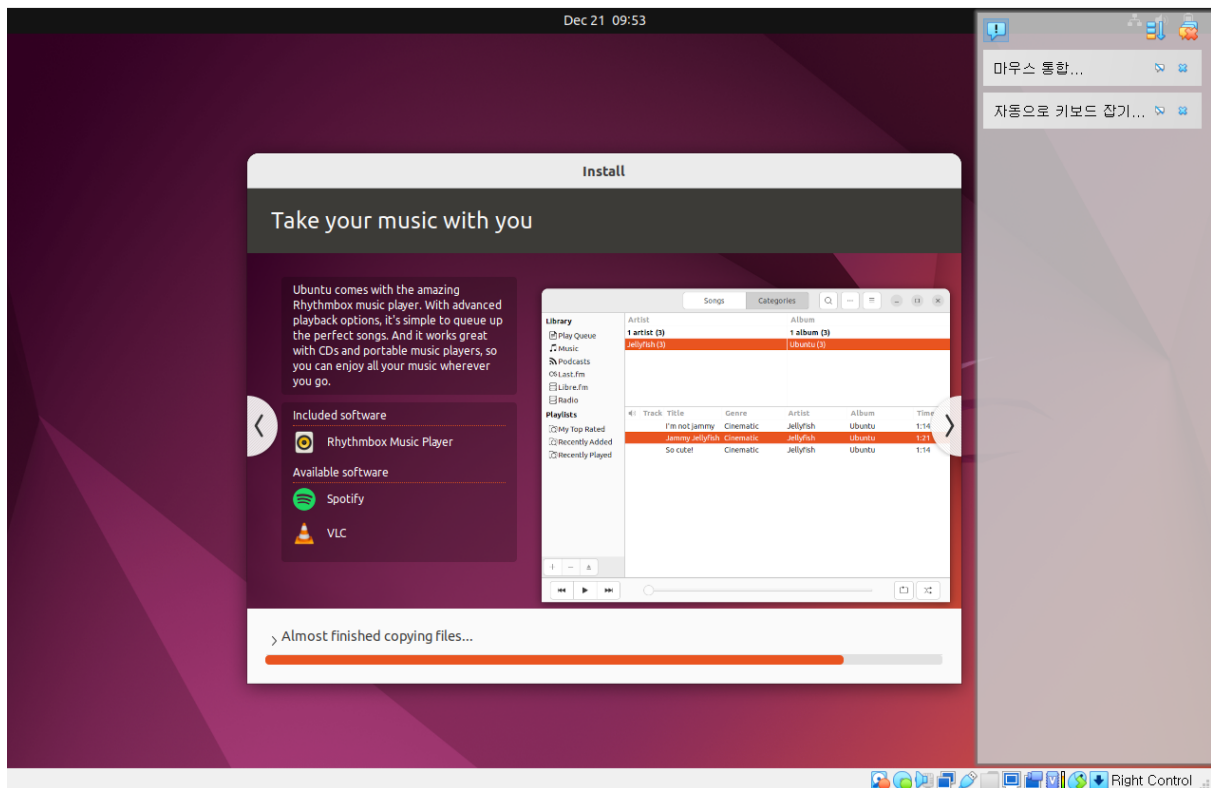
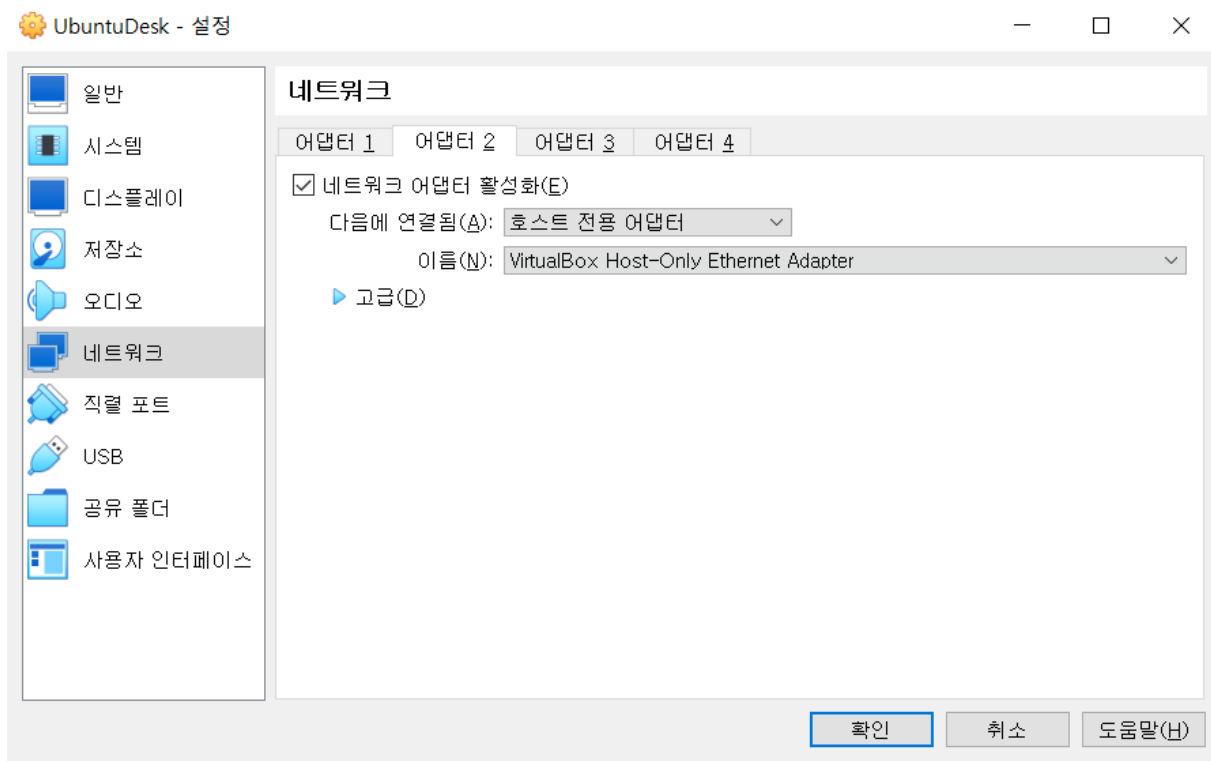
