



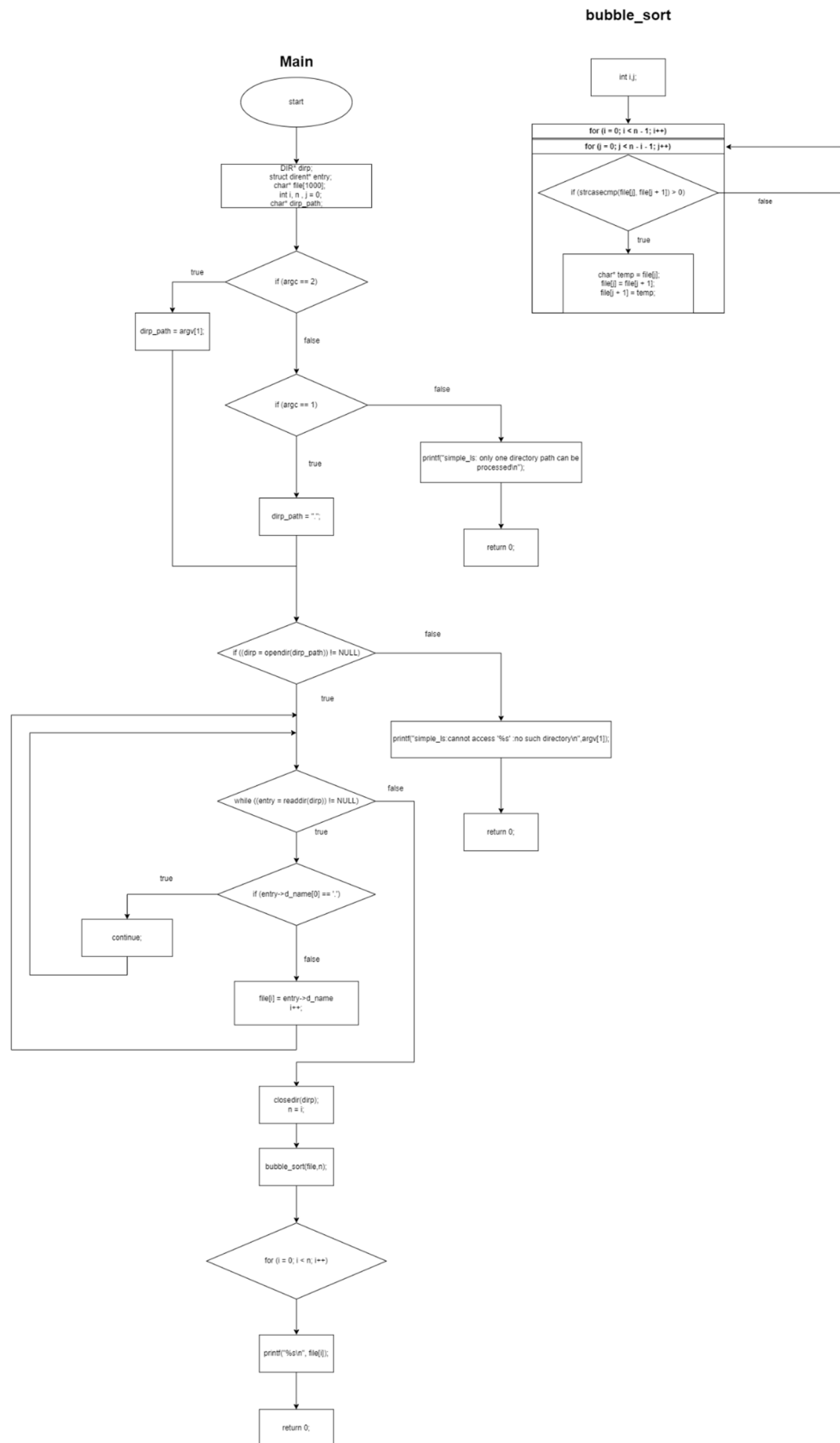
시스템프로그래밍실습  
assignment1-1 과제

수업 명 : 시스템프로그래밍실습  
과제 이름 : assignment1-1  
담당 교수님 : 김태석 교수님  
학 번 : 2019202005  
이 름 : 남종식

## 과제 소개

이번 과제는 리눅스 명령어 ls를 직접 구현하는 과제입니다. 이때 실행파일은 simple\_ls로 만들어 이 파일을 실행하면 대소문자 구문없이 알파벳순으로 정렬되어 파일 이름들이 출력되어야 합니다. 이때 정렬 함수 라이브러리는 사용하지 않아야 하며 정렬 함수 중 하나를 직접 구현하여 사용해야 합니다. 파일 경로가 인자로 따로 주어진다면 이 디렉토리의 하위 목록을 출력하고 주어지지 않는다면 현재 디렉토리의 하위 목록을 출력합니다. 만약 인자로 파일이 입력되거나 두개 이상이 입력되거나 디렉토리가 존재하지 않는다면 예외처리를 해야 합니다.

