

시스템 프로그래밍 실습

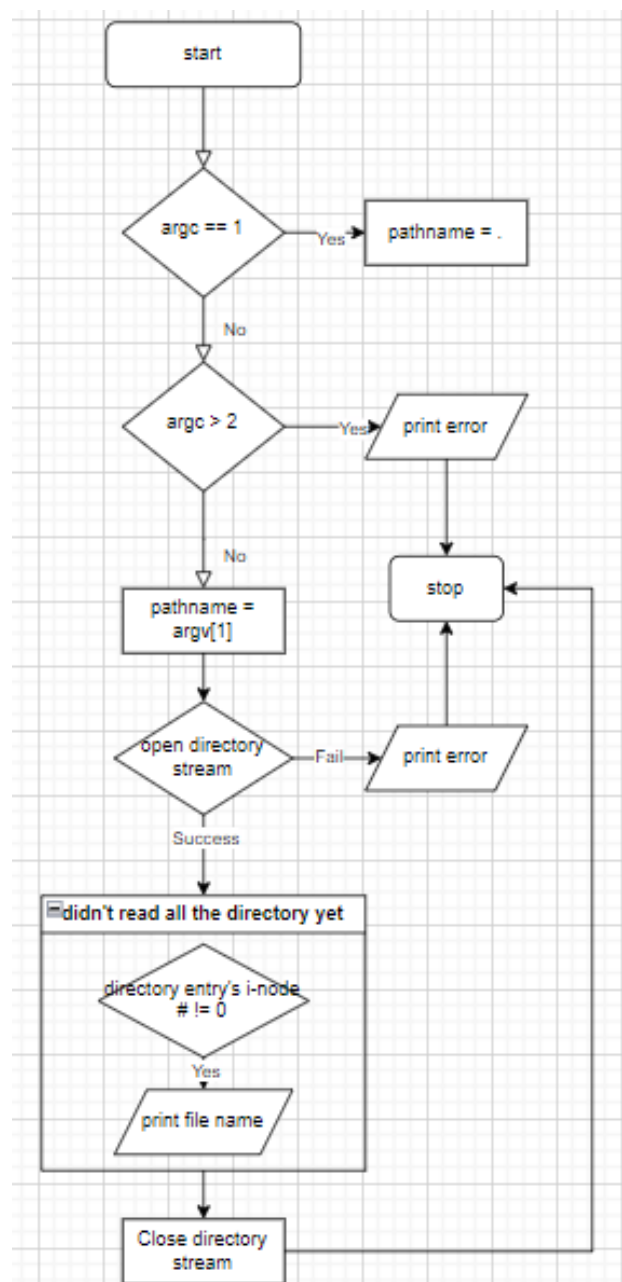
Assignment 1-2: ls

Class : 금요일 1, 2 교시
Professor : 최상호 교수님
Student ID : 2020202034
Name : 김태완

Introduction

이번 과제는 directory stream 을 열고 이를 통해 directory entry 를 읽어 ls 의 기능을 구현하는 프로그램을 만드는 과제입니다. directory stream 을 열고 이를 읽는 과정, 그리고 닫는 과정은 각각 opendir(), readdir(), closedir()가 필요합니다. 이 프로그램을 구현하는 과정에서 error handling 도 필요한데, 이는 opendir()이 directory stream 을 여는데 실패할 때 NULL 을 반환할 뿐만 아니라 errno 변수를 해당하는 error 로 세팅해주기 때문에 이를 이용해 해당하는 에러를 출력할 수 있습니다.

