

# 시스템프로그래밍실습 보고서

## Assignment 1-1

학 과: 컴퓨터정보공학부

담당교수: 김태석 교수님

실습분반: 실습3 금78(E)

학 번: 2018202018

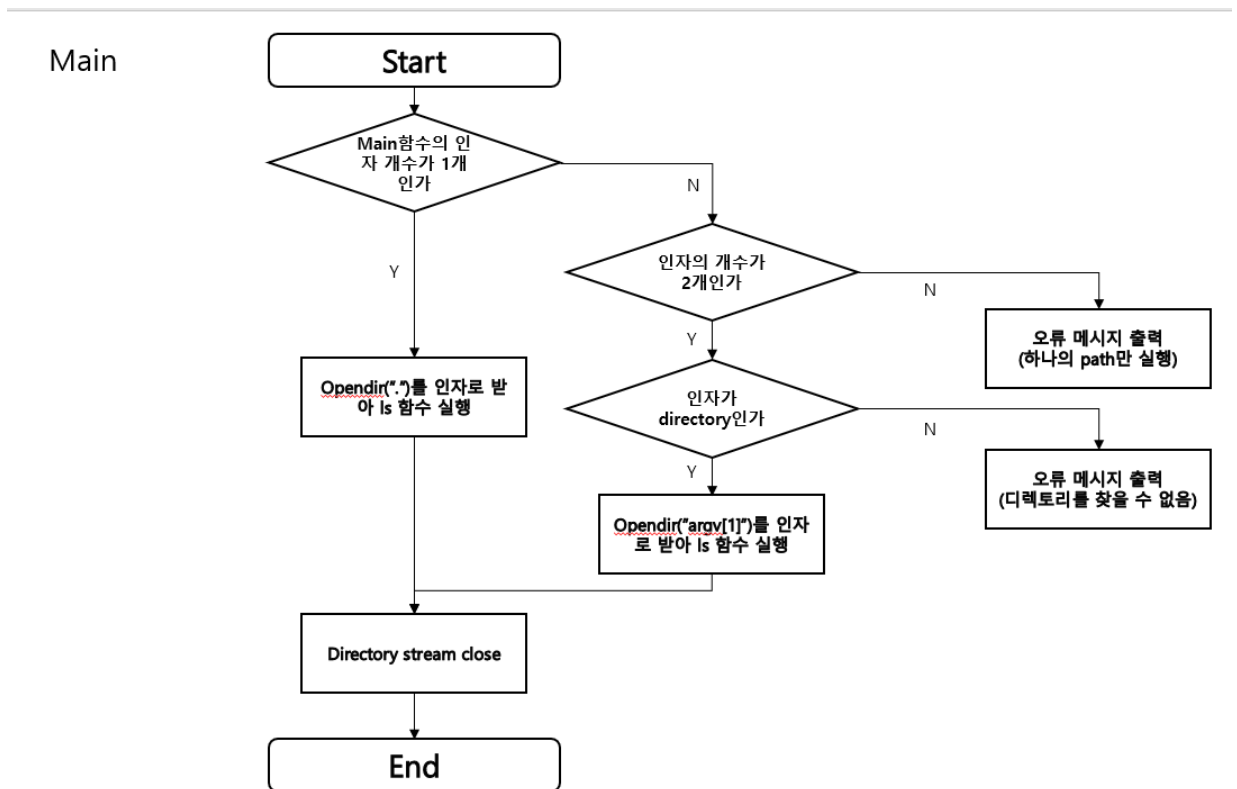
성 명: 유승재

## 1. Introduction

이번 과제는 vi editor를 통해 ubuntu 환경 내에서 c언어 코드를 만들어 gcc 컴파일러를 이용하여 컴파일을 수행하여 실행파일을 만들어내는 과정과 c언어를 이용하여 Linux의 기본 명령어 중 하나인 ls명령어의 구현 및 기능 변형을 하는 것이 이번 과제의 목표입니다. 기존의 ls 명령어와 다르게 파일의 이름들을 오름차순으로 정리하면서 숨김파일은 표시하지 않아야 하며 예외 처리를 해야 합니다.

