

Assignment#2-1

시스템 프로그래밍 실습

제출일: 4월 05일 화요일

분 반: 화요일

담당 교수: 신영주

학 번: 2015722025

학 과: 컴퓨터정보공학부

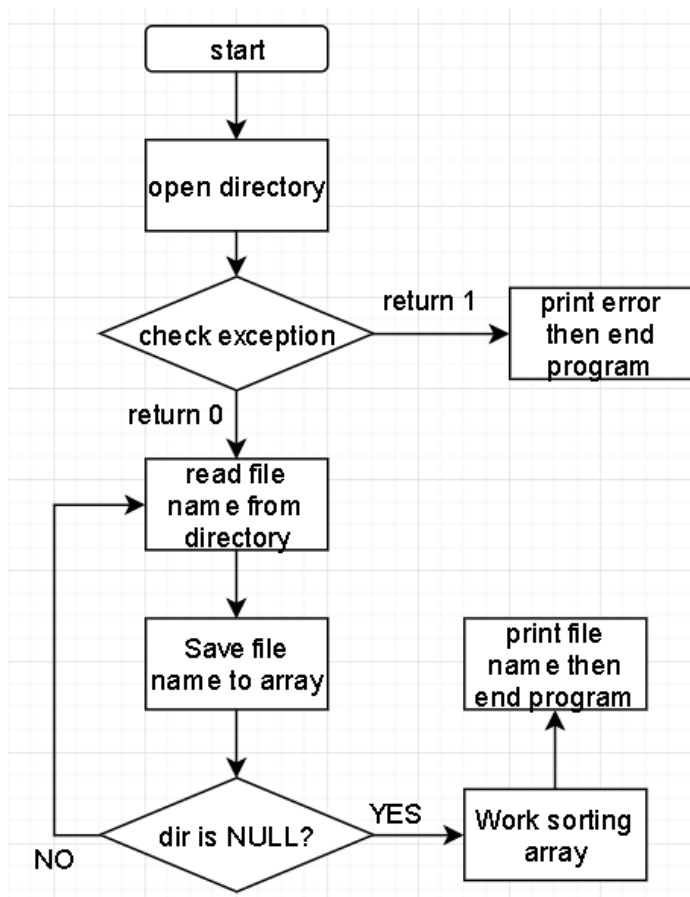
이 름: 정용훈

1. Introduction

이번 과제는 파일 이름만 출력하는 ls를 구현하는 과제다. 기존의 ls와 기능은 거의 비슷하며, 파일들의 이름이 사전 순으로 sorting되어 출력된다는 점과 예외처리를 하지 않으면 히든 파일들도 함께 출력된다는 점에 주의해야 한다. 또한 인자를 2개 이상 받았을 때와 디렉토리가 아닌 파일을 인자로 받게 되면 에러를 출력해준다.