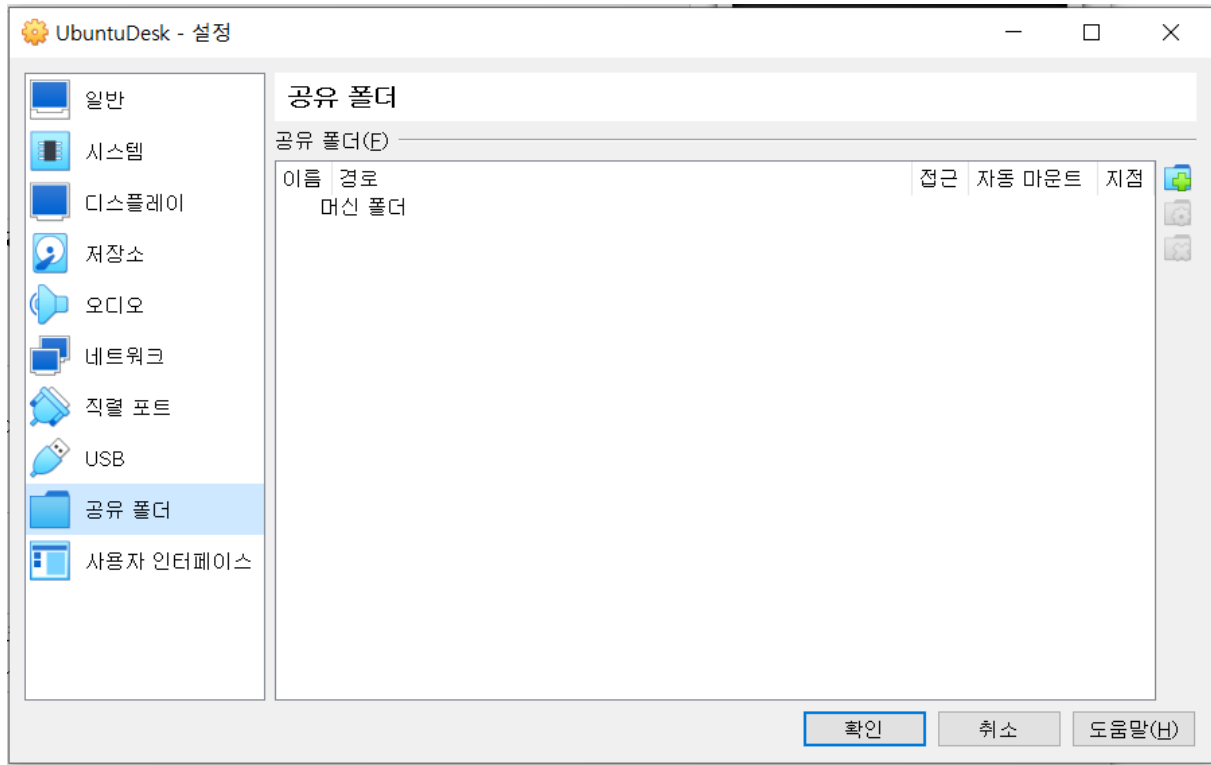


