

시스템 프로그래밍 실습

[Assignment1-2]

Class : [D]
Professor : [최상호 교수님]
Student ID : [2020202060]
Name : [홍왕기]

Introduction

이 실험에서는 `opendir()`, `readdir()`, `closedir()` 함수를 사용하여 FTP 서버에서 필요한 `ls` 명령어를 직접 구현할 것이다.

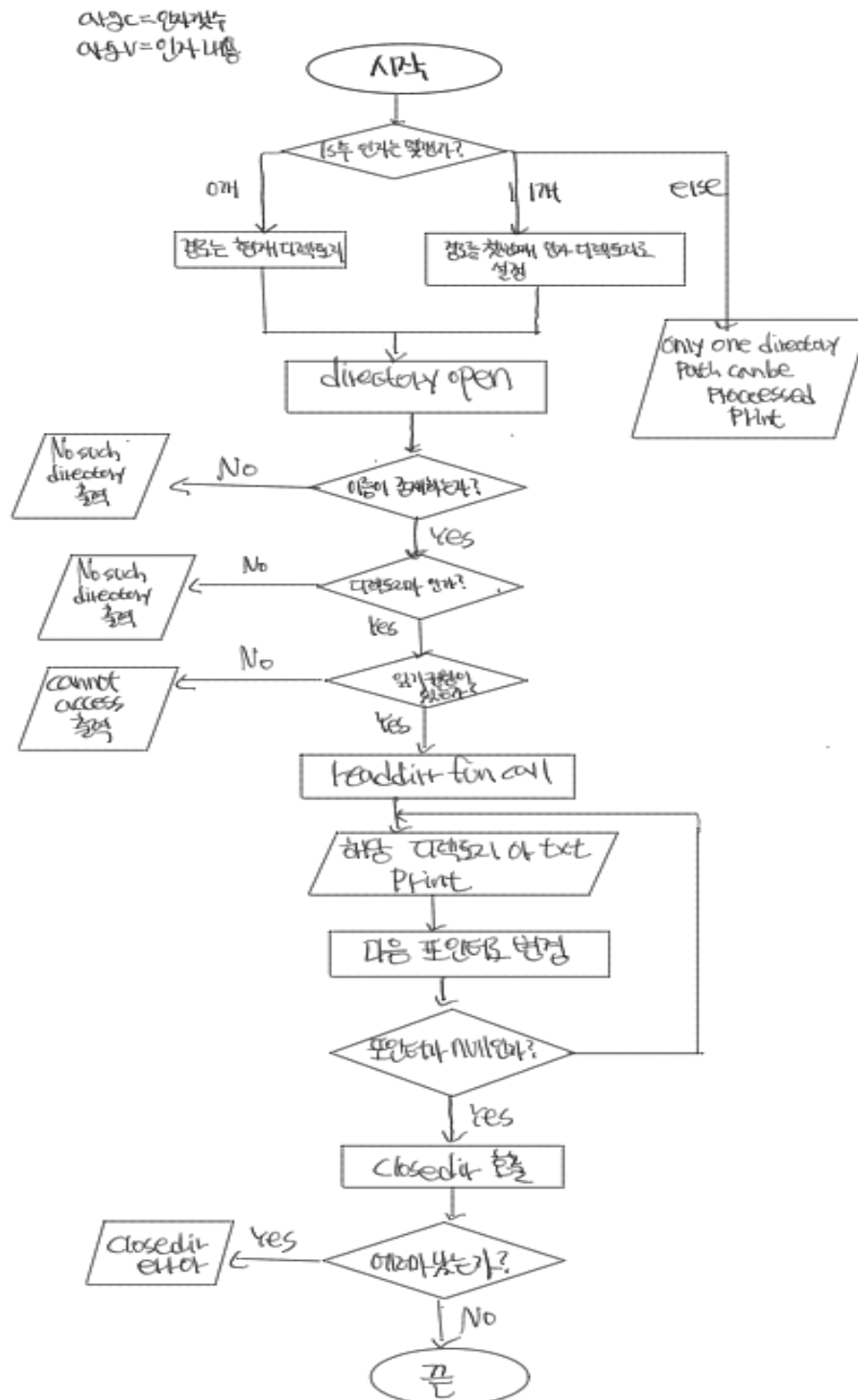
먼저, `opendir()` 함수는 디렉토리 스트림을 열기 위해 사용된다. 이 함수는 주어진 경로의 디렉토리를 열고, 해당 디렉토리 스트림에 대한 핸들을 반환한다.

그 후, `readdir()` 함수는 디렉토리 내의 다음 파일 또는 디렉토리 항목을 반환한다. 각 호출은 디렉토리 스트림에서 다음 항목을 읽어와 이 함수는 디렉토리 항목의 이름과 추가 정보를 제공하는 것이다.

