

시스템 프로그래밍 실습

Assignment1-2

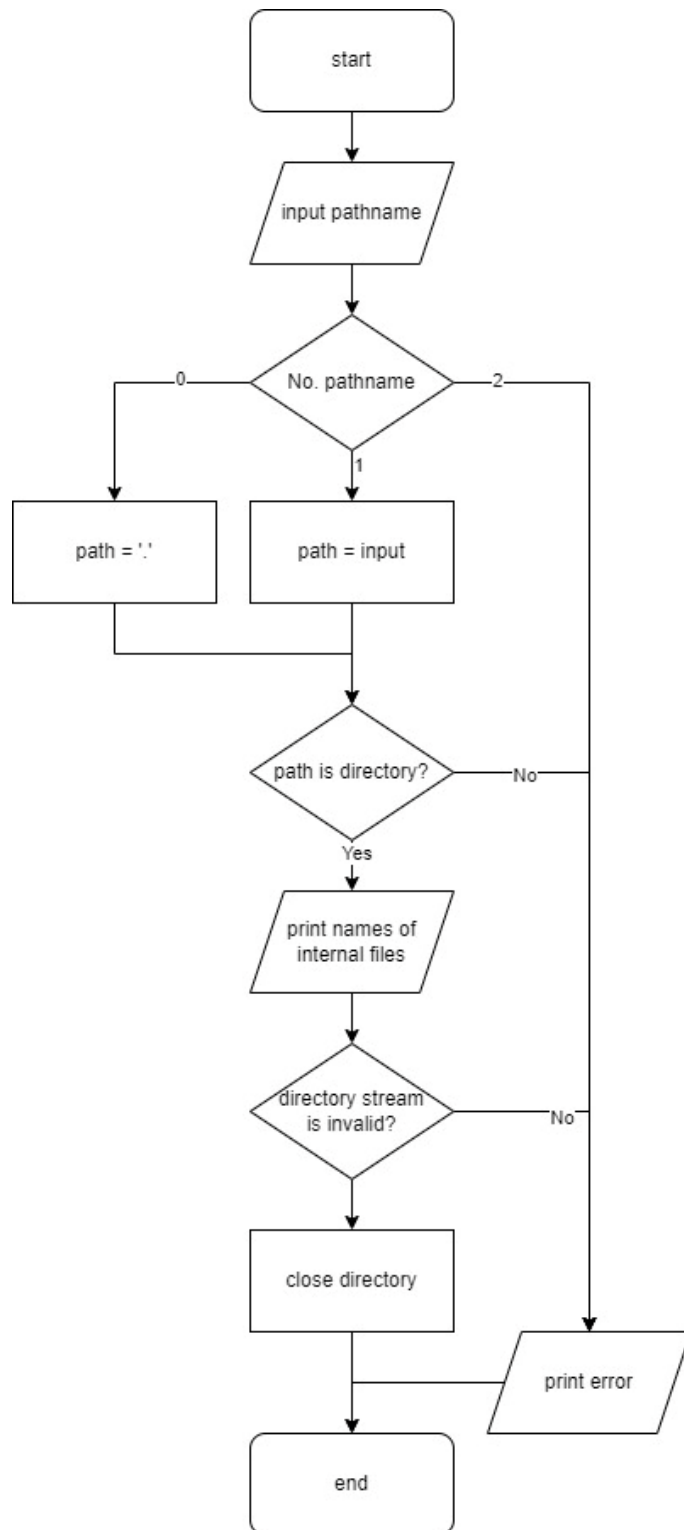
Class : 금 1, 2 분반
Professor : 최상호 교수님
Student ID : 2020202031
Name : 김재현

Introduction

이번 실습의 주요 목적은 `opendir()`, `readdir()`, `closedir()` 함수를 사용하여 리눅스에서 `ls` 명령어를 구현해보는 것입니다.

`opendir()` 함수를 사용하여 디렉토리를 열고, `readdir()` 함수를 사용하여 디렉토리 내용을 읽어오며, `closedir()` 함수를 사용하여 디렉토리를 닫습니다. 이를 통해 file system 을 이해함으로써 시스템 프로그래밍의 기초를 다지는 것이 이번 실습의 핵심입니다.

