

12201922 이규민

3. Homework

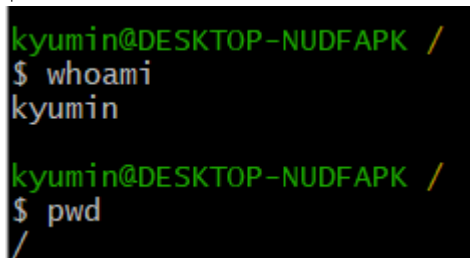
3.1 Do following steps.

1) Install VirtualBox+ubuntu or Cygwin in your PC (For Mac, open the terminal window and do the homework there). Refer to lect1 for installing Cygwin. For ubuntu, refer to Internet.

2) Log in to a Linux server. Find your login directory with "pwd". Find your user ID with "whoami".

```
$ pwd
```

```
$ whoami
```

A terminal window with a black background and green text. The prompt is 'kyumin@DESKTOP-NUDFAPK /'. The user enters '\$ whoami' and the output is 'kyumin'. Then the user enters '\$ pwd' and the output is '/'.

```
kyumin@DESKTOP-NUDFAPK /  
$ whoami  
kyumin  
  
kyumin@DESKTOP-NUDFAPK /  
$ pwd  
/
```

Whoami 를 통해 이름 확인이 가능하다. Pwd 를 통해 현재 directory 확인이 가능하다.

3) Make a file, confirm the contents, and check its permission mode

```
$ echo hello > f1
```

```
$ cat f1
```

```
hello
```

```
$ ls -l
```

```
.....  
-rw-r--r-- 1 inha inha 6 Feb 27 11:47 f1  
.....
```

-: this file is a regular file (d if a directory)

rw-: the owner can read from or write into this file but cannot execute it

r-- : the group member can read but has no write/execute permission

r-- : other people can read but has no write/execute permission

1 : there is one link to this file

inha : the user ID of the owner of this file

inha : the group ID of of the owner of this file

6 : the byte size of this file

Feb 27 11:47 : the creation time or last modification time of this file

f1 : file name

```

kyumin@DESKTOP-NUDFAPK /
$ echo hello > f1

kyumin@DESKTOP-NUDFAPK /
$ ls
Cygwin-Terminal.ico  Cygwin.ico  cygdrive  etc  home  proc  tmp  var
Cygwin.bat          bin        dev      f1   lib   sbin  usr

kyumin@DESKTOP-NUDFAPK /
$ cat f1
hello

kyumin@DESKTOP-NUDFAPK /
$ ls -l
total 366
-rw-r--r-- 1 kyumin Administrators 53342 Mar 1 17:20 Cygwin-Terminal.ico
-rwxr-xr-x 1 kyumin Administrators 88 Mar 1 17:18 Cygwin.bat
-rw-r--r-- 1 kyumin Administrators 157097 Mar 1 17:20 Cygwin.ico
drwxr-xr-x 1 kyumin Administrators 0 Mar 1 17:19 bin
dr-xr-xr-x 1 kyumin 없음 0 Mar 4 10:08 cygdrive
drwxr-xr-x 1 kyumin Administrators 0 Mar 1 17:16 dev
drwxr-xr-x 1 kyumin Administrators 0 Mar 1 17:19 etc
-rw-r--r-- 1 kyumin 없음 6 Mar 4 10:06 f1
drwxrwxrwt 1 kyumin Administrators 0 Mar 1 17:16 home
drwxr-xr-x 1 kyumin Administrators 0 Mar 1 17:17 lib
dr-xr-xr-x 9 kyumin 없음 0 Mar 4 10:08 proc
drwxr-xr-x 1 kyumin Administrators 0 Mar 1 17:17 sbin
drwxrwxrwt 1 kyumin Administrators 0 Mar 1 17:16 tmp
drwxr-xr-x 1 kyumin Administrators 0 Mar 1 17:17 usr
drwxr-xr-x 1 kyumin Administrators 0 Mar 1 17:17 var

```

Echo hello > f1 은 f1 파일의 내용을 hello 라고 쓸 수 있다. Cat f1 은 f1 의 내용을 출력한다. Ls 는 현재 디렉토리 안에 있는 디렉토리와 파일들을 볼 수 있고, -l 옵션을 추가하면 파일들의 권한, 오너 ID, 그룹원 ID,바이트, 최종수정시간을 볼 수 있다. 또한 첫글자를 통해 디렉토리인지 파일인지 구별이 가능하다.

4) Change the permission mode of f1

The current mode is 110 100 100 (644). 1 means enabled while 0 disabled. First three bits are for the owner, the next three for group members, and the last three for all others. 110 for the owner means read, write bits are enabled but not execute bit; so the owner can read or write this file but cannot execute it.

```
$ chmod 000 f1
```

```
$ ls -l
```

```

.....
----- 1 inha inha 6 Feb 27 11:47 f1
.....

```

Now no one including the owner cannot read, write, nor execute this file.

```
$ cat f1
```

```
cat: f1: Permission denied
```

```
$ chmod 777 f1
```

```
$ ls -l
```

```

.....
-rwxrwxrwx 1 inha inha 6 Feb 27 11:47 f1
.....

```

Now anyone can read/write/execute this file

```
kyumin@DESKTOP-NUDFAPK /cygdrive/c/Users/kyumin/AppData
$ ls -l
total 25
drwxrwx---+ 1 kyumin SYSTEM 0 Mar  1 18:24 Local
drwxrwx---+ 1 SYSTEM SYSTEM 0 Mar  1 18:24 LocalLow
drwxrwx---+ 1 kyumin SYSTEM 0 Dec  3 12:12 Roaming
-rwxrwxrwx+ 1 kyumin 없음  3 Mar  5 13:02 f1

kyumin@DESKTOP-NUDFAPK /cygdrive/c/Users/kyumin/AppData
$ chmod 000 f1

kyumin@DESKTOP-NUDFAPK /cygdrive/c/Users/kyumin/AppData
$ ls
Local LocalLow Roaming f1

kyumin@DESKTOP-NUDFAPK /cygdrive/c/Users/kyumin/AppData
$ ls -l
total 25
drwxrwx---+ 1 kyumin SYSTEM 0 Mar  1 18:24 Local
drwxrwx---+ 1 SYSTEM SYSTEM 0 Mar  1 18:24 LocalLow
drwxrwx---+ 1 kyumin SYSTEM 0 Dec  3 12:12 Roaming
-----+ 1 kyumin 없음  3 Mar  5 13:02 f1

kyumin@DESKTOP-NUDFAPK /cygdrive/c/Users/kyumin/AppData
$ cat f1
cat: f1: Permission denied
```

f1 파일 생성 후 chmod 000 을 이용하여 모든 권한을 없앴고, 읽기/쓰기/실행 진행 시 권한 거부 메시지가 뜬다. Ls -l 명령어를 사용하여 f1 의 모든 권한이 없어진 것을 알 수 있다.

```
kyumin@DESKTOP-NUDFAPK /cygdrive/c/Users/kyumin/AppData
$ chmod 777 f1

kyumin@DESKTOP-NUDFAPK /cygdrive/c/Users/kyumin/AppData
$ ls -l
total 25
drwxrwx---+ 1 kyumin SYSTEM 0 Mar  1 18:24 Local
drwxrwx---+ 1 SYSTEM SYSTEM 0 Mar  1 18:24 LocalLow
drwxrwx---+ 1 kyumin SYSTEM 0 Dec  3 12:12 Roaming
-rwxrwxrwx+ 1 kyumin 없음  3 Mar  5 13:02 f1

kyumin@DESKTOP-NUDFAPK /cygdrive/c/Users/kyumin/AppData
$ cat f1
hi
```

Chmod 777 를 이용하여 오너, 같은 그룹원, 기타 사람들 모두에게 읽기/쓰기/실행 권한을 줄 수 있고 f1 읽기를했을 때 오류가 발생하지 않는 것을 볼 수 있다.

5) Find out your current location again with "pwd".

```
$ pwd
```

```
kyumin@DESKTOP-NUDFAPK /cygdrive/c/Users/kyumin/AppData
$ pwd
/cygdrive/c/Users/kyumin/AppData
```

Pwd 를 통해 현 위치를 확인했다.