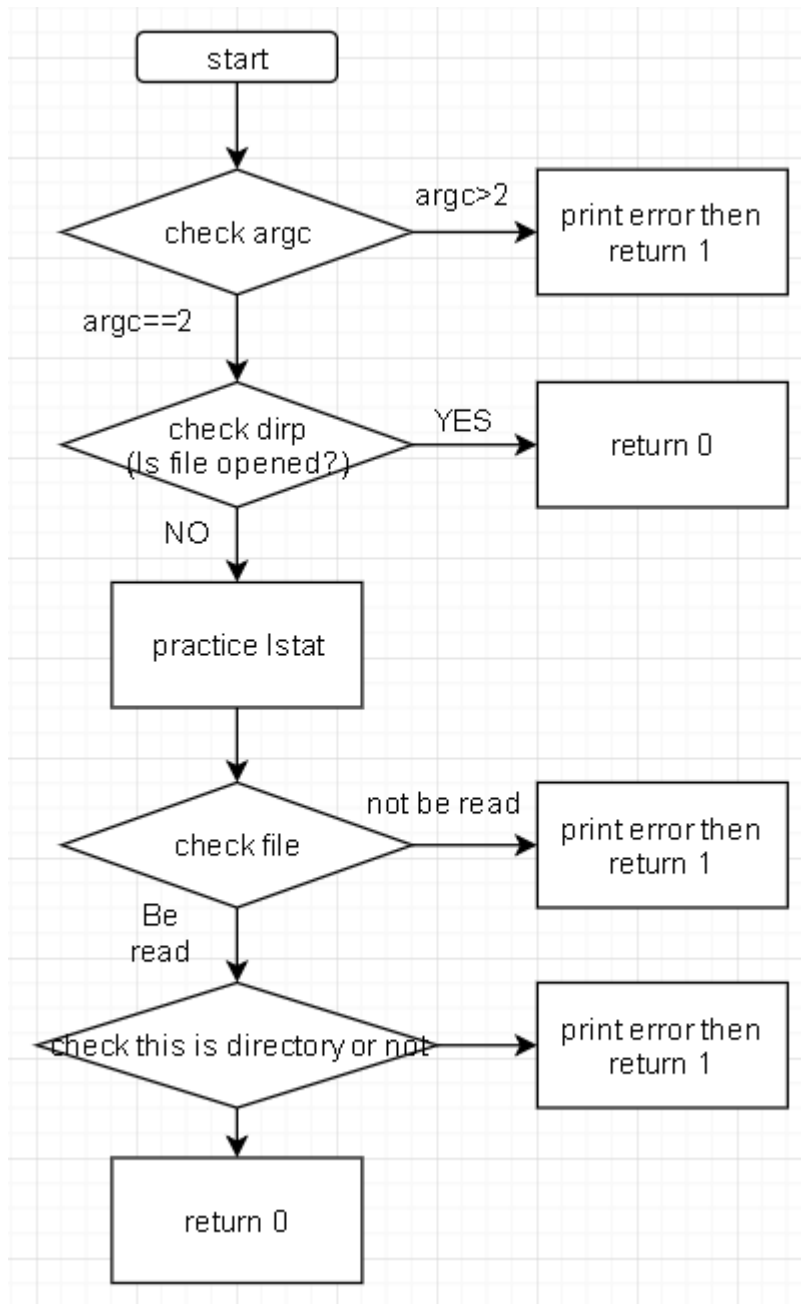
2. Flow Chart

Main



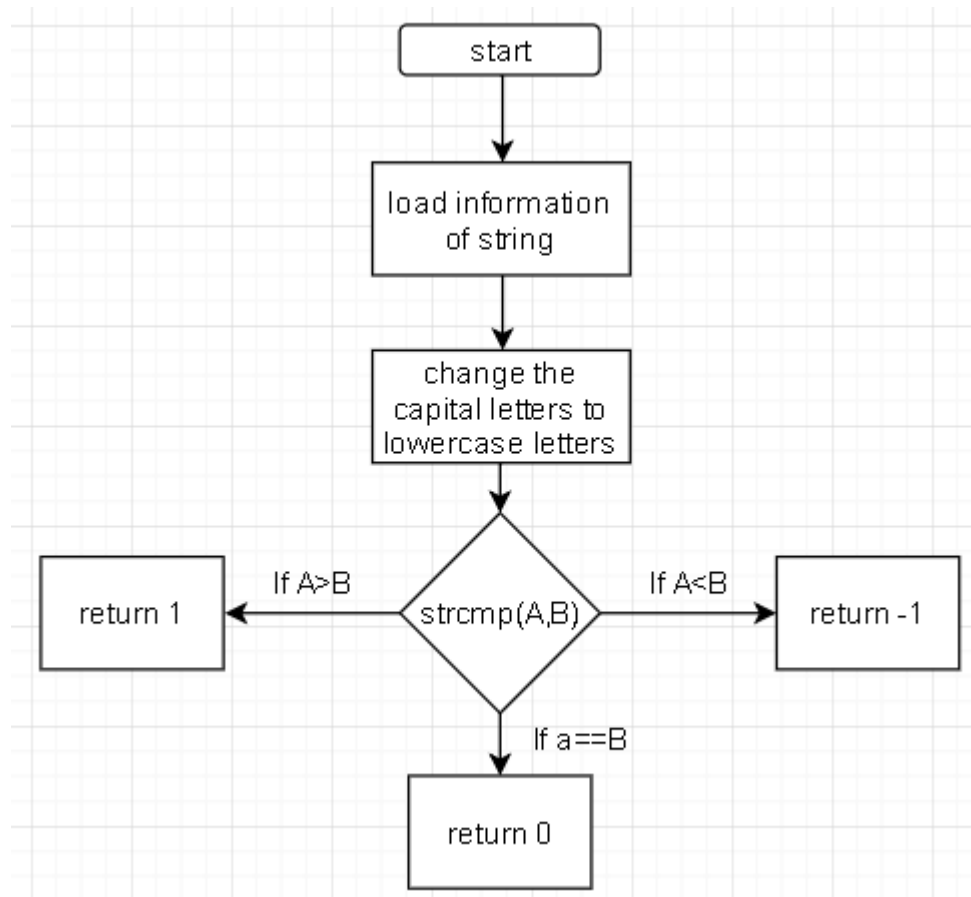
다음은 전체적인 동작을 컨트롤하는 main함수의 flow chart다. 코드를 실행하면 argv에 있는 파일의 이름을 기반으로 open함수를 실행한 후 예외처리 함수로 넘어가 예외처리를 진행하게 된다. 예외처리가 완료되었다면 파일의 이름들을 따로 선언한 array에 옮긴 후 마지막으로 파일의 이름순서로 정렬 후 조건에 맞게 출력하고 프로그램을 종료하면된다.

