밍1

#Assignment2-1

이름: 이진수

학번: 2015722013

학과: 컴퓨터공학과

김태석 교수님

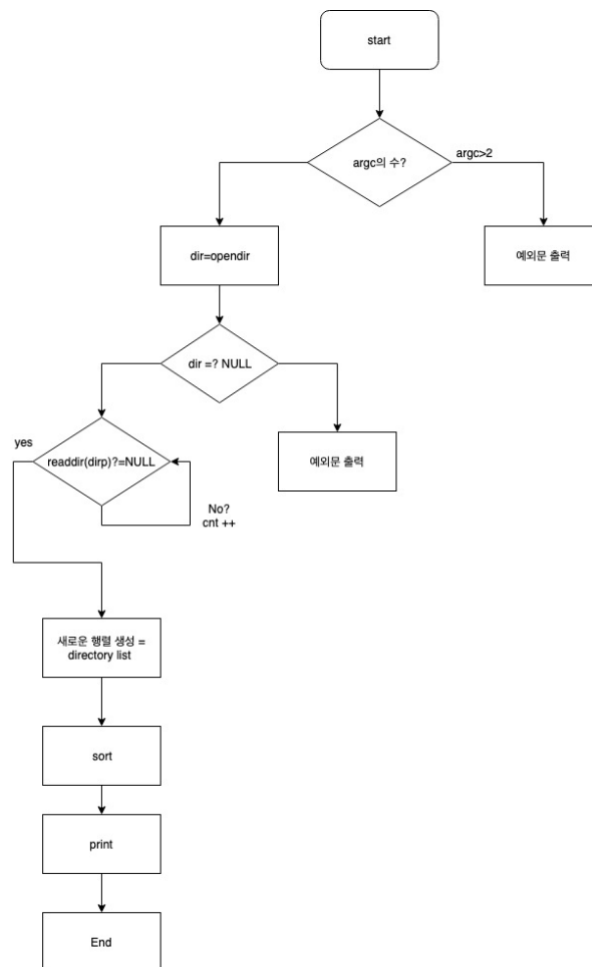
분반: 화,목 5,6교시

화요일 실습

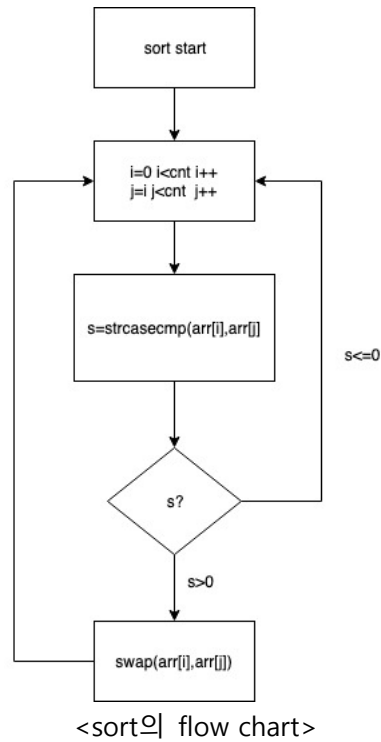
1. Introduction

이번 과제는 리눅스에서 이미 구현되어 있는 함수인 ls를 simple ls로 만드는 것이다. ls는 디렉토리 안에 있는 파일들 (숨겨져 있는 파일들 제외하고) 모두 보여준다. 이번에 만드는 ls는 파일의 이름을 단순히 보여주는거 외적으로도 파일의 이름을 오름차순으로 정렬을 해준다.

