

# 시스템프로그래밍실습 보고서

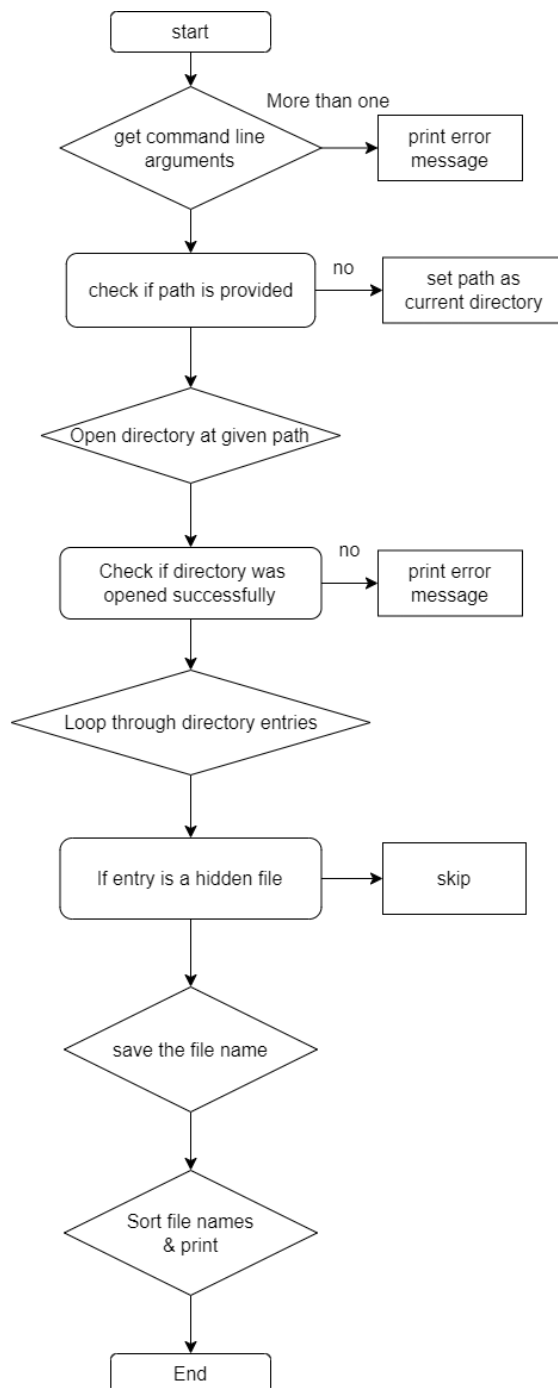
## Assignment 1-1

과목	시스템프로그래밍실습
담당교수	이기훈교수님
학과	컴퓨터정보공학부
학번	2021202058
이름	송채영

## 1. Introduction

ls 명령어를 통해 파일 이름만 출력하는 Simple ls 를 구현해본다. 하나의 파일 경로 안에 있는 파일의 이름을 출력해야 하며, 대소문자 구분 없이, 알파벳 순으로 정렬되도록 결과를 출력한다. 이 과제를 통해 gcc 를 활용해 compile 을 해본다. 더 나아가 reading directories 인, opendir, readdir, closedir 을 활용해 코드를 구현하며 data type 에 대해 알아본다.

## 2. Flow chart



코드의 전체적인 흐름을 flow chart 로 나타내었다. 자세하게 설명해보면, 우선 파일 인자를 받은 후 2 개 이상의 인자가 제공된 경우, 에러 메시지를 출력해주었다. 아닌 경우 경로가 제공된 경우인지 확인하였다. 파일 경로를 인자로 주지 않았을 경우 현재 directory 를 경로로 설정한다. 그 후 지정된 경로에서 directory 를 열어준다. Directory 가 성공적으로 열리지 않았다면 에러 메시지를 출력해주고, 열렸다면 위의 flow chart 에서는 생략하였지만, 파일 이름을 저장하기 위한 배열에 메모리를 할당한 후 directory 항목을 통해 loop 문을 돈다. 만약 숨겨진 파일, 즉 '.'으로 시작하는 파일인 경우 넘어간다. 파일 이름을 배열에 저장하고 directory 를 닫은 후 정렬해주었다. 파일 이름을 출력해주고 이 부분 역시 flow chart 에서 생략하였지만, 배열에 할당된 메모리를 비우고 프로그램을 종료한다.

### 3. Pseudo Code

1. Initialize a variable 'files1' to 0 and allocate memory for an array of pointers 'files' of size 1000.
2. Loop through each file in the directory using 'readdir' function:
  - a. If the filename starts with '.', skip it.
  - b. Otherwise, allocate memory for a string to hold the filename and store the filename in the array 'files'.
  - c. Increment the variable 'files1'.
3. Close the directory using 'closedir' function.
4. Sort the file names stored in the array 'files' using the 'strcmp' function:
  - a. Loop through each pair of files in the array:
    - i. Compare the file names using the 'strcmp' function.
    - ii. If the first file name is greater than the second file name, swap the files.
    - iii. Continue looping until all pairs of files have been compared and sorted.
5. Loop through the sorted file names in the array 'files':
  - a. Print each file name to the console.
  - b. Deallocate the memory used for the file name.
6. Deallocate the memory used for the array of pointers 'files'.
7. Exit the program.