6) Go to the root directory with "cd /" and make sure you are really at the top directory with "pwd".

```
$ cd /
```

```
$ pwd
```

```
kyumin@DESKTOP-NUDFAPK /cygdrive/c/Users/kyumin/AppData
$ cd /

kyumin@DESKTOP-NUDFAPK /
$ pwd
/
```

Cd / 를 통해 Root 디렉토리에서 이동했고 pwd 를 이용하여 이동된걸 볼 수 있다.

7) How many files you have in the root directory? Some of them are not directory files. Find them with "ls -l". Use "file" command to see more detailed info.

```
$ ls -l
```

```
$ file *
```

```
$ file lib
```

```
kyumin@DESKTOP-NUDFAPK /
$ ls -l
total 365
-rw-r--r-- 1 kyumin Administrators 53342 Mar  1 17:20 Cygwin-Terminal.ico
-rwxr-xr-x 1 kyumin Administrators   88 Mar  1 17:18 Cygwin.bat
-rw-r--r-- 1 kyumin Administrators 157097 Mar  1 17:20 Cygwin.ico
drwxr-xr-x 1 kyumin Administrators    0 Mar  1 17:19 bin
dr-xr-xr-x 1 kyumin 없음           0 Mar  5 13:09 cygdrive
drwxr-xr-x 1 kyumin Administrators    0 Mar  1 17:16 dev
drwxr-xr-x 1 kyumin Administrators    0 Mar  1 17:19 etc
drwxrwxrwt 1 kyumin Administrators    0 Mar  1 17:16 home
drwxr-xr-x 1 kyumin Administrators    0 Mar  1 17:17 lib
dr-xr-xr-x 9 kyumin 없음           0 Mar  5 13:09 proc
drwxr-xr-x 1 kyumin Administrators    0 Mar  1 17:17 sbin
drwxrwxrwt 1 kyumin Administrators    0 Mar  1 17:16 tmp
drwxr-xr-x 1 kyumin Administrators    0 Mar  1 17:17 usr
drwxr-xr-x 1 kyumin Administrators    0 Mar  1 17:17 var

kyumin@DESKTOP-NUDFAPK /
$ file *
Cygwin-Terminal.ico: MS Windows icon resource - 6 icons, 16x16, 32 bits/pixel, 2
4x24, 32 bits/pixel
Cygwin.bat:          DOS batch file, ASCII text, with CRLF line terminators
Cygwin.ico:          MS Windows icon resource - 7 icons, -104x-104, 32 bits/pixe
l, 16x16, 32 bits/pixel
bin:                 directory
cygdrive:            directory
dev:                 directory
etc:                 directory
home:                sticky, directory
lib:                 directory
proc:                directory
sbin:                directory
tmp:                 sticky, directory
usr:                 directory
var:                 directory

kyumin@DESKTOP-NUDFAPK /
$ file lib
lib: directory
```

Ls -l 을 이용하여 14 개의 파일이 있다는 것을 알 수 있다. File 명령어는 파일의 타입을 보여주는 명령어고, file * 은 현재 위치에 있는 모든 파일의 타입을 보여주는 것이다. 디렉토리 파일은 11 개 인 것을 알 수 있다. File lib 는 lib 라는 파일의 타입을 보여주는 것이고, lib 는 디렉토리인 것을 알 수 있다.

8) * is a wild card character meaning it will be replaced by all file names in the current directory.

```
$ file *
```

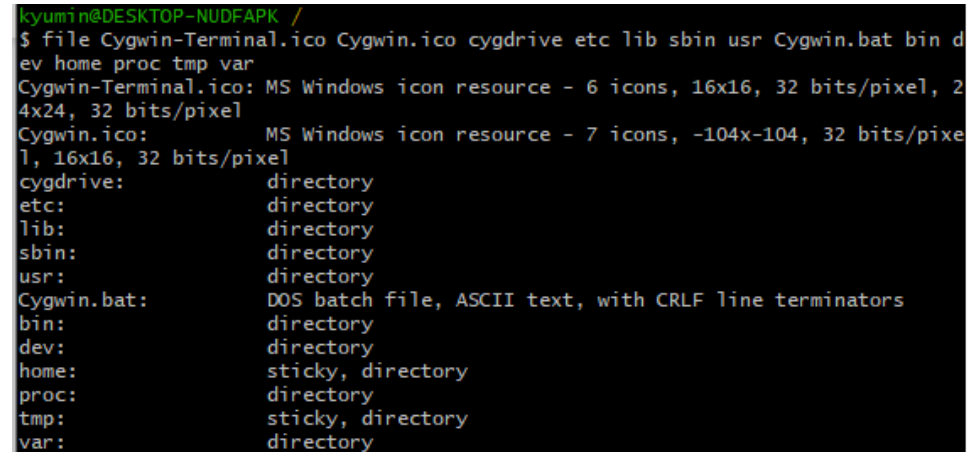
is same as

```
$ file app bin boot dev .... (all file names in the current directory)
```

```
$ file b*
```

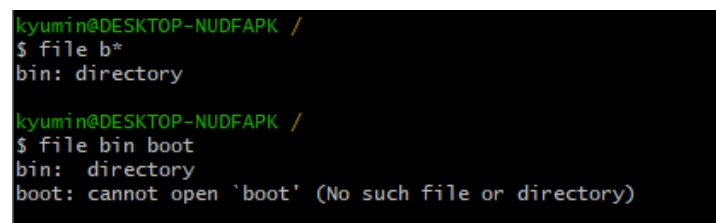
is same as

```
$ file bin boot
```



```
kyumin@DESKTOP-NUDFAPK /
$ file Cygwin-Terminal.ico Cygwin.ico cygdrive etc lib sbin usr Cygwin.bat bin d
ev home proc tmp var
Cygwin-Terminal.ico: MS Windows icon resource - 6 icons, 16x16, 32 bits/pixel, 2
4x24, 32 bits/pixel
Cygwin.ico: MS Windows icon resource - 7 icons, -104x-104, 32 bits/pixe
l, 16x16, 32 bits/pixel
cygdrive: directory
etc: directory
lib: directory
sbin: directory
usr: directory
Cygwin.bat: DOS batch file, ASCII text, with CRLF line terminators
bin: directory
dev: directory
home: sticky, directory
proc: directory
tmp: sticky, directory
var: directory
```

File Cygwin-Terminal.ico Cygwin.ico ... 은 file * 명령어와 같은 결과가 나온다. 7 번 결과와 순서는 다르지만 파일들의 타입 결과는 모두 동일한 것을 알 수 있다.



```
kyumin@DESKTOP-NUDFAPK /
$ file b*
bin: directory

kyumin@DESKTOP-NUDFAPK /
$ file bin boot
bin: directory
boot: cannot open 'boot' (No such file or directory)
```

File b*은 b로 시작하는 파일들의 타입을 보여주는 명령어다. 현 디렉토리에 b로 시작하는 파일을 bin 밖에 없다. 그러므로 file *과 file bin 은 동일한 결과가 나온다.

(문제에 첨부된 예시 건은 file bin boot 이지만 내 PC 에는 boot 디렉토리가 없어 오류가 발생하였다.)

9) You can move around the file tree with cd.