# Flow Chart



## Pseudo Code

```
bubble_sort function(parameter to store list of files, parameter to store the size of the array)
{
  Declare variable i,j to run for statement
  Repeatedly increasing by 1 from i=0 to n-1
  {
    Repeatedly increasing by 1 from j=0 to n-i-1
    {
      When comparing two consecutive files, if the preceding file is larger
      change the location of the two files
    }
  }
}

main function(Parameter to store the number of files, Parameter to save directory files)
{
  pointer to directory stream
  Pointer to a structure representing a directory entry
  char* array to store sorted files
  Declare variable i,j,n to run for statement

  set pointer to the path of the file
  when argc==2
  {
    The file path is the input path
  }
  when argc==1
  {
    file path is the current directory
  }
  when argc>3
  {
    print the exception message
    exit program
  }

  If the directory is open
  {
    Repeat until all files in the directory have been read.
    {

      If there is a hidden file
      {
        Hidden files are not output
      }
      Store the names of files in an array
      access with the following array
```

```

    }
    close directory
    Save number of files

    sort the list of files

    Repeatedly increasing by 1 from i=0 to n
    {
        print sorted files
    }
}
If the directory does not open
{
    print the exception message
    exit program
}
exit program
}

```

## 결과화면

```

kw2019202005@ubuntu:~$ cd work
kw2019202005@ubuntu:~/work$ vi 2019202005_simple_ls.c
kw2019202005@ubuntu:~/work$ vi Makefile
kw2019202005@ubuntu:~/work$ make
gcc -o simple_ls 2019202005_simple_ls.c

```

과제는 work 디렉토리에서 진행하였고 소스파일과 Makefile을 만들어 make를 진행했습니다.

```

kw2019202005@ubuntu:~/work$ ./simple_ls
2019202005_simple_ls.c
ass
exam
file2.txt
file3.txt
hello.c
html
html_ls.html
LINUX
Makefile
simple_ls
SP_lab
test
testing
text.txt

```

강의자료에 있었던 파일 및 디렉토리를 추가하였고 이들이 똑같이 출력되는 지 확인하였습니다. 대소문자 상관하지 않고 알파벳 순으로 정렬되어 출력되는 모습을 확인할 수 있습니다.

```
kw2019202005@ubuntu:~/work$ ./simple_ls jongsik.c
simple_ls: cannot access 'jongsik.c' :no such directory
```

존재하지 않는 파일을 입력했을 때 출력되는 화면입니다.

```
kw2019202005@ubuntu:~/work$ ./simple_ls hello.c
simple_ls: cannot access 'hello.c' :no such directory
```

인자로 디렉토리가 아닌 파일을 입력한 경우 출력되는 화면입니다.

```
kw2019202005@ubuntu:~/work$ ./simple_ls html LINUX
simple_ls: only one directory path can be processed
```

인자로 두 개 이상의 파일 경로를 입력하는 경우 출력되는 화면입니다.

## 고찰

먼저 main함수는 매개변수로 입력 인자의 개수와 입력 인자들을 배열형으로 받습니다. 디렉토리 스트림을 나타내는 포인터와 dirent structure를 가리키는 포인터를 만들고 opendir() 함수를 사용하여 디렉토리 내의 파일들을 읽을 수 있고 readdir() 함수를 사용하여 파일 이름들을 확인할 수 있습니다.

실행파일을 실행할 때 파일경로가 주어진다면 그 디렉토리안의 파일이름을 출력시키고 주어지지 않는다면 현재파일을 실행하는 부분과 히든파일은 출력되지 않는 부분은 조건문으로 처리하였습니다. 처음에 구했했을 때 출력결과에 .과 ..이 있는 것을 확인했습니다. 이때 히든파일을 출력하면 안 되는 조건이 생각나서 이를 출력하지 않게 조건문으로 처리했습니다. 히든파일을 제외한 파일 및 디렉토리들을 담기 위한 배열을 하나 만들어 저장했습니다. 일단 버블 정렬을 사용하기 위해 이를 함수로 정의했습니다. 이는 인접한 두 요소를 비교하여 정렬합니다. 매 반복마다 가장 큰 요소가 배열의 마지막으로 이동하기 때문에, 정렬된 부분을 제외한 나머지 요소만 비교하면 됩니다. i는 정렬된 요소의 개수이므로, n-i-1까지만 비교하면 됩니다. 버블 정렬을 이용해 이 배열에 저장된 파일 및 디렉토리들을 정렬하였고 대소문자에 관계없이 정렬하기 위해 strcasecmp함수를 사용했습니다. 이는 <string.h>헤더파일을 필요로 하며 함수 형식은 char \* strcmp(const char \*s1, const char \*s2) 입니다. 저는 첫번째 인자에 배열의 이전의 요소 두번째 인자의 배열의 다음의 요소를 넣어줬습니다. 만약 s1이 더 크다면 0보다 큰 값을 반환하고 같으면 0을 반환합니다. 그리고 s2가 더 크다면 0보다 작은 값을 반환합니다. 배열에서 이전의 요소와 다음의 요소를 비교해 이전의 요소 값이 더 크다면 다음의 요소 값과 서로 자리를

바꿔주는 버블 정렬을 사용했기 때문에 strcasecmp함수에서 1을 반환하는  
상황에만 버블 정렬을 진행해줍니다. 처음에 구현했을 때는 strcmp함수를  
이용해서 대소문자가 구별되었습니다. 그래서 찾아보니 대소문자를 구별하지  
않는 strcasecmp함수가 있어서 이를 사용해 다시 구현했습니다. 이렇게 정렬된  
파일들을 하나씩 출력하여 ls명령어의 기능을 구현했습니다.

## Reference

2023년 1학기 시스템프로그래밍 & 시스템 프로그래밍 실습 강의자료  
Assignment 1-1