Flow chart



Pseudo code

```
int ls(char* pathname)
{
    DIR *dp;
    struct dirent *dirp;

    open directory named "pathname"

    if(directory is invalid) return -1;

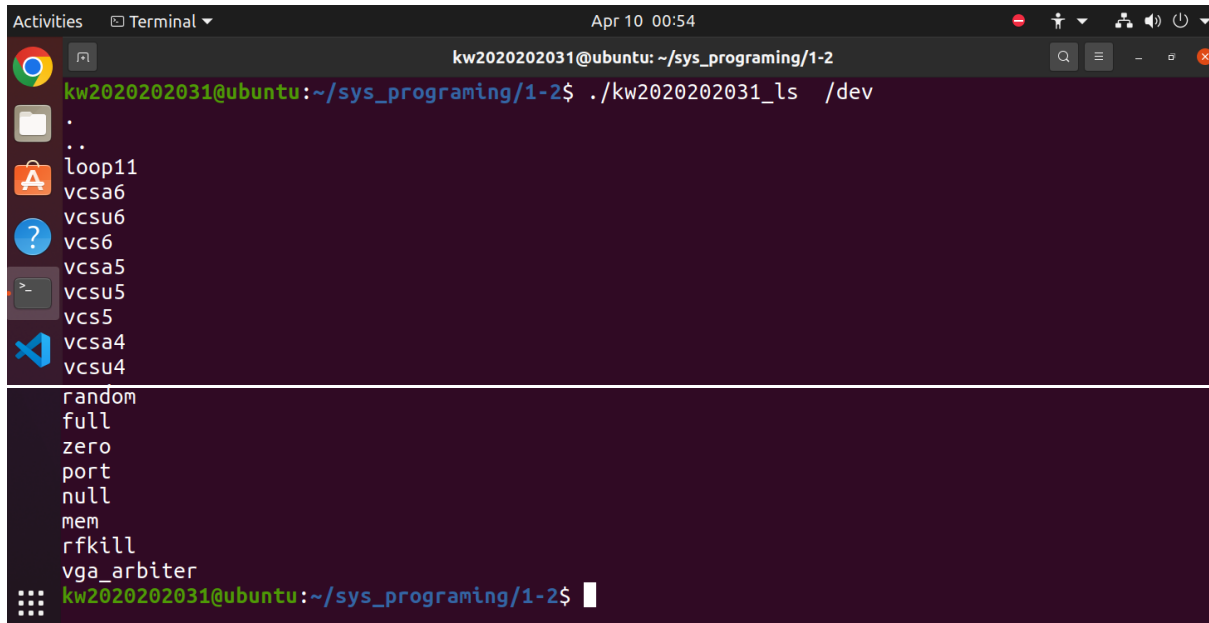
    while(all files under "pathname" directory)
        print the name of file

    close directory "pathname"

    if(directory stream is invalid) return -1

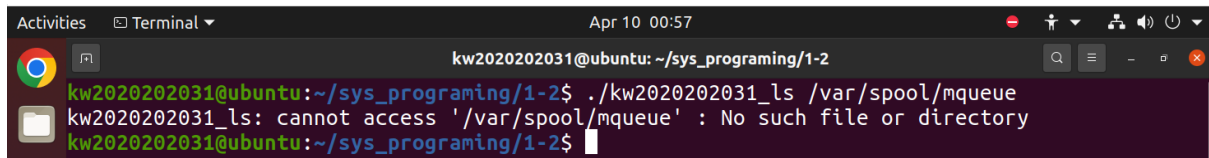
    return 0;
}
```