Use just "cd" without path to go back to the login directory.

```
$ cd
```

```
$ pwd
```

```
/home/inha
```

```
$ cd /
```

```
$ pwd
```

```
/
```

```
$ cd
```

```
$ pwd
```

```
/home/inha
```

Make some directories and move inside.

```
$ mkdir d1
```

```
$ cd d1
```

```
$ pwd
```

```
/home/inha/d1
```

Use ".." to move up.

```
$ cd ..
```

```
$ pwd
```

```
/home/inha
```

```
$ cd ..
```

```
$ pwd
```

```
/home
```

You can go to any location using "absolute path". A path is absolute whenever it starts with "/".

```
$ cd /home/inha/d1
```

```
$ pwd
```

```
/home/inha/d1
```

If the path doesnot start with "/", it is "relative path". The system will find the destination starting at the current location.

```
$ cd
```

```
$ pwd
```

```
/home/inha
```

```
$ cd home/inha/d1
```

```
-bash: cd: home/inha/d1: No such file or directory
```

"home/inha/d1" is a relative path since it doesnot start with "/", so the system looks for "home" (the first directory in home/inha/d1) starting at "/home/inha" (the current location). There is no "home" in "/home/inha", so the system gives an error message.

```
$ cd inha/d1
```

```
-bash: cd: inha/d1: No such file or directory
```

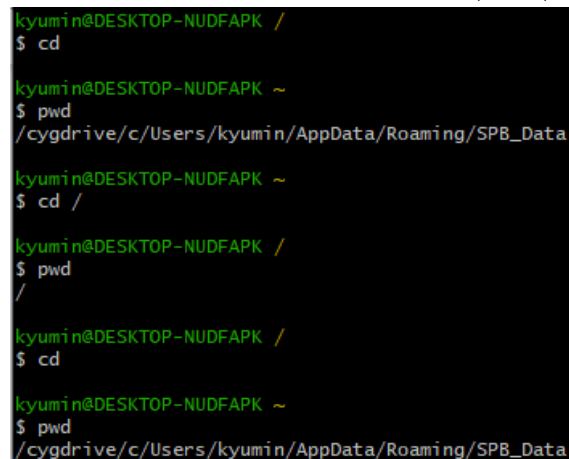
Again error because there is no "inha" in the current location(/home/inha).

```
$ cd d1
```

```
$ pwd
```

```
/home/inha/d1
```

"d1" is ok because we have "d1" in /home/inha".



```
kyumin@DESKTOP-NUDFAPK /
$ cd

kyumin@DESKTOP-NUDFAPK ~
$ pwd
/cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data

kyumin@DESKTOP-NUDFAPK ~
$ cd /

kyumin@DESKTOP-NUDFAPK /
$ pwd
/

kyumin@DESKTOP-NUDFAPK /
$ cd

kyumin@DESKTOP-NUDFAPK ~
$ pwd
/cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data
```

Cd / 명령어로 root 디렉토리로 이동했다. cd 명령어를 사용하면 첫 로그인할 때의 위치로 이동한다. 어느 위치에 있던 cd 명령어를 사용하면 로그인할 때의 위치로 이동한다는 것을 알 수 있다. (내 PC 는 서버로 접속하지 않고 바로 진행하여 로그인 위치는 강의노트와 다를 수 있지만, 처음 위치로 갈 수 있는 cd 명령어는 동일하게 작동한다)

```

kyumin@DESKTOP-NUDFAPK ~
$ pwd
/cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data

kyumin@DESKTOP-NUDFAPK ~
$ mkdir d1

kyumin@DESKTOP-NUDFAPK ~
$ cd d1

kyumin@DESKTOP-NUDFAPK ~/d1
$ pwd
/cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data/d1

kyumin@DESKTOP-NUDFAPK ~/d1
$ cd ..

kyumin@DESKTOP-NUDFAPK ~
$ pwd
/cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data

kyumin@DESKTOP-NUDFAPK ~
$ cd ..

kyumin@DESKTOP-NUDFAPK /cygdrive/c/Users/kyumin/AppData/Roaming
$ pwd
/cygdrive/c/Users/kyumin/AppData/Roaming

```

Mkdir 명령어를 이용하여 디렉토리를 만들고, cd d1 명령어로 d1 디렉토리 안에 들어갈 수 있음을 확인하였다. Cd ..을 두 번 사용하여 d1의 상위 디렉토리의 상위 디렉토리인 Roaming으로 이동했다. 그 결과는 pwd를 이용하여 확인하였다. (노란색 글씨로 현 위치가 떠야하나 ~ 표시인 이유는 상위 디렉토리가 많아 ~로 표기 된다.)

~은 /cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data 위치로 보면 된다.)

```

kyumin@DESKTOP-NUDFAPK /cygdrive/c/Users/kyumin/AppData/Roaming
$ cd /cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data/d1

kyumin@DESKTOP-NUDFAPK ~/d1
$ pwd
/cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data/d1

```

Cd 명령어 사용 시 절대경로를 입력해줬고, pwd를 사용하여 정상적으로 이동된 것을 알 수 있다.

```

kyumin@DESKTOP-NUDFAPK ~/d1
$ cd

kyumin@DESKTOP-NUDFAPK ~
$ pwd
/cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data

kyumin@DESKTOP-NUDFAPK ~
$ cd Roaming/SPB_Data/d1
-bash: cd: Roaming/SPB_Data/d1: No such file or directory

kyumin@DESKTOP-NUDFAPK ~
$ cd SPB_Data/d1
-bash: cd: SPB_Data/d1: No such file or directory

kyumin@DESKTOP-NUDFAPK ~
$ cd d1

kyumin@DESKTOP-NUDFAPK ~/d1
$ pwd
/cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data/d1

```

Cd 명령어 사용 시 상대경로를 입력하였다. 상위 디렉토리 또는 현재 위치의 디렉토리를 입력하였을 경우 오류가 발생한다. 하위 디렉토리를 입력하여 이동하니 정상적으로 이동되었다. 이로써 상대경로를 이용하여 이동 시 하위 디렉토리를 입력해야 한다는 것을 알 수 있다.

10) Go back to the login directory and Confirm your current location with "pwd".

```
$ cd  
$ pwd
```

```
kyumin@DESKTOP-NUDFAPK /cygdrive/c/Users/kyumin  
$ cd  
kyumin@DESKTOP-NUDFAPK ~  
$ pwd  
/cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data
```

Cd 는 로그인 위치로 이동하는 명령어고 pwd 를 통해 정상적으로 이동된 것을 확인할 수 있다.

11) List all files in your directory with "ls" command.

```
kyumin@DESKTOP-NUDFAPK ~  
$ ls  
cdssetup
```

Ls 명령어는 현재 디렉토리의 모든 파일을 볼 수 있다.

12) Try "echo" command.

```
$ echo korea  
korea  
$ echo hello  
Hello
```

```
kyumin@DESKTOP-NUDFAPK ~  
$ echo korea  
korea  
kyumin@DESKTOP-NUDFAPK ~  
$ echo hello  
hello
```

Echo 는 메시지를 출력할 수 있는 함수다.

11) Try "echo" with ">" symbol. ">" is called "standard output redirection".

```
$ echo hello there > f2
```

">" sends "hello there" to file "f2" so there is no output shown in the terminal but you will have a new file f2 in the current directory.

```
$ ls  
$ cat f2
```

```
kyumin@DESKTOP-NUDFAPK ~  
$ echo hello there > f2  
kyumin@DESKTOP-NUDFAPK ~  
$ ls  
cdssetup f2  
kyumin@DESKTOP-NUDFAPK ~  
$ cat f2  
hello there
```

Echo hello there > f2 는 hello there 내용을 가진 f2 파일을 생성한다. (만약 파일이 이미 존재한다면 파일 내용만 수정한다)

Ls 명령어를 이용하여 f2 파일이 생성된 것을 확인하였고, cat f2 를 이용하여 “hello there” 메시지가 출력 되었다. Echo 는 파일에 내용을 입력할 수 있음을 확인하였다.

12) Do "ls" to see you can find f1 in the current directory. Show its content with "cat".

```
$ ls
```

```
$ cat f1
```

```
kyumin@DESKTOP-NUDFAPK ~  
$ ls  
cdssetup  f2  
  
kyumin@DESKTOP-NUDFAPK ~  
$ cat f1  
cat: f1: No such file or directory  
  
kyumin@DESKTOP-NUDFAPK ~  
$ echo hi > f1  
  
kyumin@DESKTOP-NUDFAPK ~  
$ ls  
cdssetup  f1  f2  
  
kyumin@DESKTOP-NUDFAPK ~  
$ cat f1  
hi
```

F1 이라는 파일이 없는데 cat f1을 하면 오류가 발생한다. Echo 를 이용하여 f1 파일 생성해주고, 다시 cat f1을하면 메시지가 출력된다.

13) Make a directory, d2, with "mkdir".

```
$ mkdir d2
```

```
kyumin@DESKTOP-NUDFAPK ~  
$ mkdir d2  
  
kyumin@DESKTOP-NUDFAPK ~  
$ ls  
cdssetup  d2  f1  f2
```

Mkdir d2 명령어를 이용해서 d2 라는 디렉토리를 만들었다. Ls 로 d1 디렉토리가 생성된 것을 볼 수 있다.