마지막으로, `closedir()` 함수는 `opendir()`로 열린 디렉토리 스트림을 닫는 데 사용된다. 디렉토리 스트림을 닫으면 해당 핸들과 연결된 리소스가 해제된다.

이러한 함수들을 사용하여 `ls` 명령어를 프로그래밍함으로써, FTP 서버를 구현할 때 디렉토리 내의 파일 및 하위 디렉토리를 나열하는 기능을 구현할 수 있을 것이다. 이를 구현함으로써 최종적인 ftp 서버를 개발하는데 필요한 `ls` 명령어를 구현하여 최종적인 목표에 다가간다.

Flow chart



Pseudo code

```
int main(int argc, char const *argv[])
{
    // Define a pointer to store the directory path
    *directory = "./"

    // Check if the user provided a directory path as a command-line argument
    // Provided directory path as command-line argument, set directory to the provided path
    if (argc == 2) directory = argv[1];
    else if (argc > 2) print "only one directory path can be processed" return;

    // Declare a struct to hold information about the directory
    struct stat dir_stat;

    // Open the directory for reading
    DIR *dp = opendir(directory);
    if (dp == NULL)
    {
        // Check if the specified directory exists
        if directory_path does not exist :
            | print "kw2020202060_ls cannot access directory: No such directory" return
        if no directory
            | printf "kw2020202060_ls: cannot access '%s': No such directory" return ;
        // Check if the program has read and execute permissions on the directory
        if program does not have read and execute permissions on directory_path :
            | print "kw2020202060_ls: Cannot access directory: Access denied" return
    }
}
```

```
// Declare a pointer to a directory entry
struct dirent *entry;

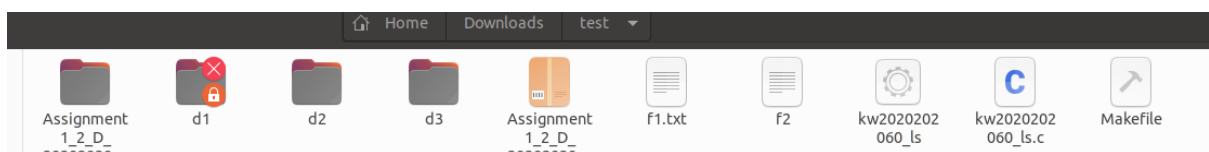
// Iterate over each directory entry and print its name
while ((entry = readdir(dp)) != NULL) print (entry name)
// Close the directory
close directory
// Return 0 to indicate successful execution of the program
return
```

결과화면

```
kw2020202060@ubuntu:~/Downloads/test$ ls -al
total 652
drwxrwxr-x 6 kw2020202060 kw2020202060 4096 Apr 10 02:43 .
drwxr-xr-x 5 kw2020202060 kw2020202060 4096 Apr 7 00:53 ..
drwxrwxr-x 2 kw2020202060 kw2020202060 4096 Apr 10 02:44 Assignment1_2_D_2020202060
-rw-rw-r-- 1 kw2020202060 kw2020202060 612663 Apr 9 06:12 Assignment1_2_D_2020202060_홍왕기.tar.gz
d----- 2 kw2020202060 kw2020202060 4096 Apr 5 01:50 d1
drwxrwxrwx 3 kw2020202060 kw2020202060 4096 Apr 5 03:53 d2
drwxrwxr-x 2 kw2020202060 kw2020202060 4096 Apr 5 01:50 d3
-rw-rw-r-- 1 kw2020202060 kw2020202060 0 Apr 8 18:30 f1.txt
-rw-rw-r-- 1 kw2020202060 kw2020202060 0 Apr 8 18:31 f2
-rwxrwxr-x 1 kw2020202060 kw2020202060 17168 Apr 10 02:43 kw2020202060_ls
-rw-rw-r-- 1 kw2020202060 kw2020202060 4022 Apr 10 02:43 kw2020202060_ls.c
-rw-rw-r-- 1 kw2020202060 kw2020202060 77 Apr 7 00:03 Makefile
```

먼저 test 경로에 대한 내용들이다. D1, d2, d3, 실행파일, .c , make file, 여러가지 파일 등이 존재하는 것을 볼 수 있다. 여기서 중요한 점은 d1 디렉토리는 아무 권한이 없다.

```
kw2020202060@ubuntu:~/Downloads/test$ ./kw2020202060_ls
d2
f1.txt
kw2020202060_ls
f2
..
kw2020202060_ls.c
Assignment1_2_D_2020202060_홍왕기.tar.gz
Makefile
.
Assignment1_2_D_2020202060
d3
d1
kw2020202060@ubuntu:~/Downloads/test$
```