Exception



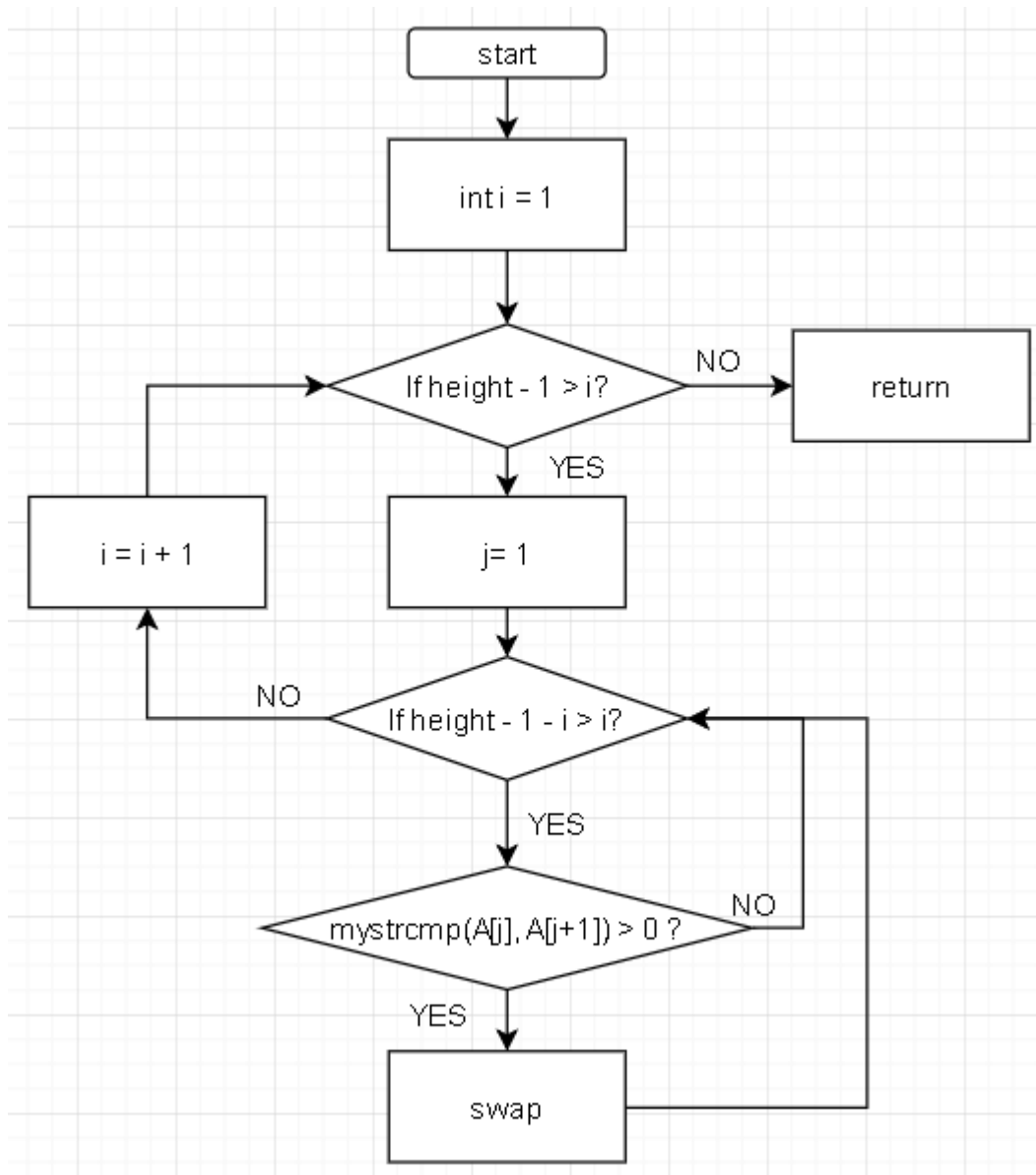
다음은 예외처리를 위한 함수의 flow chart다. 함수에 들어온 후 argc의 정수 값을 check한 후 해당 인자가 2개보다 많을 경우 인자 관련된 error메시지를 출력하고 적당한 값을 return 하면서 함수를 종료한다. 그 후 open함수를 사용하여 dirp가 정상적으로 열렸는지 check하며 해당 파일이 열렸으면 정상적으로 실행된 return 값을 준 후 함수를 종료한다. 그렇지 않은 경우는 해당 파일이 존재하지 않거나 directory가 아닌 경우를 판단하기 위해 stat함수를 사용하여 파일을 한번 더 읽는다. 만약 stat을 사용하여 읽지 않았다면 존재하지 않는 파일이고, 읽었지만 directory를 판단해주는 함수에서 directory가 아니라는 값을 반환 받게 되면 error를 print한다.