14) Copy f1 into directory d2.

```
$ cp f1 d2 -- make a copy of f1 inside directory d2 under the same name.
```

```
$ cd d2
```

```
$ ls
```

```
$ cat f1
```

```
$ cd ..
```

```
$ cp f1 d1/f2 -- make a copy of f1 inside d1 under another name f2
```

```
$ cd d1
```

```
$ ls
```

```

kyumin@DESKTOP-NUDFAPK ~
$ cp f1 d2

kyumin@DESKTOP-NUDFAPK ~
$ cd d2

kyumin@DESKTOP-NUDFAPK ~/d2
$ ls
f1

kyumin@DESKTOP-NUDFAPK ~/d2
$ cat f1
hi

kyumin@DESKTOP-NUDFAPK ~/d2
$ cd ..

```

Cp f1 d2 는 f1 파일을 d2 디렉토리 안에 복사할 수 있다. 그래서 d2 안에 들어간 후 cat f1 으로 확인해보면 “hi” 라는 메시지가 출력된다.

```

$ cp f1 d1/f2 -- make a copy of f1 inside d1 under another name f2
$ cd d1
$ ls

```

```

kyumin@DESKTOP-NUDFAPK ~
$ cp f1 d1/f2
cp: cannot create regular file 'd1/f2': No such file or directory

kyumin@DESKTOP-NUDFAPK ~
$ mkdir d1

kyumin@DESKTOP-NUDFAPK ~
$ cp f1 d1/f2

kyumin@DESKTOP-NUDFAPK ~
$ cd d1

kyumin@DESKTOP-NUDFAPK ~/d1
$ ls
f2

kyumin@DESKTOP-NUDFAPK ~/d1
$ cat f2
hi

```

Cd f1 d1/f2는 d1 디렉토리 안에 f1의 파일을 f2라는 이름으로 복사하라는 명령어다. D1 디렉토리가 없다면 위와 같이 오류가 발생한다. D1 디렉토리 생성 후에는 정상적으로 f2 파일이 생성된다.

3.2 Do followings and explain what happens and why.

```
$ cd
$ ls
$ ls -l
$ ls -al
```

```
kyumin@DESKTOP-NUDFAPK ~
$ cd

kyumin@DESKTOP-NUDFAPK ~
$ ls
cdssetup d1 d2 f1 f2

kyumin@DESKTOP-NUDFAPK ~
$ ls -l
total 2
drwx-----+ 1 kyumin  0  0 Mar 15 2021 cdssetup
drwxr-xr-x+ 1 kyumin  0  0 Mar  6 12:51 d1
drwxr-xr-x+ 1 kyumin  0  0 Mar  6 12:47 d2
-rw-r--r--+ 1 kyumin  0  3 Mar  6 10:17 f1
-rw-r--r--+ 1 kyumin  0 12 Mar  6 10:13 f2

kyumin@DESKTOP-NUDFAPK ~
$ ls -al
total 40
drwxrwx---+ 1 Administrators  0  0 Mar  8 10:33 .
drwxrwx---+ 1 kyumin         SYSTEM  0 Mar  6 13:37 ..
-rw-----+ 1 kyumin         4437 Mar  8 10:33 .bash_history
-rwx-----+ 1 kyumin         51 Nov 16 2021 .gitconfig
drwx-----+ 1 kyumin         0 Jul 15 2021 .idlerc
-rw-----+ 1 kyumin         20 Mar  5 20:16 .lessht
-rw-----+ 1 kyumin        12288 Mar  1 17:49 .test.c.swp
-rw-----+ 1 kyumin        1207 Mar  3 15:32 .viminfo
drwx-----+ 1 kyumin         0 Mar 15 2021 cdssetup
drwxr-xr-x+ 1 kyumin         0 Mar  6 12:51 d1
drwxr-xr-x+ 1 kyumin         0 Mar  6 12:47 d2
-rw-r--r--+ 1 kyumin         0  3 Mar  6 10:17 f1
-rw-r--r--+ 1 kyumin         0 12 Mar  6 10:13 f2
```

Cd 를 사용하여 로그인 위치로 간 후 ls 를 이용하여 디렉토리 및 파일을 확인했다. Ls -l 을 사용하여 현재 위치에는 3 개의 디렉토리가 있음을 알 수 있다. Ls -l 은 파일들의 권한과 오너 ID, 그룹원 ID와 용량, 파일 생성 날짜 또는 최종수정시간, 파일 이름을 볼 수 있다. Ls -al 에서 -a 는 숨겨진 파일을 포함하라는 옵션이고, -l 은 파일의 정보를 자세하게 출력하는 옵션이다.

```
$ cd /
$ file *
```

```
kyumin@DESKTOP-NUDFAPK ~
$ cd /

kyumin@DESKTOP-NUDFAPK /
$ file *
Cygwin-Terminal.ico: MS Windows icon resource - 6 icons, 16x16, 32 bits/pixel, 24x24, 32 bits/pixel
Cygwin.bat:          DOS batch file, ASCII text, with CRLF line terminators
Cygwin.ico:          MS Windows icon resource - 7 icons, -104x-104, 32 bits/pixel, 16x16, 32 bits/pixel
bin:                 directory
cygdrive:            directory
dev:                 directory
etc:                 directory
home:                sticky, directory
lib:                 directory
proc:                directory
sbin:                directory
tmp:                 sticky, directory
usr:                 directory
var:                 directory
```

Cd / 를 사용하여 Root 로 이동했다. file *을 사용하여 현재위치에 있는 모든 파일들의 타입을 확인할 수 있다. 위 사진의 경우 3 개의 파일을 제외한 나머지는 모두 디렉토리임을 알 수 있다.

```
$ find . -name stdio.h
```

```
kyumin@DESKTOP-NUDFAPK /  
$ find . -name stdio.h  
./lib/gcc/x86_64-pc-cygwin/11/include/c++/tr1/stdio.h  
./usr/include/ssp/stdio.h  
./usr/include/stdio.h  
./usr/include/sys/stdio.h  
./usr/lib/gcc/x86_64-pc-cygwin/11/include/c++/tr1/stdio.h
```

find . -name stdio.h 란 현재 디렉토리와 하위 디렉토리에서 파일명이 stdio.h 인 파일의 위치를 찾는 명령어다. (현재 위치가 root 디렉토리여서 검색 시간이 상당히 오래 걸리므로 ctrl + c 로 검색을 중단했다.)

```
$ cd bin
```

```
$ ls
```

```
$ ls bz*
```

```
kyumin@DESKTOP-NUDFAPK /  
$ cd bin  
  
kyumin@DESKTOP-NUDFAPK /bin  
$ ls  
'[.exe' lexgrog.exe  
addftinfo.exe libpython3.9.dll  
addr2line.exe link.exe  
alias lkbib.exe  
apropos ln.exe  
ar.exe locale.exe  
arch.exe locate.exe  
as.exe logger.exe  
ash.exe login.exe  
awk logname.exe  
b2sum.exe look.exe  
base32.exe lookbib.exe  
base64.exe ls.exe  
basename.exe lsattr.exe  
basenc.exe lto-dump.exe  
basb.exe... lzcat  
ld.exe zstd.exe  
ldd.exe zstdcat  
ldh.exe zstdgrep  
less.exe zstdless  
lessecho.exe zstdmt  
lesskey.exe  
  
kyumin@DESKTOP-NUDFAPK /bin  
$ ls bz*  
bzcat.exe bzdiff bzfgrep bzip2.exe bzless  
bzcmp bzegrep bzgrep bzip2recover.exe bzmore
```

Cd bin, ls 를 이용하여 Bin 디렉토리 확인해보니 파일이 매우 많이 있다. Ls bz*을 사용하면 bz 로 시작하는 파일만 볼 수 있다.