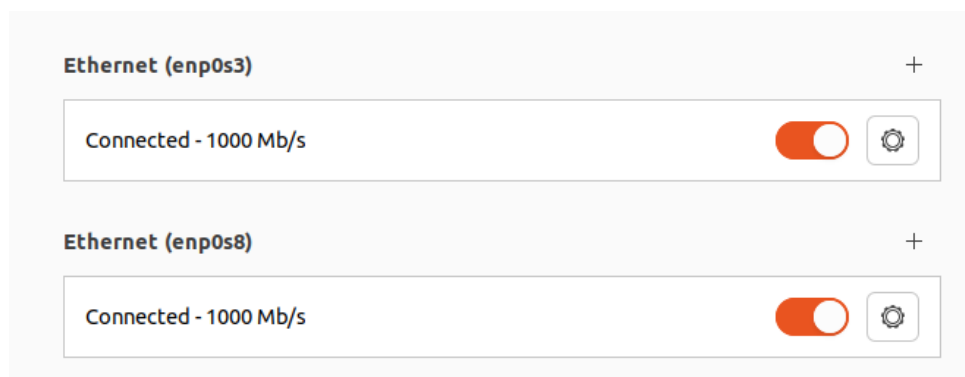
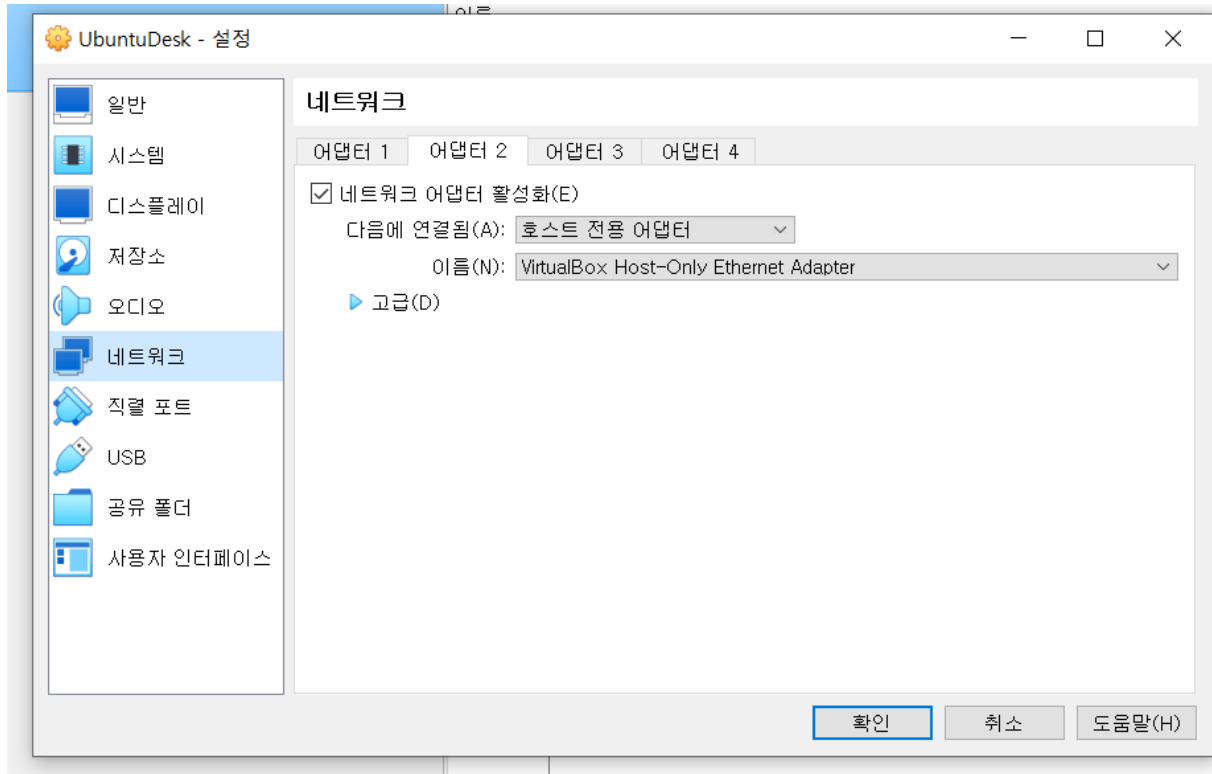
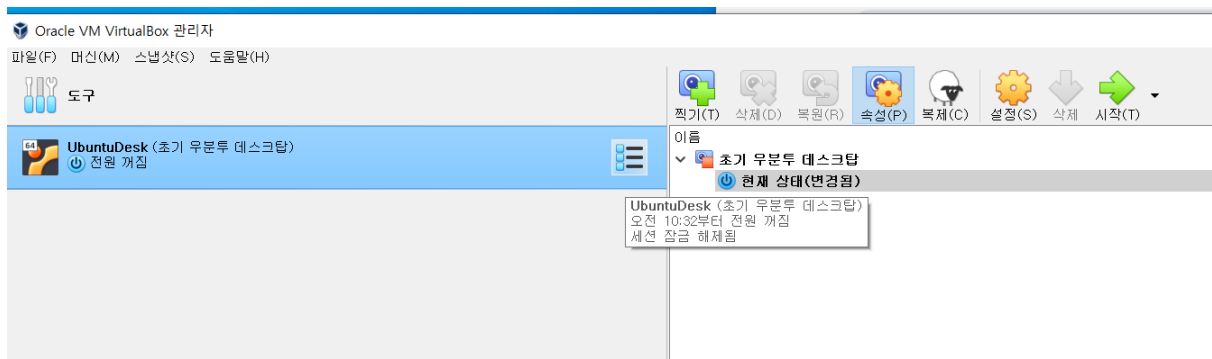


