

# 시스템프로그래밍실습

## assignment 1-1

학 과: 컴퓨터정보공학부

담당교수: 이기훈 교수님

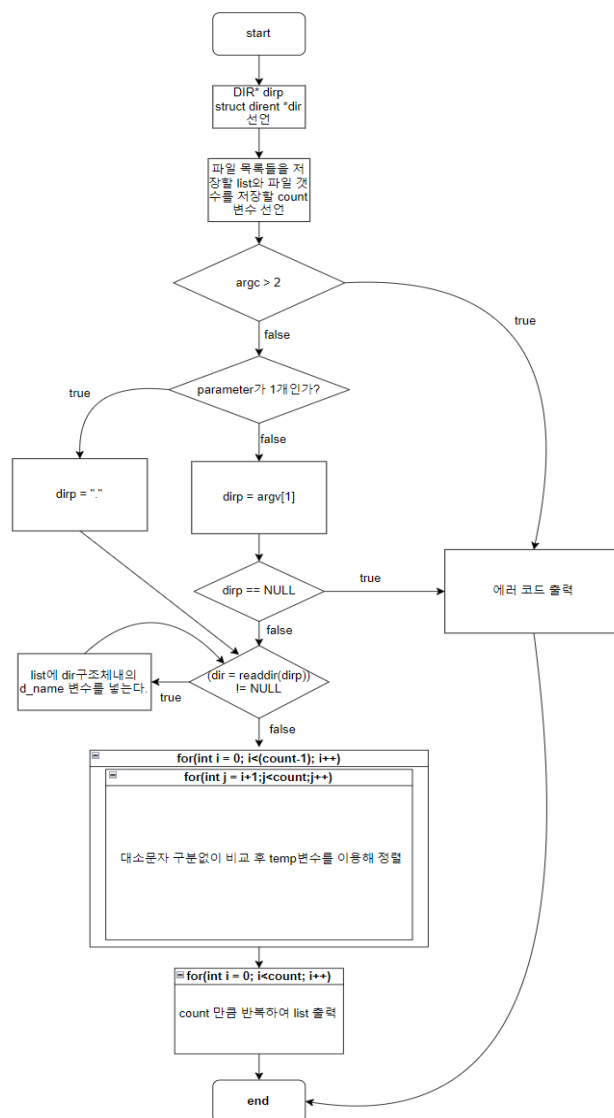
학 번: 2019202050

성 명: 이강현

## 1. Introduction

숨김 파일을 출력하지 않고 정렬조건을 충족하는 간단한 ls 명령어의 역할을 하는 파일을 작성해본다. Data types 중 DIR 과 dirent 그리고 system calls & functions 에 해당하는 부분 중 opendir(), readdir(), closedir()를 사용해보면서 그 쓰임새를 익힌다. gcc 를 이용한 compile 과 binary 파일을 실행해보는 것을 실습해본다.

## 2. Flow chart



코드의 흐름은 위와 같다. directory 에 접근하기 위한 DIR\* 변수와 directory entry 정보를 저장할 dirent 구조체 변수를 선언하여 경로를 받는다면 해당 경로로 받지 않는다면 현재 디렉토리를 기본값으로 하여 디렉토리에 접근한다. 경로는 실행파일과 함께 입력되는 인자값으로 받는데 인자의 개수에 따른 분기가 일어난다. 오직 하나의 경로만을 받기 때문이다.(그렇지 않을시 에러코드 출력) dirent 구조체내 d\_name 이라는 변수에서 파일들의 이름을 가져와 list 배열에 저장한다. (이는 정렬을 하기 위함이다.) for 문을 중첩하여 사용하는 selection sort 방식을 사용하고 strcmp 함수를 통해 대소문자를 구분하지 않고 문자열을 오름차순으로 정렬한다.

### 3. Pseudo Code

1. Include the necessary header files: stdio.h, dirent.h, and string.h.
2. Declare a DIR pointer dirp, a struct dirent pointer dir, a 2D character array list, and an integer count.
3. Check the number of arguments passed to the program:
  - a. If there are more than 2 arguments, print an error message and return -1.
  - b. If there is only 1 argument, set dirp to the current directory (.).
  - c. Otherwise, set dirp to the directory passed as an argument.
4. If dirp is NULL, print an error message and return -1.
5. Use a while loop to read the directory entries with readdir.
6. For each entry, copy the name to the list array and increment count.