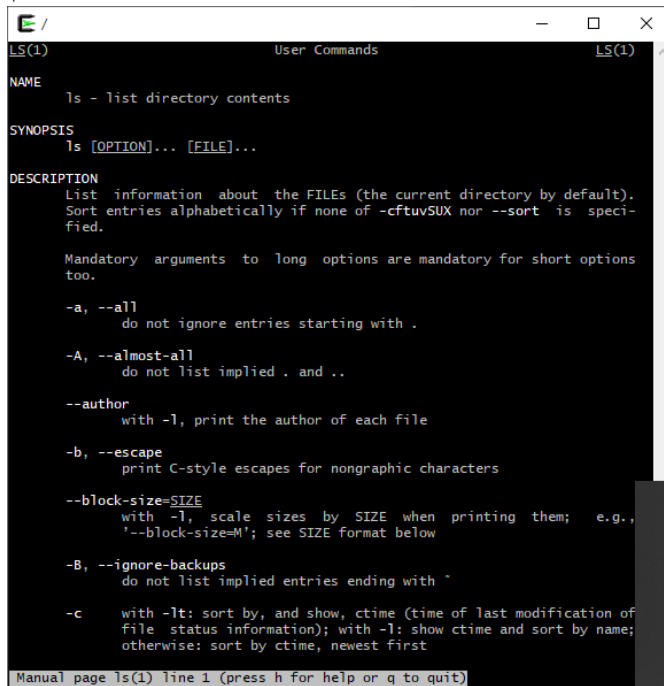
```
$ cd
```

```
$ pwd
```

```
kyumin@DESKTOP-NUDFAPK ~  
$ cd  
  
kyumin@DESKTOP-NUDFAPK ~  
$ pwd  
/cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data
```

Cd 명령어는 로그인할 때의 위치로 이동한다. 이동 후 pwd 를 이용하여 현 위치 확인이 가능하다.

```
$ man ls
```



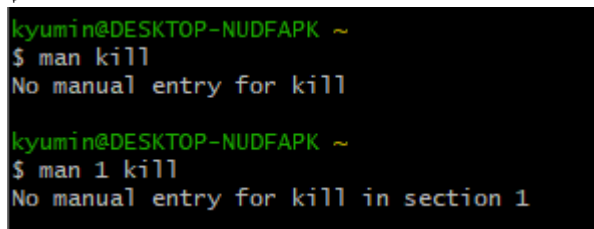
man ls 명령어를 사용하여 ls 명령어에 대한 설명을 볼 수 있다. Enter 로 한 줄씩 또는 SpaceBar 로 한 페이지씩 이동하면서 설명을 볼 수 있고, q 를 눌러 나갈 수 있다.

```
$ man kill
```

```
$ man 1 kill
```

```
$ man 2 kill
```

```
$ man 3 kill
```

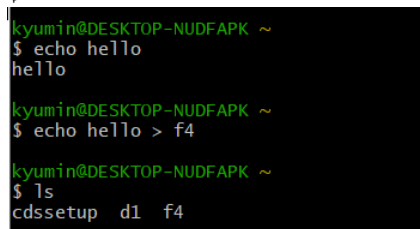


윈도우 cygwin 에서 진행하였기 때문에 kill 명령어를 검색하면 오류 메시지가 발생한다. 다른 명령어(ex. Ls, Cd, cp, chmod 등)들은 문제없다.

```
$ echo hello
```

```
$ echo hello > f4
```

```
$ ls
```



Echo hello 명령어는 단순히 hello 라는 메시지만 출력한다. Echo hello > f4 는 hello 라는 내용을 가진 f4 파일을 생성한다. (만약 f4 라는 파일이 이미 존재한다면, f4 의 내용을 hello 로 수정한다)

```
$ cp f4 f2
$ cat f4
$ cat f2
```

```
kyumin@DESKTOP-NUDFAPK ~
$ cp f4 f2

kyumin@DESKTOP-NUDFAPK ~
$ cat f4
hello

kyumin@DESKTOP-NUDFAPK ~
$ cat f2
hello

kyumin@DESKTOP-NUDFAPK ~
$ ls
cdssetup d1 f2 f4
```

Cp f4 f2 는 f4 파일과 동일한 파일이 복사되며 복사된 파일 이름은 f2 다. (f2 파일이 이미 존재한다면 내용을 hello 로 수정한다)
Cat f4 f2 를 통해 동일한 결과가 출력되는 것을 볼 수 있다.

```
$ cat f2 > f3
$ ls -l f*
$ rm f2
$ ls
$ cat f4
$ xxd f4
```

```
kyumin@DESKTOP-NUDFAPK ~
$ cat f2 > f3

kyumin@DESKTOP-NUDFAPK ~
$ ls -l f*
-rw-r--r--+ 1 kyumin 없음 6 Mar 6 09:05 f2
-rw-r--r--+ 1 kyumin 없음 6 Mar 6 09:15 f3
-rw-r--r--+ 1 kyumin 없음 6 Mar 6 09:05 f4

kyumin@DESKTOP-NUDFAPK ~
$ rm f2

kyumin@DESKTOP-NUDFAPK ~
$ ls
cdssetup d1 f3 f4

kyumin@DESKTOP-NUDFAPK ~
$ cat f4
hello

kyumin@DESKTOP-NUDFAPK ~
$ xxd f4
00000000: 6865 6c6c 6f0a                hello.
```

Cat f2 > f3 은 f2 파일을 복사하는 것이며 복사된 파일이름은 f3 이다.(f3 파일이 이미 존재한다면 내용만 수정된다)

Ls -l f* 은 f 로 시작하는 파일들을 보여달라는 명령어다. -l 을 추가하면 권한/오너와 그룹원의 ID/용량/최종수정시간/파일이름을 확인할 수 있다.

Rm f2 를 이용하여 f2 파일을 제거했고 ls 를 통해 확인해보면 f2 는 삭제된 것을 알 수 있다.

Xxd f4 는 f4 파일의 내용을 16 진수로 표기한다. 아스키코드와 비교해보면 결과로 나온 16 진수는 hello 임을 알 수 있다.

```
$ mkdir d2
$ cp f4 d2
$ cd d2
$ pwd
$ ls
```

```
kyumin@DESKTOP-NUDFAPK ~
$ mkdir d2

kyumin@DESKTOP-NUDFAPK ~
$ cp f4 d2

kyumin@DESKTOP-NUDFAPK ~
$ cd d2

kyumin@DESKTOP-NUDFAPK ~/d2
$ pwd
/cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data/d2

kyumin@DESKTOP-NUDFAPK ~/d2
$ ls
f4
```

Mkdir d2 는 d2 라는 이름의 디렉토리를 만든다. Cd f4 d2 는 복사하는 명령어이기는 하지만 d2 가 파일이 아닌 디렉토리다. 이런 경우는 d2 라는 디렉토리 안에 f4 파일이 복사된다.

Cd d2 로 d2 디렉토리 안에 들어가서, pwd 로 현위치가 d2 디렉토리임을 확인했고, ls 를 이용하여 d2 디렉토리에는 f4 라는 파일이 존재하는 것을 확인할 수 있다.

```
$ cd ..
$ grep -nr "he" *
```

```
kyumin@DESKTOP-NUDFAPK ~/d2
$ cd ..

kyumin@DESKTOP-NUDFAPK ~
$ grep -nr "he" *
cdssetup/OrCAD_Capture/17.2.0/Capture.ini:74:Enable Button Checked=1
cdssetup/OrCAD_Capture/17.2.0/Capture.ini:167:CheckConnection..=True
cdssetup/OrCAD_Capture/17.2.0/Capture.ini:172:CheckReadOnlyOnViewActivate.....=True
cdssetup/OrCAD_Capture/17.2.0/Capture.ini:237:DisplayCISPartPropWrtAnotherProp=
cdssetup/OrCAD_Capture/17.2.0/Capture.ini:301:Indicates whether to show the border=False
cdssetup/OrCAD_Capture/17.2.0/Capture.ini:324:Indicates whether to show the border=False
cdssetup/OrCAD_Capture/17.2.0/Capture.ini:367:[Property Spreadsheet Size]
cdssetup/OrCAD_Capture/17.2.0/Capture.ini:379:[Spreadsheet]
cdssetup/OrCAD_Capture/17.2.0/Capture.ini:560:[Configured Schematic Macros]
cdssetup/OrCAD_Capture/17.2.0/capture_69579000.ini:22:CheckConnection..=True
cdssetup/OrCAD_Capture/17.2.0/capture_69579000.ini:29:CheckReadOnlyOnViewActivate.....=True
cdssetup/OrCAD_Capture/17.2.0/capture_69579000.ini:80:[Property Spreadsheet Size]
cdssetup/OrCAD_Capture/17.2.0/capture_69579000.ini:85:[Spreadsheet]
cdssetup/OrCAD_Capture/17.2.0/capture_69579000.ini:141:[Configured Schematic Macros]
cdssetup/OrCAD_Capture/17.2.0/capture_69579000.ini:291:Indicates whether to show the border=False
cdssetup/OrCAD_Capture/17.2.0/capture_69579000.ini:314:Indicates whether to show the border=False
cdssetup/OrCAD_Capture/17.2.0/preprop.txt:1246: ("Schematic Create Time" show)
cdssetup/OrCAD_Capture/17.2.0/preprop.txt:1247: ("Schematic Path" show)
cdssetup/OrCAD_Capture/17.2.0/preprop.txt:1275: ("Schematic Nets" "" 1 1)
cdssetup/OrCAD_PSpice/17.2.0/PSpice.ini:127:height=331
cdssetup/OrCAD_PSpice/17.2.0/PSpice.ini:195:[Schematics MRP LIST]
cdssetup/OrCAD_PSpice/17.2.0/PSpice.ini:320:height=349
d2/f4:1:hello
f3:1:hello
f4:1:hello

kyumin@DESKTOP-NUDFAPK ~
$ |
```