Virtual Box - Ubuntu Desk - Window

≡ 제목





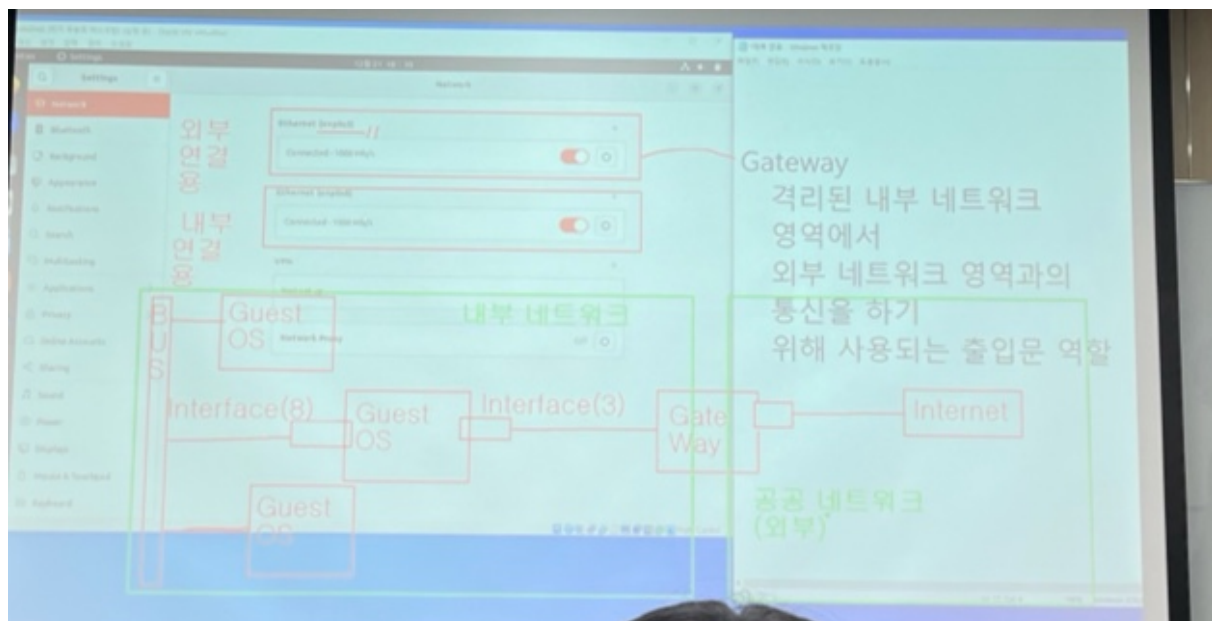
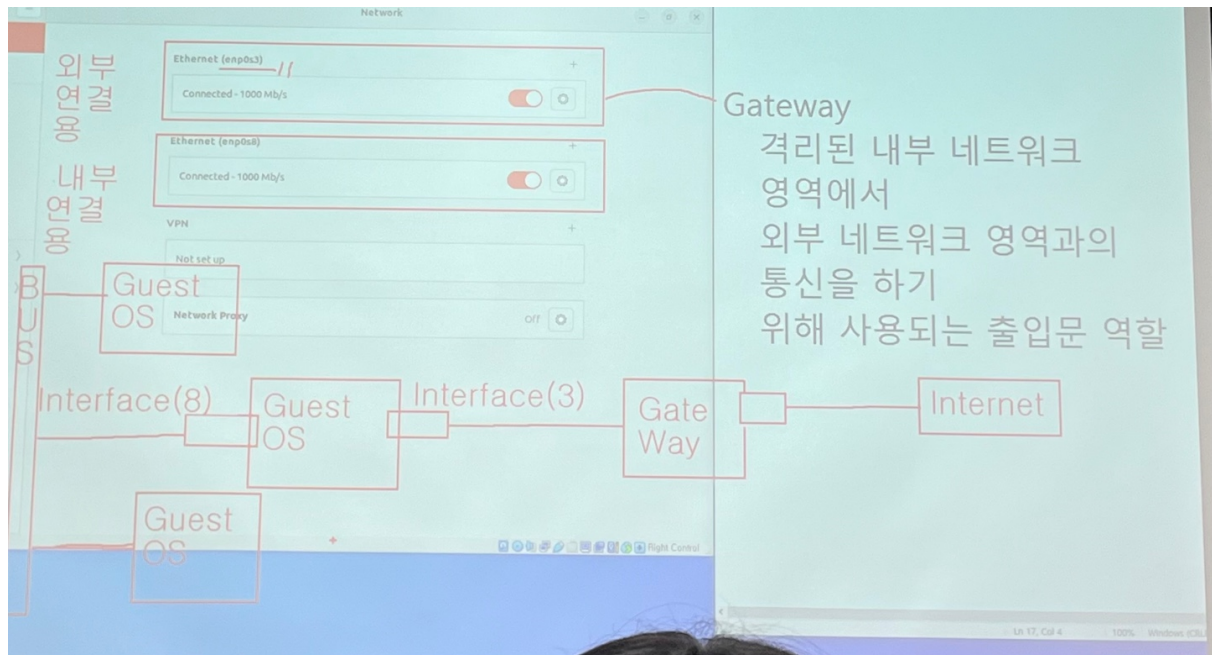