Mystrcmp



해당 flow chart는 대소문자 구분을 하지 않기 위한 strcmp함수를 구현한 것이다. 해당 함수의 원리는 간단하다. 모든 파일의 이름을 소문자로 바꾼 후 library함수인 strcmp를 사용하여 알맞은 값들을 return 해주면 간단하게 해결할 수 있다.

Sorting



가장 이해하기 쉬운 버블소트의 flow chart다. 모든 string이 정렬될 때까지 계속해서 반복문을 통하여 정렬한다. 정렬이 모두 끝나면 함수가 종료된다.

3. Pseudo code

Main

```
int main(int arg, char** argv)
{
    DIR *dirp;
    struct dirent *dir;
    int height = 2;

    if(argument count > 1)
        dirp=opendir(argv[1]);
    else
        dirp=opendir(".");

    if(exist exception)
        end program

    while(exist information of file name in opened directory)
    {
        dir=readdir(dirp);
        copy dir->d_name to other array
    }

    sorting array;

    print array;
}
```

Exception

```
int exception(int argc, char** argv, DIR* dirp)
{
    struct dirent *dir;
    struct stat buf;

    if(argc>2)
    {
        print error string;
        return 1;
    }

    if(argc==2)
    {
        if(directory is not opened)
        {
            if(load file attribute<0)
            {
                print error;
                return 1;
            }
            if(!check is directory or not)
            {
                print error;
                return 1;
            }
        }
    }
    return 0;
}
```

mystrcmp

```
int mystrcmp(char* C, char* D)
{
    char* A=(char*)malloc(sizeof(int)*300);
    char* B=(char*)malloc(sizeof(int)*300);
    strcpy(A, C);
    strcpy(B, D);

    while(all string of A is not chnaged)
        A[each element]+ 32;

    while(all string of B is not chnaged)
        B[each element]+ 32;

    if (A is bigger than B)
        return 1;
    else if (B is bigger than A)
        return -1;
    else
        return 0;
}
```

Sorting

```
void sorting(char**A, int height)
{
    char* tmp=(char*)malloc(sizeof(int)*300);

    for (int i = 1; i < height - 1; i++)
    {
        for (int j = 1; j < height - 1 - i; j++)
        {
            if (A[j] is bigger than A[j+1])
                swap A[j] and A[j+1];
        }
    }
}
```


4. Result

```
sp20157522025@ubuntu:~$ ls
Desktop    Downloads    J            Music        Public        Templates    test2
Documents  examples.desktop  Makefile    Pictures     simple_ls.c   test        Videos
```

먼저 ls명령을 통하여 존재하는 파일의 list를 확인한다.

```
sp20157522025@ubuntu:~$ make
gcc -g simple_ls.c -o simple_ls
sp20157522025@ubuntu:~$ ls
Desktop    examples.desktop  Music        simple_ls    test
Documents  J                 Pictures     simple_ls.c  test2
Downloads  Makefile          Public       Templates    Videos
```

그 후 make명령을 통해 해당 c파일을 컴파일 한다.