2. Flow Chart



<전체적인 flow chart>



3. Pseudo Code

```

main(int argc, char **argv){
    if(argc == 1){
        opendir(.);
        while(readdir(.)!=null)
            cnt++;
    }
    else if(argc >2){
        printf(예외문 처리);
    }
    else {
        if(dirp == null)
        { printf(예외문 처리);
          k++;}
        else {
            while(readdir(dirp)!=NULL){
                cnt++
            }
        }
    }

    If(argc<3 && k==0){
        arr[cnt][256];
    }
}

```

```

while(readdir(dirp)!=null){
    arr[tmp]=d_name;
    tmp++;
}
for(i=0; i<cnt; i++){
    for(j=0; j<cnt; j++){
        s=strcasecmp(arr[i],arr[j]);
        if(s>0)
            swap(arr[i], arr[j]);

        printf(arr[i]);
    }
}

```

4. Result

```

[sp2015722013@jinsulee-VirtualBox:~$ cd work
[sp2015722013@jinsulee-VirtualBox:~/work$ cd code
[sp2015722013@jinsulee-VirtualBox:~/work/code$ vim ls.c
[sp2015722013@jinsulee-VirtualBox:~/work/code$ vim ls.c
[sp2015722013@jinsulee-VirtualBox:~/work/code$ vim Makefile
[sp2015722013@jinsulee-VirtualBox:~/work/code$ make
gcc -c ls.c
gcc -o simple_ls ls.o
[sp2015722013@jinsulee-VirtualBox:~/work/code$ ./simple_ls
a
a.out
A.txt
db
ls.c
ls.o
ls1.c
Makefile
one
one.txt
prac.c
run
simple_ls
spls
two
[sp2015722013@jinsulee-VirtualBox:~/work/code$ ./simple_ls one
one.txt
three.txt
two.txt
[sp2015722013@jinsulee-VirtualBox:~/work/code$ █

```

<현재 디렉토리의 파일을 나열, 현재 디렉토리 내부 one이라는 디렉토리 내부의 파일 나열>

```

sp2015722013@jinsulee-VirtualBox:~/work/code$ ./simple_ls ~/
Desktop
Documents
Downloads
examples.desktop
Music
Pictures
prac
Public
Templates
Videos
work
sp2015722013@jinsulee-VirtualBox:~/work/code$ ./simple_ls ~/Documents/
cli
cli.c
fileA.txt
fileB.txt
mf
one
prac
srv
srv.c
three
two
sp2015722013@jinsulee-VirtualBox:~/work/code$

```

<path를 통해서도 출력이 가능하다.>

```

sp2015722013@jinsulee-VirtualBox:~/work/code$ ./simple_ls ls.c
'ls.c': No such directory
sp2015722013@jinsulee-VirtualBox:~/work/code$ ./simple_ls Not_file
'Not_file': No such directory
sp2015722013@jinsulee-VirtualBox:~/work/code$ ./simple_ls one two
Only one directory path can be processed
sp2015722013@jinsulee-VirtualBox:~/work/code$

```

<예외문 처리들>

처음은 directory가 아닌 파일을 출력 하려고 해서 예외문 처리를 해 줬다.
두번째는 현 directory 안에 없는 파일을 출력하려고 해서 예외문 처리를 해주었다.
세번째는 출력하려는 directory가 두개 이상이여서 예외문 처리 해주었다.

5. Conclusion

이번 과제를 통해서 리눅스 상의 command의 대략의 흐름에 대해 알게 되었다. Simple ls를 만들때 가장 힘들었던 점은 정렬을 할 때 과연 어떤 logic을 써야 할지에 대한 고민이었다. 생각 했었던 로직이 총 두개가 있는데 하나는 이번 코드에서 사용한 새로운 행렬을 동적할당으로 생성하여 directory에 있는 파일 이름들을 새로운 행렬에 저장 한 다음 행렬을 정렬 하여 출력 하는 방식이 었다. 다른 하나의 방식은 linked list를 사용하여 파일 이름을 하나씩 linked list에 삽입 하면서 정렬도 같이 해주는 방식이 었다. 현재 이 코드에서는 둘의 효율성 차이가 별로 없어 새로운 행렬을 동적할당으로 받아 오는 형식을 취했다. 하지만 만약 두개의 directroy를 출력하는 프로그램을 짤때에는 필연적으로 linked list를 사용해야 겠다는 생각을 하였다.