위에있는 3번이 외부 연결용 - 외부하고 연결 시도해볼 때 3번이랑 테스트하면 됨

아래의 8번이 내부 연결용

Gateway

격리된 내부 네트워크 영역에서 외부 네트워크 영역과의 통신을 하기 위해 사용되는 출입문 역할



terminal

ip address

```

ubuntu@UbuntuDesk:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:a8:c5:47 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 85582sec preferred_lft 85582sec
    inet6 fe80::2c93:c771:bed6:632c/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:32:04:bc brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global dynamic noprefixroute enp0s8
        valid_lft 382sec preferred_lft 382sec
    inet6 fe80::4199:1975:2048:38dc/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
ubuntu@UbuntuDesk:~$

```

inet



- 위 사진에서 24를 말하는 거

prefix 아니면 CIDR 이라고 부른다

이 값은 0 에서 32까지만 사용 가능

→ 이거 사용 목적이 뭐냐면 ! prefix/ CIDR : 네트워크 주소에서 변하면 안되는 bit의 범위를 설정하여 네트워크 영역의 크기를 자유롭게 조절하기 위해 사용.

- 여기서 24번째까지는 변하면 안된다는 거.

즉 10.0.2.255 중에서 8비트씩이니깐 10.0.2 여기까지는 변하면 안되는 거 8x3 이니까 3번째 자리까지는 변하면 안된다는 걸 말하는 거

pc 입장에서 나의 범위가 아니라는 것은 ⇒ 외부

외부. 내부 분리되어 있으면 격리되어 있음 → 통신안됨

주소를 어떤 식으로 설정하냐에 따라서 내부 외부 영역 구성 → 격리가 되는 상황 (논리적 격리된 상황)

→ 여기서 둘이 통신을 원하면 GATEWAY 가 필요해지는 거

- IPv4 는 3bit 사용하기 때문에 0에서 32까지 사용가능

10. 여기서 . 점은 하나씩 하나씩 들어가면 왜 4개로 구분짓냐 하나면

이 숫자 하나하나가 8bit 4개가 있음 최대 32bit 를 나눈 거 4개로

그 다음에 8 bit 로 십진수로 나누었는데

8 bit = 0 ~ 255 까지의 숫자로만 사용 가능하다

→ 10.0.2.256 이런 건 없음

```
ubuntu@UbuntuDesk:~$ ip route
default via 10.0.2.2 dev enp0s3 proto dhcp metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
169.254.0.0/16 dev enp0s3 scope link metric 1000
192.168.56.0/24 dev enp0s8 proto kernel scope link src 192.168.56.101 metric 101
```

```
default via 10.0.2.2 dev enp0s3 proto dhcp metric 100
```