Parameter가 default인경우

```
sp20157522025@ubuntu:~$ ./simple_ls
Desktop
Documents
Downloads
examples.desktop
J
Makefile
Music
Pictures
Public
simple_ls
simple_ls.c
Templates
test
test2
Videos
```

다음은 parameter가 없는 경우로 주어진 조건에 따라 현재 directory의 파일들의 목록을 print한다.

Parameter가 주어진 경우

```
sp20157522025@ubuntu:~$ ls test
a  abcd  A.c  aecc  asdkxcvn  B      Makefile  qweqxcv  test
A  aBcD  aCdf  apple  b          dgjf  pizza    sun!
sp20157522025@ubuntu:~$ ./simple_ls test
a
A
A.c
aBcD
abcd
aCdf
aecc
apple
asdkxcvn
b
B
dgjf
Makefile
pizza
qweqxcv
sun!
test
```

Parameter로 directory의 경로가 들어온 경우다.

Parameter가 상대경로인 경우

```
sp20157522025@ubuntu:~$ ls test2/ABCD
AA AB aBcd A.c ACbd BB B.c CA CC ET ls.c Makefile spls
sp20157522025@ubuntu:~$ ./simple_ls test2/ABCD
A.c
AA
AB
aBcd
ACbd
B.c
BB
CA
CC
ET
ls.c
Makefile
spls
```

상대경로로 '/'를 통하여 directory안에 directory도 탐색할 수 있어야 한다.

그 외의 경로들

다음은 그 외의 절대 경로와 현재 directory와 parent directory를 뜻하는 .. 과 . 표시를 통한 접근을 예로 들고 있다.

```
sp20157522025@ubuntu:~$ ./simple_ls ..
sp20157522025
sp20157522025@ubuntu:~$ ./simple_ls .
Desktop
Documents
Downloads
examples.desktop
J
Makefile
Music
Pictures
Public
simple_ls
simple_ls.c
Templates
test
test2
Videos
```

```
sp20157522025@ubuntu:~$ ./simple_ls /home
sp20157522025
```

```
sp20157522025@ubuntu:~$ ./simple_ls /boot
abi-4.15.0-29-generic
config-4.15.0-29-generic
config-4.15.0-46-generic
grub
initrd.img-4.15.0-29-generic
initrd.img-4.15.0-46-generic
memtest86+.bin
memtest86+.elf
memtest86+_multiboot.bin
retpoline-4.15.0-29-generic
System.map-4.15.0-29-generic
System.map-4.15.0-46-generic
vmlinuz-4.15.0-29-generic
vmlinuz-4.15.0-46-generic
```

예외처리 결과

```
sp20157522025@ubuntu:~$ ls
Desktop  ET          Makefile  Public    Templates Videos
Documents examples.desktop Music      simple_ls test
Downloads J           Pictures  simple_ls.c test2
sp20157522025@ubuntu:~$ ./simple_ls examples.dektop
simple_ls: cannot access 'examples.dektop' : No such directory
sp20157522025@ubuntu:~$ ./simple_ls Jeong
simple_ls: cannot access 'Jeong' : No such directory
sp20157522025@ubuntu:~$ ./simple_ls J ET
simple_ls: only one directory path can be processed
```

첫 번째부터 directory가 아닌 file에 접근할 때 error메시지를 print하며, 두 번째는 존재하지 않는 directory에 접근할 때 error메시지를 print하게 되며, 마지막으로 parameter가 두 개 이상 들어오는 경우의 예외처리다.

5. Conclusion

이번 과제는 리눅스를 사용하면서 처음으로 메인함수에 인자를 사용하여 명령어를 구현하는 과제였다. 우선 메인함수에 인자를 받는 방식 자체가 처음 접해보는 방법이며, argv 변수 자체가 2차원 배열로 되어있기 때문에 1학년때 배웠던 C언어의 문법을 다시 보는 계기가 되었다. 명령어 구현자체는 argc와 argv의 사용방법과 동적 할당이나 배열의 사용이 익숙하다면 쉽게 구현할 수 있는 과제였다. 특히 이번에 배우게 된 stat함수를 사용하여 파일을 읽어 파일의 이름을 받아올 수 있는 편리한 함수가 있는 것을 알게 되었고, 리눅스에서 개발을 할 때 굉장히 강력하게 쓰일 것으로 예상이 되었다.