위 결과를 확인하면 ls 뒤 아무 인자가 없으므로 test 경로에 대한 내용이 출력되는 것을 볼 수 있다.

```
kw2020202060@ubuntu:~/Downloads/test$ ./kw2020202060_ls /dev
.
..
loop9
vcsa6
vcsu6
vcs6
vcsa5
vcsu5
vcs5
vcsa4
vcsu4
vcs4
```

↓ 더중재

```
kw2020202060@ubuntu:~/Downloads/test$ ls /dev
autofs      log          sda1         tty24        tty51        ttyS2        vcs5
block       loop0        sda2         tty25        tty52        ttyS20       vcs6
bsg         loop1        sda5         tty26        tty53        ttyS21       vcsa
btrfs-control loop2        sg0          tty27        tty54        ttyS22       vcsa1
bus         loop3        sg1          tty28        tty55        ttyS23       vcsa2
cdrom       loop4        shm          tty29        tty56        ttyS24       vcsa3
cdrw        loop5        snapshot     tty3         tty57        ttyS25       vcsa4
char        loop6        snd          tty30        tty58        ttyS26       vcsa5
console     loop7        sr0          tty31        tty59        ttyS27       vcsa6
core        loop8        stderr       tty32        tty6         ttyS28       vcsu
cpu         loop9        stdin        tty33        tty60        ttyS29       vcsu1
cpu_dma_latency loop-control stdout        tty34        tty61        ttyS3        vcsu2
cuse        mapper      tty          tty35        tty62        ttyS30       vcsu3
```

↓ 더중재

위는 dev 디렉토리에있는 내용들이 출력됨을 볼 수 있다. 이 파일들이 맞는가를 확인하기 위해서 본래의 ls 명령어를 통해 같은 내용들이 출력되는 것을 볼 수 있다.

```
d1
kw2020202060@ubuntu:~/Downloads/test$ ./kw2020202060_ls /not_exist_dir
kw2020202060_ls: cannot access '/not_exist_dir': No such directory
kw2020202060@ubuntu:~/Downloads/test$
```

위 테스트는 없는 디렉토리에 대한 내용을 출력하는 케이스이다. 위를 확인하면 없는 디렉토리임을 출력해주는 형태를 확인할 수 있다.

```
kw2020202060_ls: cannot access '/not_exist_dir': No such directory
kw2020202060@ubuntu:~/Downloads/test$ ./kw2020202060_ls d1
kw2020202060_ls: cannot access 'd1': Access denied
kw2020202060@ubuntu:~/Downloads/test$
```

위는 접근 제한된 디렉토리에 대한 예외를 처리하는 case 이다. 위 제공된 사진과 같이 d1 은 아무 접근을 못하는 디렉토리임을 확인하였다. 따라서 이 디렉토리에 대한 내용을 출력했을 때 접근이 제한되었다는 출력을 확인할 수 있다.

```

kw2020202060@ubuntu:~/Downloads/test$ ./kw2020202060_ls d4
kw2020202060_ls: cannot access 'd4': No such directory
kw2020202060@ubuntu:~/Downloads/test$ ./kw2020202060_ls d2
Dtest
Dtest1
..
test1.txt
.
kw2020202060@ubuntu:~/Downloads/test$

```

위 테스트는 하위 디렉토리에 대한 내용을 출력하는 case 이다. test 경로에 대한 하위에 d4 디렉토리를 출력해보려고 할 때, d4 디렉토리는 존재하지 않으므로 오류 출력이 되는 것을 볼 수 있다. 반대로 d2 디렉토리가 있는 경우에는 d2 디렉토리 안에 있는 내용들이 출력됨을 볼 수 있다.

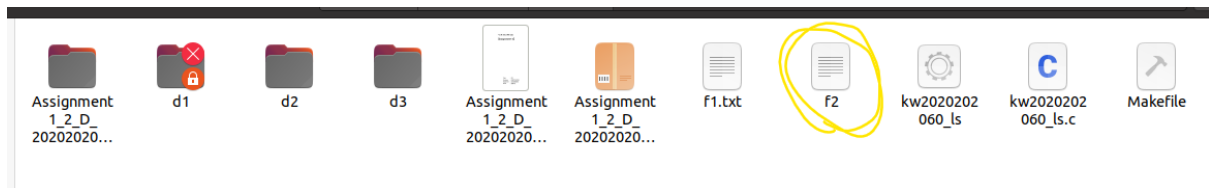
```

kw2020202060@ubuntu:~/Downloads/test$ ./kw2020202060_ls d2
Dtest
Dtest1
..
test1.txt
.
kw2020202060@ubuntu:~/Downloads/test$ ./kw2020202060_ls d2/Dtest1
test5.txt
..
.
kw2020202060@ubuntu:~/Downloads/test$ ./kw2020202060_ls ./ ../
only one directory path can be processed
kw2020202060@ubuntu:~/Downloads/test$

```

위 결과는 d2 디렉토리 안에 내용을 출력한 것을 볼 수 있다. 이때 d2/Dtest1 를 통해서 d2 디렉토리 안에 있는 Dtest1 디렉토리를 검색한 결과 Dtest1 안에 내용들이 출력됨을 볼 수 있다. 반면에 명령어 입력 후 두개 이상에 인자가 들어온다면 하나에 디렉토리만 검색이 가능하다는 오류가 출력됨을 확인할 수 있다.