#### 4. 결과화면

```
file1.txt file2.txt file3.txt fileA.txt fileB.txt fileC.txt file1.txt file2.txt file3.txt fileA.txt fileB.txt fileC.txt
kw2021202058@ubuntu:~/work$ make
gcc 2021202058_simple_ls.c -o simple_ls
kw2021202058@ubuntu:~/work$ ls
2021202058_simple_ls.c file2.txt fileB.txt LINUX simple_ls text.txt
empty.txt file3.txt fileC.txt ls.c SP_lab
file1.txt fileA.txt hello.txt Makefile spIs
```

c 파일을 생성한 후 'make' 명령어로 컴파일 한 후 실행파일인 simple\_ls 를 생성하였다.

```
simple_ls: only one directory path can be processed
kw2021202058@ubuntu:~/work$ ./simple_ls
2021202058_simple_ls.c
empty.txt
file1.txt
file2.txt
file3.txt
fileA.txt
fileB.txt
fileC.txt
hello.txt
LINUX
ls.c
Makefile
simple_ls
SP_lab
spIs
text.txt
kw2021202058@ubuntu:~/work$ ./simple_ls ..
2021202058
Assignment1
cli
cli.c
Desktop
Documents
Downloads
examples.desktop
hello
Music
Pictures
Public
splab_commands
srv
srv.c
Templates
Videos
work
```

숨김파일을 출력하지 않으며, 대소문자 구분 없이 파일 이름이 알파벳 순으로 정렬된 결과를 볼 수 있다.

```
kw2021202058@ubuntu:~/work$ ./simple_ls ..
2021202058
Assignment1
cli
cli.c
Desktop
Documents
Downloads
examples.desktop
hello
Music
Pictures
Public
splab_commands
srv
srv.c
Templates
Videos
work
kw2021202058@ubuntu:~/work$ cd ..
kw2021202058@ubuntu:~$ ls
2021202058 cli.c Downloads Music splab_commands Templates
Assignment1 Desktop examples.desktop Pictures srv Videos
cli Documents hello Public srv.c work
kw2021202058@ubuntu:~$
```

위의 사진은 '.', 즉 work directory 의 이전 directory 인 home directory 의 파일 이름을 출력하였다. 역시 숨김파일을 출력하지 않으며, 대소문자 구분 없이 파일 이름이 알파벳

순으로 정렬된 것을 볼 수 있다. ls 명령어를 통해 home directory 의 파일들을 볼 수 있으며, 이를 통해 home directory 의 파일들이 잘 정렬됐음을 확인하였다.

```
kw2021202058@ubuntu:~/work$ ls
2021202058_simple_ls.c  file2.txt  fileB.txt  LINUX      simple_ls  text.txt
empty.txt              file3.txt  fileC.txt  ls.c       SP_lab
file1.txt              fileA.txt  hello.txt  Makefile   spls
kw2021202058@ubuntu:~/work$ ./simple_ls hello.c
simple_ls : cannot access 'hello.c' : No such directory
kw2021202058@ubuntu:~/work$ ./simple_ls Not_file
simple_ls : cannot access 'Not_file' : No such directory
kw2021202058@ubuntu:~/work$ ./simple_ls html Linux
simple_ls : only one directory path can be processed
kw2021202058@ubuntu:~/work$
```

디렉토리가 아닌 파일을 입력한 경우, 존재하지 않는 디렉토리를 입력한 경우, 두 개 이상의 파일 경로를 입력한 경우 각각에 해당하는 에러 메시지를 적절하게 출력한 것을 확인할 수 있다.

## 5. 고찰

지난 과제에서 마우스 커서로 위치를 지정하지 못하는 점과 vi editor 의 사용법이 낯설고 헷갈려 과제를 진행하는데 어려움이 있었다고 고찰에 적었었다. 하지만 계속해서 사용해보니 basic 실습, 과제를 수행했을 때 보다 vi editor 와 Linux 에 익숙해진 것 같다. 또한 이제는 자유자재로 명령어를 활용해서 파일을 삭제하고 이동시키고 등등의 것을 window 에서 하는 것처럼 할 수 있게 되었다. 하지만 vi editor 에서 코드를 작성하면서 나는 에러, 예를 들면 ()를 맞춰주지 않았던지, ,를 .으로 쓴다던지 등등 작은 실수가 있었다. gcc 를 활용하여 컴파일을 했을 때 하나하나 확인하고 고쳐야 한다는 어려움이 있었지만, set number 명령을 통해 line 을 확인한 후 해당 line 의 오류를 해결하면 됐기에 큰 어려움은 없었던 것 같다. 또한 주어진 조건을 만족하는 코드를 구현하기 위해 함수도 찾아보고 하며 C 에 대해서도 공부할 수 있었다. 이번 과제에서 배웠던 opendir, readdir, closedir 를 활용해 코드를 작성하였지만, 더 익숙하게 사용하기 위해서는 공부가 필요하다고 느꼈다.

## 6. Reference

시스템프로그래밍실습 강의자료