Grep -nr "he" * 실행하니 많은 결과가 나왔다. 이 명령어는 "he" 라는 내용이 들어간 모든 파일을 찾는 것이다. * 은 현재 디렉토리에서 찾으라는 것이다. -n은 내용이 들어간 행번호를 출력하라는 것이고, -r은 하위 모든 디렉토리에서 찾으라는 것이다.

그래서 Grep -nr "he" * 명령어는 현디렉토리와 하위 디렉토리에서 "he"라는 내용이 들어간 파일을 찾고, 그 파일의 위치와 he 가 들어간 행번호를 출력하는 것이다.

```
$ ps
$ ps -ef
$ ps -ef | more
```

```
kyumin@DESKTOP-NUDFAPK ~
$ ps
  PID   PPID   PGID   WINPID   TTY        UID     STIME  COMMAND
  1007    964    1007    13360   pty0       197609  09:52:58 /usr/bin/ps
    964    963    964    14596   pty0       197609  09:03:23 /usr/bin/bash
    963     1    963     8992    ?          197609  09:03:23 /usr/bin/mintty

kyumin@DESKTOP-NUDFAPK ~
$ ps -ef
  UID     PID   PPID   TTY        STIME  COMMAND
  kyumin   1008    964   pty0       09:53:01 /usr/bin/ps
  kyumin   964    963   pty0       09:03:23 /usr/bin/bash
  kyumin   963     1    ?          09:03:23 /usr/bin/mintty

kyumin@DESKTOP-NUDFAPK ~
$ ps -ef | more
  UID     PID   PPID   TTY        STIME  COMMAND
  kyumin   1010    964   pty0       09:53:05 /usr/bin/more
  kyumin   964    963   pty0       09:03:23 /usr/bin/bash
  kyumin   1009    964   pty0       09:53:05 /usr/bin/ps
  kyumin   963     1    ?          09:03:23 /usr/bin/mintty
```

Ps는 사용자의 프로세스를 보여주는 명령어다.

Ps -ef는 모든 사용자의 프로세스를 보여주는 명령어다.

Ps -ef | more는 모든 사용자의 프로세스는 풀포맷으로 보여주는 명령어라고 한다. 그러나 윈도우 cygwin에서 진행해서 그런지 ps -ef 와 동일한 형태로 결과가 나온다.

3.3 Run following commands and explain what happens.

chmod, clear, gzip, date, dd, df, diff, dmesg, du, env, exit, file, find, head, ifconfig, ln, mount, netstat, objdump, ping, stat, tail, time, touch, tty, gunzip, whereis, which, whoami,

```
kyumin@DESKTOP-NUDFAPK ~
$ chmod 000 f1

kyumin@DESKTOP-NUDFAPK ~
$ cat f1
cat: f1: Permission denied

kyumin@DESKTOP-NUDFAPK ~
$ ls -l
total 2
drwx-----+ 1 kyumin 없음 0 Mar 15 2021 cdssetup
drwxr-xr-x+ 1 kyumin 없음 0 Mar 6 12:51 d1
drwxr-xr-x+ 1 kyumin 없음 0 Mar 6 12:47 d2
-----+ 1 kyumin 없음 3 Mar 6 10:17 f1
-rw-r--r--+ 1 kyumin 없음 12 Mar 6 10:13 f2

kyumin@DESKTOP-NUDFAPK ~
$ chmod 777 f1

kyumin@DESKTOP-NUDFAPK ~
$ ls -l
total 2
drwx-----+ 1 kyumin 없음 0 Mar 15 2021 cdssetup
drwxr-xr-x+ 1 kyumin 없음 0 Mar 6 12:51 d1
drwxr-xr-x+ 1 kyumin 없음 0 Mar 6 12:47 d2
-rwxrwxrwx+ 1 kyumin 없음 3 Mar 6 10:17 f1
-rw-r--r--+ 1 kyumin 없음 12 Mar 6 10:13 f2
```

Chmod는 파일의 권한을 설정할 수 있는 명령어다. Chmod 000 f1 명령어를 사용하면 f1 파일의 모든 권한이 없어진다. Ls -l을 이용하여 f1파일의 권한이 모두 사라진 걸 볼 수 있고, cat f1 명령어를 사용해도 읽기 권한이 없기 때문에 오류가 발생하는 걸 볼 수 있다. Chmod 777으로 모든 권한을 주었다.

```
E ~
kyumin@DESKTOP-NUDFAPK ~
$
```

Clear 명령어는 터미널(명령어 입력 창)의 모든 내용을 제거하는 것이다.

```
kyumin@DESKTOP-NUDFAPK ~
$ gzip f1

kyumin@DESKTOP-NUDFAPK ~
$ ls -l
total 2
drwx-----+ 1 kyumin 없음 0 Mar 15 2021 cdssetup
drwxr-xr-x+ 1 kyumin 없음 0 Mar 6 12:51 d1
drwxr-xr-x+ 1 kyumin 없음 0 Mar 6 12:47 d2
-rwxrwxrwx+ 1 kyumin 없음 26 Mar 6 10:17 f1.gz
-rw-r--r--+ 1 kyumin 없음 12 Mar 6 10:13 f2
```

Gzip f1은 f1 파일을 압축하는 명령어다. 압축을 통해 디스크 사용량을 줄일 수 있다.

```
kyumin@DESKTOP-NUDFAPK ~  
$ date  
Sat Mar 9 21:54:47 KST 2024
```

Date는 현재 시간을 출력하는 명령어다.

```
kyumin@DESKTOP-NUDFAPK ~  
$ ls  
cdssetup d1 d2 f1.gz f2  
  
kyumin@DESKTOP-NUDFAPK ~  
$ dd if=f2 of=f3  
0+1 records in  
0+1 records out  
12 bytes copied, 0.0003773 s, 31.8 kB/s  
  
kyumin@DESKTOP-NUDFAPK ~  
$ ls  
cdssetup d1 d2 f1.gz f2 f3
```

Dd는 데이터를 복사하거나 변환하는 명령어다. 위 경우는 f2라는 파일을 f3라는 이름으로 복사하였다. 해당 명령어의 옵션으로는 입력 단위 바이트 지정, 파일 변환 등 다양한 옵션이 있다.

```
kyumin@DESKTOP-NUDFAPK ~  
$ df  
  
Filesystem      1K-blocks      Used Available Use% Mounted on  
C:/cygwin64    248665960 193005424  55660536   78% /
```

Df는 시스템 전체의 디스크 사용량을 볼 수 있는 명령어다.

```
kyumin@DESKTOP-NUDFAPK ~/d1  
$ diff f2 f3  
1c1  
< hi  
---  
> hello
```

Diff는 파일의 내용을 비교하는 명령어다. 위 사진의 경우 f2와 f3의 내용이 다르다는 결과를 볼 수 있다.

```
kyumin@DESKTOP-NUDFAPK ~/d1  
$ dmesg  
-bash: dmesg: command not found
```

Dmesg는 시스템이 부팅될 때 출력되는 메시지의 기록을 확인하는 명령어다. 해당 명령어는 cygwin에서 사용했기 때문에 오류가 발생한다.

```

kyumin@DESKTOP-NUDFAPK ~
$ du -h
4.0K    ./idlerc
101K    ./cdssetup/OrCAD_Capture/17.2.0
101K    ./cdssetup/OrCAD_Capture
12K     ./cdssetup/OrCAD_PSpice/17.2.0
0       ./cdssetup/OrCAD_PSpice/backup
12K     ./cdssetup/OrCAD_PSpice
113K    ./cdssetup
2.0K    ./d1
1.0K    ./d2
153K    .