결과화면



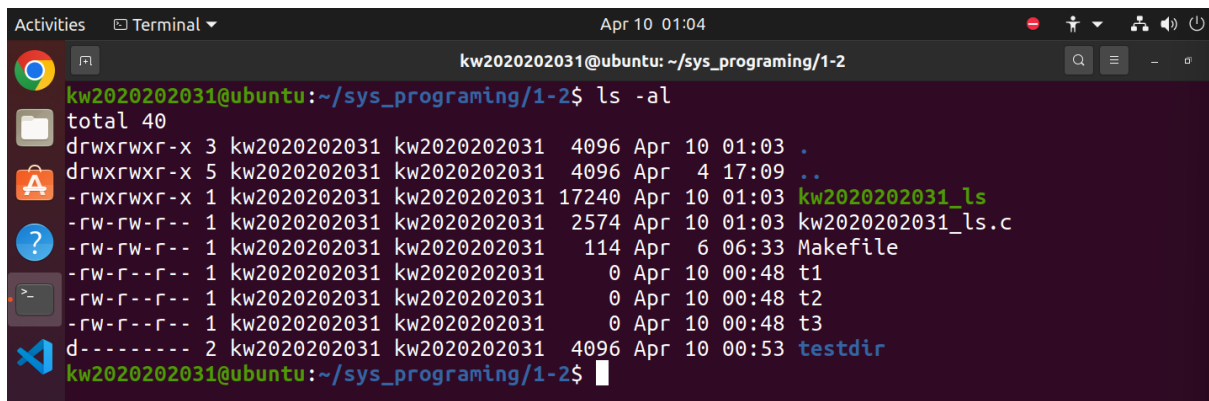
```
kw2020202031@ubuntu: ~/sys_programing/1-2
kw2020202031@ubuntu:~/sys_programing/1-2$ ./kw2020202031_ls /dev
.
..
loop11
vcsa6
vcsu6
vcs6
vcsa5
vcsu5
vcs5
vcsa4
vcsu4
random
full
zero
port
null
mem
rfkill
vga_arbiter
kw2020202031@ubuntu:~/sys_programing/1-2$
```

루트(/) 하위디렉토리 /dev 를 인자로 실행파일을 실행했더니 /dev 내부 파일들이 전부 출력되는 것을 확인할 수 있습니다.



```
kw2020202031@ubuntu: ~/sys_programing/1-2
kw2020202031@ubuntu:~/sys_programing/1-2$ ./kw2020202031_ls /var/spool/mqueue
kw2020202031_ls: cannot access '/var/spool/mqueue': No such file or directory
kw2020202031@ubuntu:~/sys_programing/1-2$
```

/var/spool/ 디렉토리 내에 mqueue 라는 이름의 파일이 존재하지 않기 때문에 No such file 오류가 출력되는 것을 확인할 수 있습니다.



```
kw2020202031@ubuntu: ~/sys_programing/1-2
kw2020202031@ubuntu:~/sys_programing/1-2$ ls -al
total 40
drwxrwxr-x 3 kw2020202031 kw2020202031 4096 Apr 10 01:03 .
drwxrwxr-x 5 kw2020202031 kw2020202031 4096 Apr 4 17:09 ..
-rwxrwxr-x 1 kw2020202031 kw2020202031 17240 Apr 10 01:03 kw2020202031_ls
-rw-rw-r-- 1 kw2020202031 kw2020202031 2574 Apr 10 01:03 kw2020202031_ls.c
-rw-rw-r-- 1 kw2020202031 kw2020202031 114 Apr 6 06:33 Makefile
-rw-r--r-- 1 kw2020202031 kw2020202031 0 Apr 10 00:48 t1
-rw-r--r-- 1 kw2020202031 kw2020202031 0 Apr 10 00:48 t2
-rw-r--r-- 1 kw2020202031 kw2020202031 0 Apr 10 00:48 t3
d----- 2 kw2020202031 kw2020202031 4096 Apr 10 00:53 testdir
kw2020202031@ubuntu:~/sys_programing/1-2$
```

```
Activities Terminal Apr 10 01:08
kw2020202031@ubuntu: ~/sys_programing/1-2
kw2020202031@ubuntu:~/sys_programing/1-2$ ls -al
total 40
drwxrwxr-x 3 kw2020202031 kw2020202031 4096 Apr 10 01:03 .
drwxrwxr-x 5 kw2020202031 kw2020202031 4096 Apr 4 17:09 ..
-rwxrwxr-x 1 kw2020202031 kw2020202031 17240 Apr 10 01:03 kw2020202031_ls
-rw-rw-r-- 1 kw2020202031 kw2020202031 2574 Apr 10 01:03 kw2020202031_ls.c
-rw-rw-r-- 1 kw2020202031 kw2020202031 114 Apr 6 06:33 Makefile
-rw-rw-r-- 1 kw2020202031 kw2020202031 0 Apr 10 00:48 t1
-rw-rw-r-- 1 kw2020202031 kw2020202031 0 Apr 10 00:48 t2
-rw-rw-r-- 1 kw2020202031 kw2020202031 0 Apr 10 00:48 t3
d----- 2 kw2020202031 kw2020202031 4096 Apr 10 00:53 testdir
kw2020202031@ubuntu:~/sys_programing/1-2$ ./kw2020202031_ls
t1
kw2020202031_ls.c
Makefile
.
t2
t3
..
kw2020202031_ls
testdir
kw2020202031@ubuntu:~/sys_programing/1-2$
```