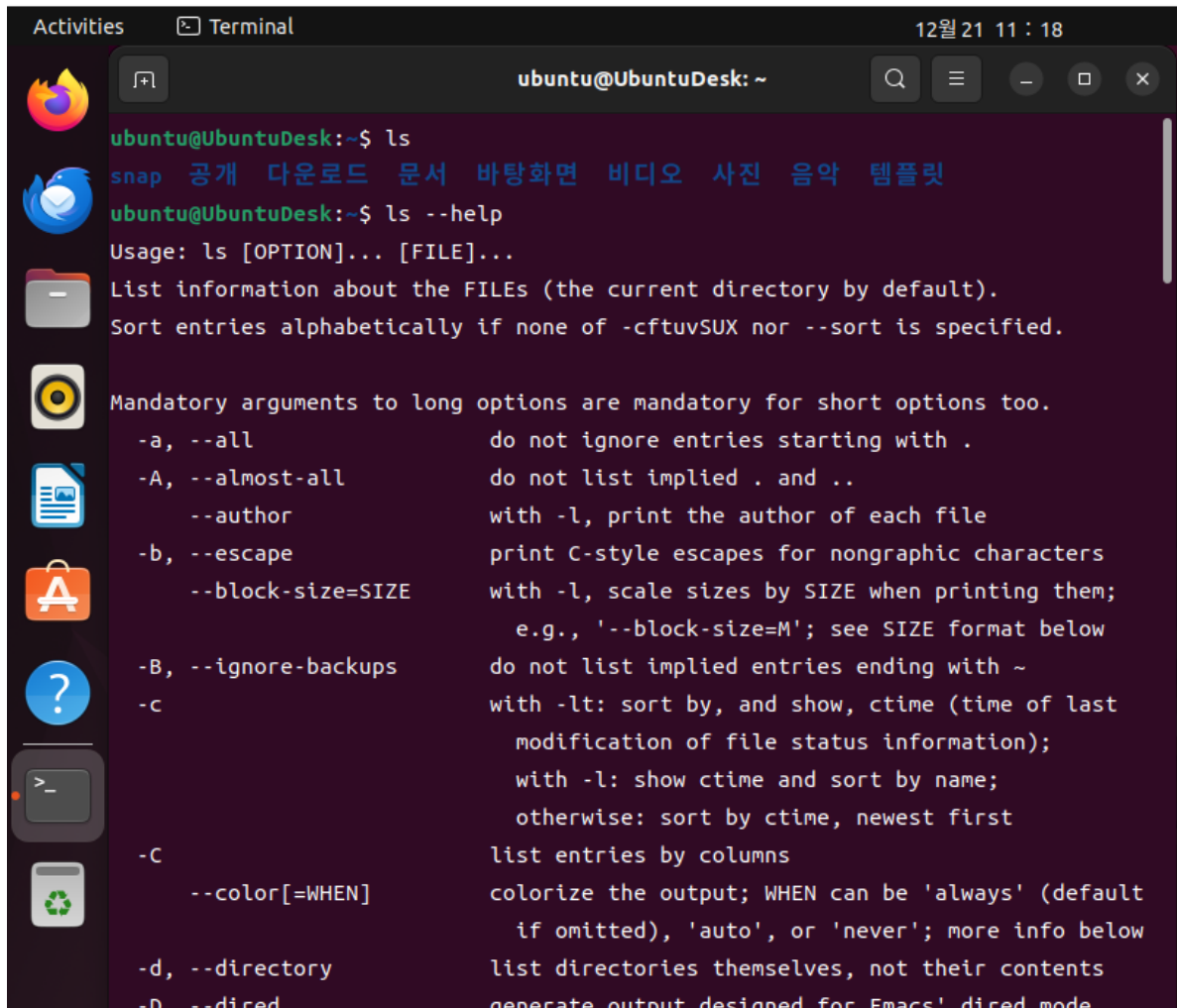
default 이거는 디폴트로 외부 통신과 연결하는데 사용하는 거

필요하면 언제든지 주소 정보를 바꾸는 게 필요하다

```
ubuntu@UbuntuDesk:~$ ls /etc/netplan
01-network-manager-all.yaml
ubuntu@UbuntuDesk:~$ ls
snap  공개  다운로드  문서  바탕화면  비디오  사진  음악  템플릿
ubuntu@UbuntuDesk:~$
```