```

Du는 특정 디렉토리의 디스크 사용량을 확인할 수 있는 명령어다. 위 사진의 경우 하위 디렉토리들의 사용량이 표기된 것을 볼 수 있고, -h 옵션을 사용하여 단위(k)도 확인이 된다. 명령어 뒤에 디렉토리 위치를 추가하면 그 디렉토리만을 확인할 수 있다.

```

kyumin@DESKTOP-NUDFAPK ~
$ env
ProgramFiles(x86)=C:\Program Files (x86)
!:::~\
CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
SHELL=/bin/bash
NUMBER_OF_PROCESSORS=8
FPS_BROWSER_USER_PROFILE_STRING=Default
PROCESSOR_LEVEL=6
TERM_PROGRAM_VERSION=3.7.1
USERDOMAIN_ROAMINGPROFILE=DESKTOP-NUDFAPK
HOSTNAME=DESKTOP-NUDFAPK
PROGRAMFILES=C:\Program Files
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
OS=Windows_NT
HOMEDRIVE=C:
USERDOMAIN=DESKTOP-NUDFAPK
PWD=/cygdrive/c/Users/kyumin/AppData/Roaming/SPB_Data
USERPROFILE=C:\Users\kyumin
PRINTER=OneNote (Desktop)
TZ=Asia/Seoul
ALLUSERSPROFILE=C:\ProgramData
ORIGINAL_PATH=/cygdrive/c/WINDOWS/system32:/cygdrive/c/WINDOWS:/cygdrive/c/WINDOWS/System32/Wbem:/cygdrive/c/WINDOWS/System32/WindowsPowerShell/v1.0:/cygdrive/c/WINDOWS/System32/OpenSSH:/cygdrive/c/Program Files/Git/cmd:/cygdrive/c/Program Files/dotnet:/cygdrive/c/Users/kyumin/AppData/Local/Programs/Python/Python37/Scripts:/cygdrive/c/Users/kyumin/AppData/Local/Programs/Python/Python37:/cygdrive/c

```

env는 현재 지정되어 있는 환경 변수들을 출력하거나, 새로운 환경 변수를 설정 및 적용된 내용을 출력할 수 있는 명령어다.

```

kyumin@DESKTOP-NUDFAPK ~
$ exit

```

터미널을 종료하는 명령어다.

```
kyumin@DESKTOP-NUDFAPK ~
$ ls
cdssetup d1 d2 f1.gz f2 f3

kyumin@DESKTOP-NUDFAPK ~
$ file d1 f2
d1: directory
f2: ASCII text
```

File은 파일의 형식을 보여주는 명령어다.

```
kyumin@DESKTOP-NUDFAPK ~
$ find d1
d1
d1/f2
d1/f3

kyumin@DESKTOP-NUDFAPK ~
$ find . -name "f*"
./d1/f2
./d1/f3
./d2/f1
./f1.gz
./f2
./f3
```

Find는 파일을 검색할 수 있는 명령어다. find . -name f*는 현재 디렉토리와 하위 디렉토리에서 f로 시작하는 파일의 위치를 찾아주는 명령어다.

```
kyumin@DESKTOP-NUDFAPK ~
$ head f2
hello there
```

Head f2는 f2 파일의 내용 앞부분을 10행까지 출력하는 명령어다. -n 옵션을 사용하면 원하는 행만큼 출력할 수 있다. (Head -n 20 f2라는 명령어를 사용하면 f2의 앞부분을 20행까지 출력할 수 있다)

```
kyumin@DESKTOP-NUDFAPK ~
$ ifconfig
-bash: ifconfig: command not found
```

Ifconfig는 네트워크 인터페이스를 설정하거나 확인하는 명령어다. 그러나 cygwin에서 진행했기 때문에 오류가 발생했다.

```
kyumin@DESKTOP-NUDFAPK ~
$ ls -l
total 2
drwx-----+ 1 kyumin 없음 0 Mar 15 2021 cdssetup
drwxr-xr-x+ 1 kyumin 없음 0 Mar 9 22:13 d1
drwxr-xr-x+ 1 kyumin 없음 0 Mar 6 12:47 d2
-rwxrwxrwx+ 1 kyumin 없음 26 Mar 6 10:17 f1.gz
-rw-r--r--+ 1 kyumin 없음 12 Mar 9 23:51 f2
```

```
kyumin@DESKTOP-NUDFAPK ~
$ ln f2 HL
```

```
kyumin@DESKTOP-NUDFAPK ~
$ cat HL
hello there
```

```
kyumin@DESKTOP-NUDFAPK ~
$ rm f2
```

```
kyumin@DESKTOP-NUDFAPK ~
$ cat HL
hello there
```

```
kyumin@DESKTOP-NUDFAPK ~
$ ls -l
total 2
-rw-r--r--+ 1 kyumin 없음 12 Mar 9 23:51 HL
drwx-----+ 1 kyumin 없음 0 Mar 15 2021 cdssetup
drwxr-xr-x+ 1 kyumin 없음 0 Mar 9 22:13 d1
drwxr-xr-x+ 1 kyumin 없음 0 Mar 6 12:47 d2
-rwxrwxrwx+ 1 kyumin 없음 26 Mar 6 10:17 f1.gz
```

```
kyumin@DESKTOP-NUDFAPK ~
$ ls
cdssetup d1 d2 f1.gz f2
```

```
kyumin@DESKTOP-NUDFAPK ~
$ ln -s f2 SL
```

```
kyumin@DESKTOP-NUDFAPK ~
$ rm f2
```

```
kyumin@DESKTOP-NUDFAPK ~
$ cat SL
cat: SL: No such file or directory
```

```
kyumin@DESKTOP-NUDFAPK ~
$ ls -l
total 1
lrwxrwxrwx 1 kyumin 없음 2 Mar 9 23:54 SL -> f2
drwx-----+ 1 kyumin 없음 0 Mar 15 2021 cdssetup
drwxr-xr-x+ 1 kyumin 없음 0 Mar 9 22:13 d1
drwxr-xr-x+ 1 kyumin 없음 0 Mar 6 12:47 d2
-rwxrwxrwx+ 1 kyumin 없음 26 Mar 6 10:17 f1.gz
```

In 명령어는 하드 링크 또는 심볼릭 링크를 만드는 명령어다. 첫번째 사진의 경우 하드 링크를 생성했다. 동일한 파일이 생성되었고, 원본 파일을 제거해도 cat HL을 하면 f2와 동일한 내용이 출력됨을 알 수 있다. 두번째 사진의 경우 심볼릭 링크를 생성했다. 원본 파일을 제거 후 cat SL를 하면 오류가 발생한다.

```
kyumin@DESKTOP-NUDFAPK /home
$ mount
C:/cygwin64/bin on /usr/bin type ntfs (binary,auto)
C:/cygwin64/lib on /usr/lib type ntfs (binary,auto)
C:/cygwin64 on / type ntfs (binary,auto)
C: on /cygdrive/c type ntfs (binary,posix=0,user,noumount,auto)
```

Mount는 다른 파일 시스템을 현재 디렉토리에 임시로 연결하는 작업이며, 마운트를 통해 다른 파일 시스템에 접근할 수 있다.

```
kyumin@DESKTOP-NUDFAPK /home
$ netstat -a
```