## 2. Flow chart

### A. Main

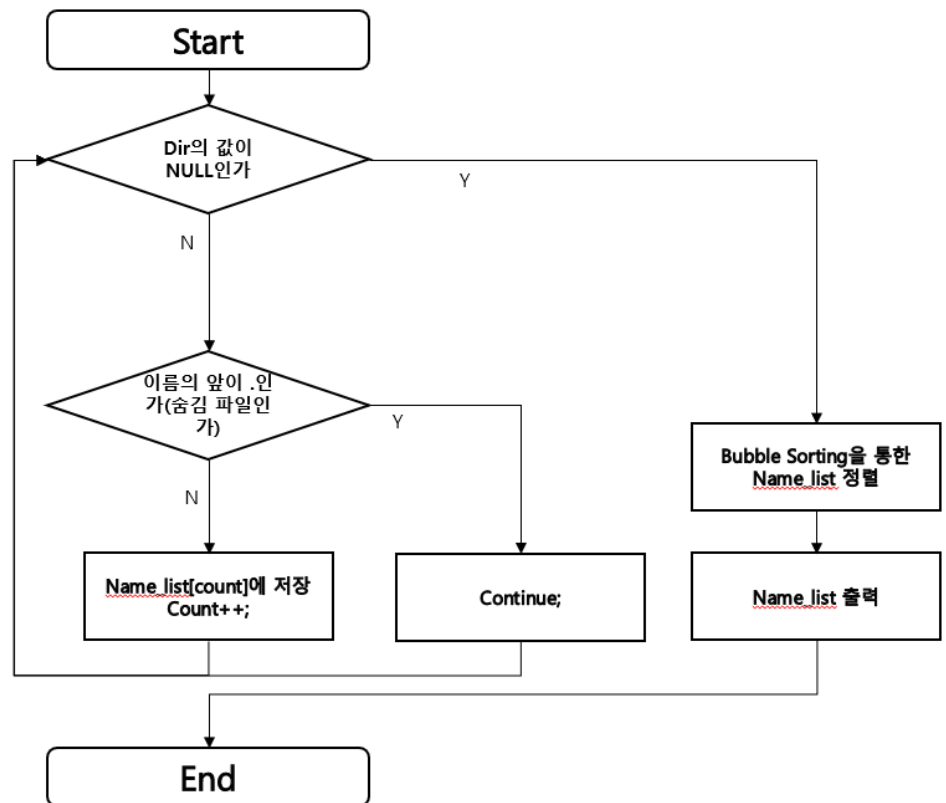


Main 함수에서는 Main 함수 자체에서 argc와 argv를 이용해 인자를 받고 인자의 개수와 종류에 따라서 flow를 다르게 설정했습니다.

### B. Ls

Ls 함수에서는 main 함수에서 DIR\* 자료형의 값을 넘겨주고 그것을 인자로 하는 함수입니다. DIR\* 값을 받아 이를 통해 `readdir`를 사용하여 `d_name`들을 저장하고 정렬 후 출력하는 형식의 함수입니다.

ls



### 3. Pseudo Code

```
void ls(DIR *a) {  
  
    int count = 0;  
  
    char *temp; // use to sort  
  
    char *name_list[1024];  
  
    struct dirent *dir;  
  
    While (디렉토리에 있는 파일의 수만큼){  
  
        if (맨 앞의 문자가 "."일 경우(숨김 파일일 경우)){  
  
            다시 While로 돌아감;  
  
        }  
    }  
}
```

```

        Name_list에 파일 이름을 저장;

        Count 하나 증가;

    }

    저장된 Name_list를 버블 정렬을 통해 알파벳순으로 정렬;

}

Name_list 전부 순서대로 출력;

}

}

int main (int argc, char *argv[]){

    DIR *dirp;

    if (실행 시에 인자가 없을 경우){ // if no argument or argument is "."

        현재의 디렉토리의 포인터로 입력받음; // assign current directory

        현재의 디렉토리 포인터를 인자로 ls함수 출력;

    }

    else {

        if (인자가 1개가 아닐경우){ // if arguments are more than 2

            에러 메시지 (경로는 1개만 입력해주세요) 출력;

            프로그램 종료;

        }

        if (인자의 readdir return 값이 NULL일 경우 (인자가 디렉토리가 아니거나 없는
파일일 경우)