추가로 디렉토리가 아닌 파일인 경우에 테스트를 실행한다. 다음 과 같은 파일이 있다고 가정하자.



이를 테스트하면 다음과 같다.

```
kw2020202060@ubuntu:~/Downloads/test$ ls -al
total 652
drwxrwxr-x 6 kw2020202060 kw2020202060 4096 Apr 10 02:43 .
drwxr-xr-x 5 kw2020202060 kw2020202060 4096 Apr 7 00:53 ..
drwxrwxr-x 2 kw2020202060 kw2020202060 4096 Apr 10 02:44 Assignment1_2_D_2020202060
-rw-rw-r-- 1 kw2020202060 kw2020202060 612663 Apr 9 06:12 Assignment1_2_D_2020202060_홍왕기.tar.gz
d----- 2 kw2020202060 kw2020202060 4096 Apr 5 01:50 d1
drwxrwxrwx 3 kw2020202060 kw2020202060 4096 Apr 5 03:53 d2
drwxrwxr-x 2 kw2020202060 kw2020202060 4096 Apr 5 01:50 d3
-rw-rw-r-- 1 kw2020202060 kw2020202060 0 Apr 8 18:30 f1.txt
-rw-rw-r-- 1 kw2020202060 kw2020202060 0 Apr 8 18:31 f2
-rwxrwxr-x 1 kw2020202060 kw2020202060 17168 Apr 10 02:43 kw2020202060_ls
-rw-rw-r-- 1 kw2020202060 kw2020202060 4022 Apr 10 02:43 kw2020202060_ls.c
-rw-rw-r-- 1 kw2020202060 kw2020202060 77 Apr 7 00:03 Makefile
kw2020202060@ubuntu:~/Downloads/test$ ./kw2020202060_ls f2
kw2020202060_ls: cannot access 'f2': No such directory
```

위 결과를 확인하면 f2 는 directory 가 아닌 것으로 출력된다.

다음 테스트는 spool 경로 안에 있는 디렉토리인 mqueue 디렉토리를 출력해보는 케이스이다. 이는 아래와 같이 아무 권한이 없는 디렉토리이며, 실행은 아래와 같다.

```
kw2020202060@ubuntu:~/Downloads/test$ cd /var/spool
kw2020202060@ubuntu:/var/spool$ ls -al
total 28
drwxr-xr-x 7 root root 4096 Apr 9 05:38 .
drwxr-xr-x 14 root root 4096 Mar 16 2023 ..
drwxr-xr-x 2 root root 4096 Mar 7 02:35 anacron
drwxr-xr-x 3 root root 4096 Mar 16 2023 cron
drwx--x-- 3 root lp 4096 Mar 16 2023 cups
lrwxrwxrwx 1 root root 7 Mar 7 02:21 mail -> ../mail
d----- 2 root root 4096 Apr 9 05:38 mqueue
drwx----- 2 syslog adm 4096 Feb 11 2020 rsyslog
kw2020202060@ubuntu:/var/spool$ cd -
/home/kw2020202060/Downloads/test
kw2020202060@ubuntu:~/Downloads/test$ ./kw2020202060_ls /var/spool/mqueue
kw2020202060_ls: cannot access '/var/spool/mqueue': Access denied
kw2020202060@ubuntu:~/Downloads/test$
```

위를 확인하면 mqueue 디렉토리는 읽기권한이 없기에 접근이 불가능한 결과를 볼 수 있다.

고찰

"ls" 기능을 구현하는 과정에서 파일 시스템 및 시스템 호출에 대한 이해 부족과 오류 처리에 어려움을 겪었다. 파일 시스템 관련 함수 및 이해와 시스템 호출의 사용법을 익히는 데 시간이 많이 걸렸고, 디렉토리나 파일에 대한 적절한 권한 확인과 오류 처리를 위한 구현하는 과정에도 어려움을 겪었다. 하지만 강의자료를 통해서 함수에 대한 이해도를 높이고 오랜 시간을 사용하여 해결할 수 있었다. 이번 프로젝트를 통해 파일 시스템과 관련된 개념과 리눅스 안 명령어들이 어떤 식으로 구현 되어있을지 알 수 있게 되는 경험이었다.