활 성 연 결

프 로 토 콜	로컬 주소	외부 주소	상 태
TCP	0.0.0.0:135	DESKTOP-NUDFAPK:0	LISTENING
TCP	0.0.0.0:445	DESKTOP-NUDFAPK:0	LISTENING
TCP	0.0.0.0:5040	DESKTOP-NUDFAPK:0	LISTENING
TCP	0.0.0.0:5357	DESKTOP-NUDFAPK:0	LISTENING
TCP	0.0.0.0:7680	DESKTOP-NUDFAPK:0	LISTENING
TCP	0.0.0.0:49664	DESKTOP-NUDFAPK:0	LISTENING
TCP	0.0.0.0:49665	DESKTOP-NUDFAPK:0	LISTENING
TCP	0.0.0.0:49666	DESKTOP-NUDFAPK:0	LISTENING
TCP	0.0.0.0:49667	DESKTOP-NUDFAPK:0	LISTENING
TCP	0.0.0.0:49668	DESKTOP-NUDFAPK:0	LISTENING
TCP	0.0.0.0:49671	DESKTOP-NUDFAPK:0	LISTENING

Netstat은 네트워크 연결상태, 라우팅 테이블, 인터페이스 상태 등을 보여주는 명령어다.

```

kyumin@DESKTOP-NUDFAPK ~
$ objdump
Usage: objdump <option(s)> <file(s)>
Display information from object <file(s)>.
At least one of the following switches must be given:
-a, --archive-headers    Display archive header information
-f, --file-headers       Display the contents of the overall file header
-p, --private-headers    Display object format specific file header contents
-P, --private=OPT,OPT... Display object format specific contents
-h, --[section-]headers  Display the contents of the section headers
-x, --all-headers        Display the contents of all headers
-d, --disassemble        Display assembler contents of executable sections
-D, --disassemble-all   Display assembler contents of all sections
    --disassemble=<sym>  Display assembler contents from <sym>
-S, --source             Intermix source code with disassembly
    --source-comment[=<txt>] Prefix lines of source code with <txt>
-s, --full-contents      Display the full contents of all sections requested
-Z, --decompress         Decompress section(s) before displaying their content
s
-g, --debugging          Display debug information in object file
-e, --debugging-tags     Display debug information using ctags style
-G, --stabs              Display (in raw form) any STABS info in the file
-W, --dwarf[a/=abbrev, A/=addr, r/=aranges, c/=cu_index, L/=decodedline,
    f/=frames, F/=frames-interp, g/=gdb_index, i/=info, o/=loc,
    m/=macro, p/=pubnames, t/=pubtypes, R/=Ranges, l/=rawline,
    s/=str, O/=str-offsets, u/=trace_abbrev, T/=trace_aranges,
    U/=trace_info]       Display the contents of DWARF debug sections
    --dwarf-links        Display the contents of sections that link to

```

Objdump는 라이브러리, 컴파일된 오브젝트 모듈, 공유 오브젝트 파일, 독립 실행파일 등의 바이너리 파일들의 정보를 보여주는 명령어다.

```

kyumin@DESKTOP-NUDFAPK ~
$ ping 8.8.8.8

Ping 8.8.8.8 32바이트 데이터 사용 :
8.8.8.8의 응답 : 바이트=32 시간=66ms TTL=58
8.8.8.8의 응답 : 바이트=32 시간=43ms TTL=58
8.8.8.8의 응답 : 바이트=32 시간=41ms TTL=58
8.8.8.8의 응답 : 바이트=32 시간=39ms TTL=58

8.8.8.8에 대한 Ping 통계 :
   패킷 : 보낸 = 4, 받은 = 4, 손실 = 0 (0% 손실),
   왕복 시간 (밀리초):
   최소 = 39ms, 최대 = 66ms, 평균 = 47ms

```

ping 명령어는 네트워크 상태를 확인할 수 있는 명령어다. 위 사진의 경우 구글 DNS 서버인 8.8.8.8에 패킷을 보내고 응답 시간을 확인할 수 있다.

```

kyumin@DESKTOP-NUDFAPK ~
$ ls
SL cdssetup d1 d2 f1.gz

kyumin@DESKTOP-NUDFAPK ~
$ stat SL
  File: SL -> f2
  Size: 2          Blocks: 0          IO Block: 65536  symbolic link
Device: 23089,16961 Inode: 8444249301487349 Links: 1
Access: (0777/lrwxrwxrwx)  Uid: (197609/  kyumin)   Gid: (197121/  없음 )
Access: 2024-03-09 23:54:07.701923100 +0900
Modify: 2024-03-09 23:54:07.701923100 +0900
Change: 2024-03-09 23:54:07.702915200 +0900
 Birth: 2024-03-09 23:54:07.701923100 +0900

```

Stat은 파일의 상세정보를 보여주는 명령어다.

```

kyumin@DESKTOP-NUDFAPK ~/d1
$ tail f2
hi

```

Tail은 파일의 마지막 10행을 보여주는 명령어다. Head와 비슷하지만 마지막 내용을 출력한다는 차이점이 있다. -n 옵션을 사용한다면 원하는 행만큼 출력할 수 있다.

```

kyumin@DESKTOP-NUDFAPK ~/d1
$ time cat f2
hi

real    0m0.080s
user    0m0.000s
sys     0m0.031s

```

Time은 프로그램의 실행시간 보여주는 명령어다. Real 은 전체 실행시간을 의미한다. User는 Kernel 밖에서 실행되는 시간이고, sys는 Kernel 내부에서 system call에 사용된 CPU time만을 의미한다.

```

kyumin@DESKTOP-NUDFAPK ~/d1
$ ls -l
total 2
-rw-r--r--+ 1 kyumin 없음 3 Mar  6 12:51 f2
-rw-r--r--+ 1 kyumin 없음 6 Mar  9 22:13 f3

kyumin@DESKTOP-NUDFAPK ~/d1
$ touch -t 202401011230 f3

kyumin@DESKTOP-NUDFAPK ~/d1
$ ls -l
total 2
-rw-r--r--+ 1 kyumin 없음 3 Mar  6 12:51 f2
-rw-r--r--+ 1 kyumin 없음 6 Jan  1 12:30 f3

```

Touch는 파일 생성하거나 파일 수정 날짜 및 시간을 조정할 수 있는 명령어다. -t은 수정 시간을 임의로 조정할 수 있는 옵션이고, 위 사진은 f3 파일의 수정 시간을 다른 시간으로 조정한 것이다.

```
kyumin@DESKTOP-NUDFAPK ~/d1
$ tty
/dev/pty0

kyumin@DESKTOP-NUDFAPK ~/d1
$ ...

kyumin@DESKTOP-NUDFAPK ~
$ tty
/dev/pty1

kyumin@DESKTOP-NUDFAPK ~
$ tty
/dev/pty2

kyumin@DESKTOP-NUDFAPK ~
$
```

현재 커널과 연결된 가상 터미널 장치 이름을 확인할 수 있다. cygwin에서도 여러 개의 콘솔을 띄운 후 tty를 사용하면 몇 번 콘솔인지 확인할 수 있다.

```
kyumin@DESKTOP-NUDFAPK ~
$ ls
SL cdssetup d1 d2 f1.gz

kyumin@DESKTOP-NUDFAPK ~
$ gunzip f1.gz

kyumin@DESKTOP-NUDFAPK ~
$ ls
SL cdssetup d1 d2 f1

kyumin@DESKTOP-NUDFAPK ~
$ cat f1
hi
```

Gunzip은 압축을 풀기위한 명령어다. F1.gz 파일을 압축 해제했고 cat f1을 통해 정상적으로 읽기가 되는 것을 알 수 있다.

```
kyumin@DESKTOP-NUDFAPK ~
$ whereis ls
ls: /usr/bin/ls.exe /usr/share/man/man1/ls.1.gz

kyumin@DESKTOP-NUDFAPK ~
$ whereis kill
kill: /usr/bin/kill.exe

kyumin@DESKTOP-NUDFAPK ~
$ whereis ifconfig
ifconfig:
```

whereis 명령어는 프로그램 실행 파일 소스 코드 및 man 페이지의 위치를 보여준다.


```
kyumin@DESKTOP-NUDFAPK /usr/bin  
$ which ls  
/usr/bin/ls
```

which는 명령어의 실행파일 경로를 찾아주는 명령어이다. 차이점으로 whereis 명령어는 입력한 명령어의 실행 파일뿐만 아니라 매뉴얼 페이지와 소스 코드 파일의 위치도 찾아준다.

```
kyumin@DESKTOP-NUDFAPK /usr/bin  
$ whoami  
kyumin
```

Whoami 현재 내가 로그인한 사용자 ID를 보여주는 명령어다.