```

에러 메시지 ( 잘못된 경로입니다 ) 출력;

프로그램 종료;

}

Main 함수의 인자의 디렉토리 포인터로 입력받음;

Ls 함수 출력;

}

디렉토리 포인터 해제;

return 0;

}

#### 4. 결과 화면

##### A. 인자가 1개일 경우

```
kw2018202018@ubuntu:~/work$ ./simple_ls /home/kw2018202018
assignment1
assignment2
Desktop
Documents
Downloads
examples.desktop
Makefile
Maketest
Music
Pictures
Public
simple_ls
simple_ls.c
splab_commands
Templates
test
Videos
work
kw2018202018@ubuntu:~/work$ ls /home/kw2018202018
assignment1 Documents Makefile Pictures simple_ls.c test
assignment2 Downloads Maketest Public splab_commands Videos
Desktop examples.desktop Music simple_ls Templates work
```

Ls 함수와 동일한 목록을 출력합니다. Simple\_ls는 알파벳 순으로 정렬하여 출력합니다.

B. 인자가 없을 경우

```
kw2018202018@ubuntu: ~/work
kw2018202018@ubuntu:~/work$ ./simple_ls
2018202018_simple_ls.c
file1.txt
file2.txt
file3.txt
fileC.txt
hello.txt
Makefile
simple_ls
SP_lab
kw2018202018@ubuntu:~/work$ ls
2018202018_simple_ls.c  file2.txt  fileC.txt  Makefile  SP_lab
file1.txt               file3.txt  hello.txt  simple_ls
kw2018202018@ubuntu:~/work$
```

현재의 디렉토리를 그대로 출력합니다.

C. 인자가 2개일 경우

```
kw2018202018@ubuntu: ~/work
kw2018202018@ubuntu:~/work$ ./simple_ls /home ~
simple_ls: only one directory path can be processed
kw2018202018@ubuntu:~/work$
```

두 개의 인자일 시에는 에러 메시지를 출력합니다.

D. 인자가 directory가 아니거나 존재하지 않을 경우

```
kw2018202018@ubuntu:~/work$ ls
2018202018_simple_ls.c  file2.txt  fileC.txt  Makefile  SP_lab
file1.txt               file3.txt  hello.txt  simple_ls
kw2018202018@ubuntu:~/work$ ./simple_ls hello.txt
simple ls: cannot access 'hello.txt' : No such directory
kw2018202018@ubuntu:~/work$ ./simple_ls NON
simple ls: cannot access 'NON' : No such directory
kw2018202018@ubuntu:~/work$
```

에러 메시지를 출력합니다.

5. 고찰

이번 과제는 Linux 환경에서 C언어로 된 코드를 작성하고 컴파일하며 코드를 구현하는 제대로 된 첫 과정이었습니다. Vi editor를 사용하여 코드를 작성하는 과정에서도 이전에 사용하던 VS나 VS code같은 다른 에디터와는 다르게 편의성으로 제공해주는 것이 크게 없었기 때문에 불편한 것을 느꼈습니다. 또한 컴파일하는 과정 또한 제가 Makefile을 직접 만들고 그를 통하여 컴파일 하는 것이 번거로운 과정이었고 그 과정내에서 익숙하지

않은 환경에 적응하는 것이 중요했다고 생각합니다. 또한 기존의 c 코딩과는 다르게 새로운 함수들을 사용함으로써 이전과는 다르게 코드를 작성할 때 많은 시간이 소요되었습니다. 이런 어려움을 겪게 되니 새로운 코딩 환경에 빠르게 적응하는 것이 중요하다고 생각하게 되었고 또한 새로운 함수, 명령어들을 몸에 익히고 많이 사용해보는 것이 중요하다고 생각하게 되었습니다. 그리고 보고서를 작성하며 Flow chart와 Pseudo Code를 작성해보게 되었고 이를 통해 제 프로그램의 작동 알고리즘을 조금 더 명확하게 정하고 프로그램 구현을 시작할 수 있게 된 것 같습니다.

## 6. Reference

강의자료

<https://gmlwjd9405.github.io/2018/05/06/algorithm-bubble-sort.html> - 버블정렬 참고