Flow chart



Pseudo code

1. start
2. If `argc(argument #) == 1`
 - Set pathname to current directory `(".")`
3. Else if `argc > 2`
 - Print error message
 - Stop
4. Else
 - Copy the first command-line argument (`argv[1]`) to pathname
5. Open the directory with pathname
6. If directory opening failed
 - Print error message indicating the failure
 - Stop
7. Iterate over each directory entry until it is all read
8. Within the loop, check if the directory entry is available
 - Print the name of the directory entry
9. Close the directory
10. Stop

결과화면

```
kw2020202034@ubuntu:~/system_programming_1/1-2$ make  
gcc -o testls kw2020202034 ls.c
```

makefile 을 이용해 소스코드를 컴파일하였습니다.

```
kw2020202034@ubuntu:~/system_programming_1/1-2$ ./testls /dev | tail  
kmsg  
urandom  
random  
full  
zero  
port  
null  
mem  
rfkill  
vga arbiter
```

루트 디렉토리에 있는 /dev directory 를 tail 명령어를 이용해 존재하는 하위 10 개의 파일명만 출력하도록 하였습니다.

```
kw2020202034@ubuntu:~/system_programming_1/1-2$ ./testls /var/spool/mqueue  
testls: cannot access '/var/spool/mqueue': No such directory
```

루트 디렉터리 아래 /var/spool 내의 존재하지 않는 디렉토리명을 인자로 주었을 때, 적절한 에러 메시지를 출력하는 것을 볼 수 있습니다.

```
kw2020202034@ubuntu:~/system_programming_1/1-2$ ./testls  
kw2020202034_ls.c  
Makefile  
testls  
..  
.
```

인자를 주지 않고 실행했을 때 현재 디렉토리 내의 파일명을 출력하는 것을 볼 수 있습니다.

```
kw2020202034@ubuntu:~/system_programming_1/1-2$ mkdir -m 364 testdir  
kw2020202034@ubuntu:~/system_programming_1/1-2$ ls -la  
total 40  
drwxrwxr-x 3 kw2020202034 kw2020202034 4096 Apr  8 19:46 .  
drwxrwxr-x 6 kw2020202034 kw2020202034 4096 Apr  8 19:17 ..  
-rw-rw-r-- 1 kw2020202034 kw2020202034 1869 Apr  8 19:39 kw2020202034_ls.c  
-rw-rw-r-- 1 kw2020202034 kw2020202034 128 Apr  4 20:08 Makefile  
d-wxrw-r-- 2 kw2020202034 kw2020202034 4096 Apr  8 19:46 testdir  
-rwxrwxr-x 1 kw2020202034 kw2020202034 17168 Apr  8 19:39 testls  
kw2020202034@ubuntu:~/system_programming_1/1-2$ ./testls testdir  
testls: cannot access 'testdir': Access denied
```

권한이 364(user=write, execute)인 디렉토리를 만들어 제가 read(directory 의 관점에서 ls)를 못 하게 만들고, 권한을 ls -la 로 확인하였습니다. 그 후, 이를 인자로 주었을 때, 'Access denied'라는 적절한 에러 메시지가 출력되었습니다.

```
kw2020202034@ubuntu:~/system_programming_1/1-2$ ./testls t1
testls: cannot access 't1': No such directory
```

또, 존재하지 않는 디렉토리 t1 을 인자로 주었을 때도 적절한 에러가 출력됨을 볼 수 있습니다.

```
kw2020202034@ubuntu:~/system_programming_1/1-2$ ./testls . ..
only one directory path can be processed
```

마지막으로 인자를 2 개(current directory, parent directory) 주었을 때도 적절한 에러가 출력됨을 볼 수 있습니다.

고찰

file stream 을 열어 이를 통해 file 의 내용을 읽고, 쓰는 코드는 전에 짜 보았는데, 디렉토리 또한 file stream 과 같은 방식으로 stream 을 열어 내용을 읽어, ls 를 직접 구현할 수 있다는 사실이 신기하였습니다. 그리고 error handling 을 할 때, 몇몇 함수는 각각의 error 상황을 의미하는 errno 변수 값을 errno 에 세팅한다는 사실을 알게 되어 이 errno 변수 값을 읽으면 해당 error 의 종류를 알아 error handling 을 직접 할 수 있다는 사실 또한 신기하였습니다.