실행파일 kw2020202031_ls 는 현재 디렉토리 내의 숨김파일까지 전부 출력하는 것을 확인할 수 있습니다.

```
Activities Terminal Apr 10 01:11
kw2020202031@ubuntu: ~/sys_programing/1-2
kw2020202031@ubuntu:~/sys_programing/1-2$ ./kw2020202031_ls /not_exist_dir
kw2020202031_ls: cannot access '/not_exist_dir' : No such file or directory
kw2020202031@ubuntu:~/sys_programing/1-2$ ./kw2020202031_ls testdir
kw2020202031_ls: cannot access 'testdir' : Permission denied
kw2020202031@ubuntu:~/sys_programing/1-2$ ./kw2020202031_ls t1
kw2020202031_ls: cannot access 't1' : Not a directory
kw2020202031@ubuntu:~/sys_programing/1-2$ ./kw2020202031_ls ./ ../
only one directory path can be processed
kw2020202031@ubuntu:~/sys_programing/1-2$
```

루트 디렉토리 내에 not_exist_dir 라는 이름의 파일이나 디렉토리가 존재하지 않기 때문에 No such file or directory 라는 오류 메시지가 출력되는 것을 확인할 수 있습니다.

현재 디렉토리 내에 testdir 이라는 디렉토리가 존재하나, 권한이 000 으로 r 권한이 없기 때문에, opendir()에서 Permission denied 오류 메시지를 출력하는 것을 확인할 수 있습니다.

현재 디렉토리 내에 t1 이라는 파일은 존재하나, 이는 디렉토리가 아니기 때문에 Not a directory 오류 메시지가 출력되는 것을 확인할 수 있습니다.

실행파일에 ./ , ../ 두 개의 인자가 전달됐기 때문에 only one directory path can be processed 오류 메시지가 출력되는 것을 확인할 수 있습니다.

고찰

opendir(), readdir(), closedir()을 활용하여 ls 명령어를 구현하는 것은 매우 간단했습니다. 하지만 에러가 발생한다면 opendir(), readdir(), closedir() 중 어느 부분에서 에러가 발생하는지, 어떤 에러가 발생했는지를 알 수 없었습니다.

system call 이나 라이브러리 함수가 호출될 때 발생하는 에러 정보를 저장하는 errno 이라는 변수가 errno.h 라이브러리 내에 존재한다는 것을 알게 되었고, strerror() 함수로 errno 에 부합하는 에러메시지를 알아내고, 이를 fprintf() 함수로 stderr 표준에러 버퍼를 통해 출력했습니다.

Reference

errno manual page

<https://android.googlesource.com/kernel/lk/+dima/for-travis/include/errno.h>

system call 에서 발생할 수 있는 error number 들이 매크로로 정의돼있는 것을 확인할 수 있습니다.

<https://man7.org/linux/man-pages/man3/strerror.3.html>

strerror 를 통해 errno 에 저장된 error number 에 맞는 에러 메시지를 출력할 수 있음을 확인할 수 있습니다.