7. If the first character of the name is a dot (.), skip the entry.
8. Use a nested for loop to sort the list array in ascending order using `strcasecmp` function.
9. Finally, use a for loop to print the sorted list array to the console.
10. Return 0 to indicate successful execution.

#### 4. 결과화면

```
kw2019202050@ubuntu:~/work$ ls
2019202050_simple_ls.c  B.txt  good.txt  quit.txt  word.txt
a.txt                  C.txt  Makefile  Test.txt
kw2019202050@ubuntu:~/work$ make
gcc -o simple_ls 2019202050_simple_ls.c
kw2019202050@ubuntu:~/work$ ./simple_ls
2019202050_simple_ls.c
a.txt
B.txt
C.txt
good.txt
Makefile
quit.txt
simple_ls
Test.txt
word.txt
kw2019202050@ubuntu:~/work$ ls
2019202050_simple_ls.c  B.txt  good.txt  quit.txt  Test.txt
a.txt                  C.txt  Makefile  simple_ls  word.txt
kw2019202050@ubuntu:~/work$
```

우선 위의 실행 코드를 보자. 2019202050\_simple\_ls.c 파일을 생성하고 `make` 를 통해 컴파일 하여 실행파일을 생성 후 실행한다. 그 결과 대소문자를 구분하지 않고 오름차순으로 숨김파일을 제외한 파일들의 리스트가 출력되는 것을 볼 수 있다. `ls` 는 정렬되지 않았지만 실행파일을 통한 리스트 출력을 정렬되어있음을 알 수 있다.

```

kw2019202050@ubuntu:~$ work/simple_ls
Desktop
Documents
Downloads
examples.desktop
Music
Pictures
Public
splab_commands
Templates
Videos
work
kw2019202050@ubuntu:~$ work/simple_ls work
2019202050_simple_ls.c
a.txt
B.txt
C.txt
good.txt
quit.txt
simple_ls
Test.txt
word.txt
kw2019202050@ubuntu:~$ work/simple_ls work Desktop
simple_ls: only one directory path can be processed
kw2019202050@ubuntu:~$ work/simple_ls good
simple_ls: cannot access 'good' : No such directory
kw2019202050@ubuntu:~$ touch a.txt
kw2019202050@ubuntu:~$ work/simple_ls a.txt
simple_ls: cannot access 'a.txt' : No such directory
kw2019202050@ubuntu:~$

```

이번에는 실행파일을 실행하면서 인자를 주는 경우이다. 우선 현재 작업 디렉토리가 아닌 곳에서 파일을 실행할 수 있음을 보여주고 그 다음 work 라는 폴더의 리스트를 출력하게끔 work 라는 인자를 넣었다. 따라서 work 라는 경로의 디렉토리 파일들이 출력되었다. 인자를 두개 넣은 경우와 존재하지 않는 디렉토리 그리고 디렉토리가 아닌 파일명을 인자로 받은 경우 에러코드를 출력하는 것을 확인할 수 있다.

## 5. 고찰

과제를 진행하면서 디렉토리 경로에 따른 파일들에 접근하는 것이 익숙치 않아 어려웠다. 또한 인자를 활용하는 방법을 처음 사용해보면서 문법오류가 많이 났던 것 같다. opendir()이나 readdir()함수의 인자와 반환형에 대한 format 을 숙지하여야 적절하게 사용이 가능했다. 그래야 올바른 값을 반환했을 시 적절한 변수에 값을 할당할 수 있고 잘못된 값을 반환할 시 그 값이 NULL 값이라는 것을 알아야 조건문을 활용할 수 있었기 때문이다. 또한 숨김파일을 출력하지 않는 법이 무엇일까에 대한

고민을 하게 됐다. 해결은 list 변수에 readdir()함수를 통해 불러올 때 첫번째 문자가 .일때 count 를 증가시키지 않아 다음 파일을 불러오게 된다면 덮어쓰게 하였고 다음파일이 없아해도 count 변수를 기반으로 반복문을 실행하게 되므로 문제를 해결할 수 있게 되었다. 실습을 통해 directory contents 에 대해 알아볼 수 있는 경험을 하게 되었다.

## 6. Reference

시스템프로그래밍실습 강의자료 참조