cat : 특정 파일의 내용을 확인하는 거

파일 내용을 확인하는 거 → 수정 vi → vi 에디터 사용 빔, 나노 사용하는 거 많음.



```
ubuntu@UbuntuDesk:~$ ls
snap  공개  다운로드  문서  바탕화면  비디오  사진  음악  템플릿
ubuntu@UbuntuDesk:~$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
-a, --all                do not ignore entries starting with .
-A, --almost-all        do not list implied . and ..
--author                with -l, print the author of each file
-b, --escape            print C-style escapes for nongraphic characters
--block-size=SIZE       with -l, scale sizes by SIZE when printing them;
                        e.g., '--block-size=M'; see SIZE format below
-B, --ignore-backups    do not list implied entries ending with ~
-c                      with -lt: sort by, and show, ctime (time of last
                        modification of file status information);
                        with -l: show ctime and sort by name;
                        otherwise: sort by ctime, newest first
-C                      list entries by columns
--color=[WHEN]          colorize the output; WHEN can be 'always' (default
                        if omitted), 'auto', or 'never'; more info below
-d, --directory        list directories themselves, not their contents
-D, --dired             generate output designed for Emacs' dired mode
```

명령어들과 명령어 사용할 수 있는 옵션과

그 다음에

-a —all

동일한 거, 단축 옵션을 사용하냐 마냐의 차이가 있는 거

옵션 중에 지금처럼 옵션만 있으면 쓰면 되는데

ls —all : 이걸 쓰기만 하면 돼 → 명령에 맞는 출력 결과가 나옴

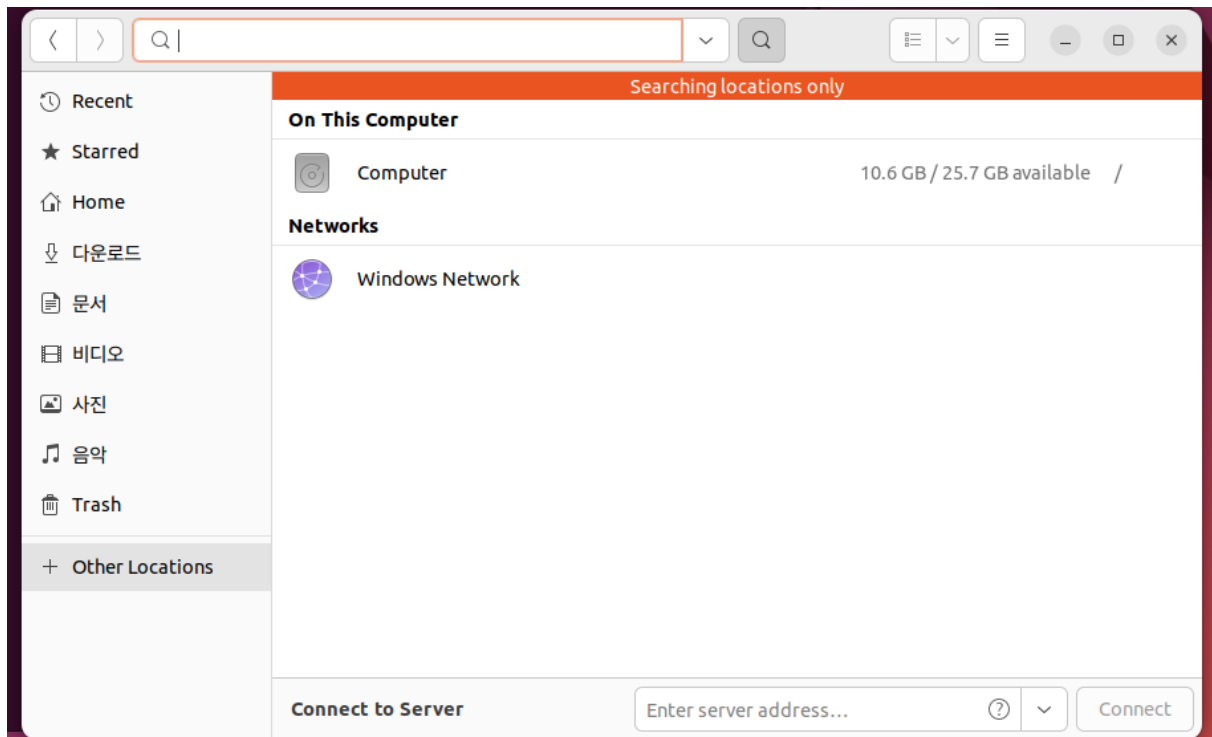
근데 만약에 몇몇 옵션들의 경우에

—blockk -sizs = SIZE : 값을 넣어주는 옵션

man ls : 더 자세하게 메뉴얼이 나옴

종료하고 싶으면 q

gpt 명령어 잘 물어봐!~~!~

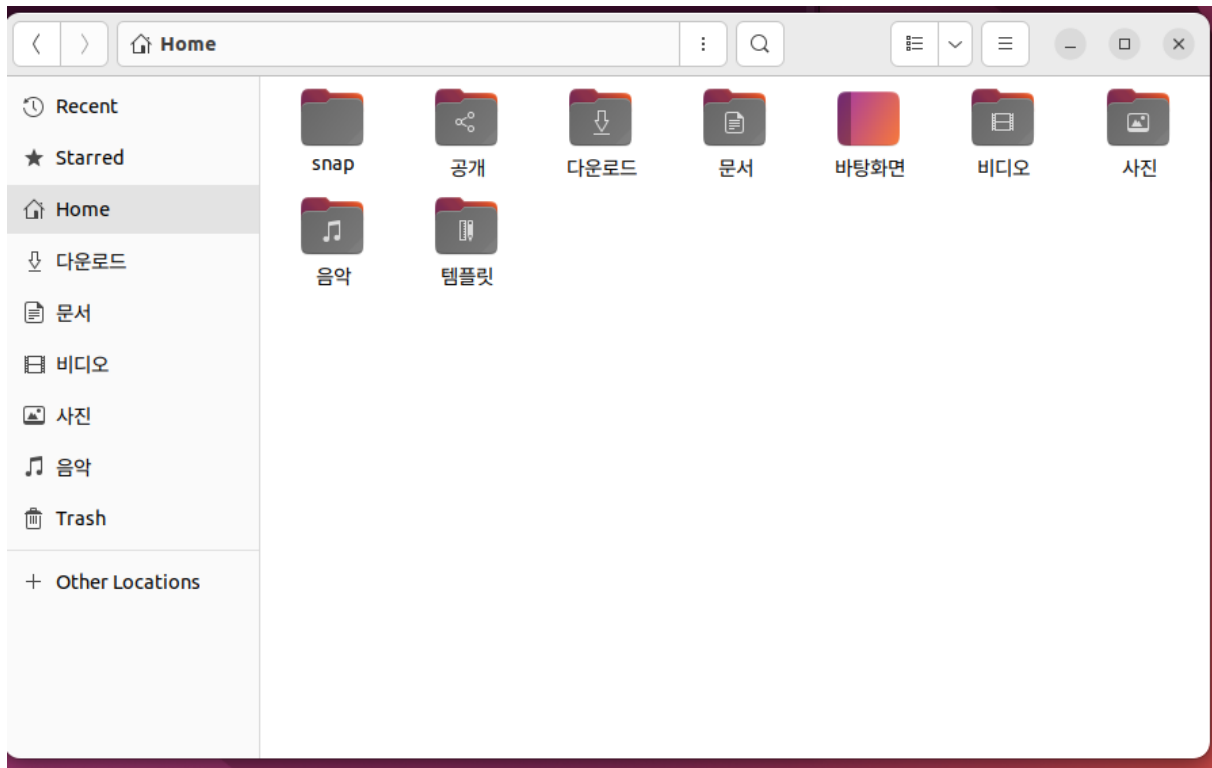


컴퓨터라는 폴더가 루트 폴더다

리눅스에서는 / 이러한 슬래시 하나로 표현 root

윈도우에서는 저 루트가 c:

루트에서 홈 들어가면



/home/ubuntu/

/home/계정명/

→ 이것이 바로 사용자 디렉토리

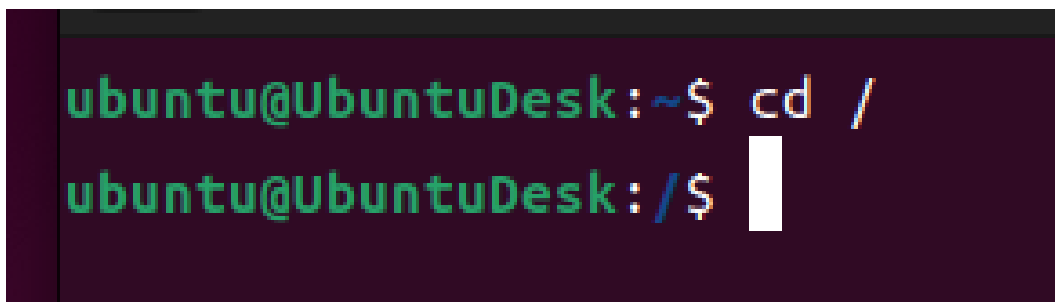
윈도우에서는 사용자의 MZC 여기에 있겠지

C:\Users\mzc

터미널에서 사용자 디렉터리 찾아가는 거

cd 이동

cd



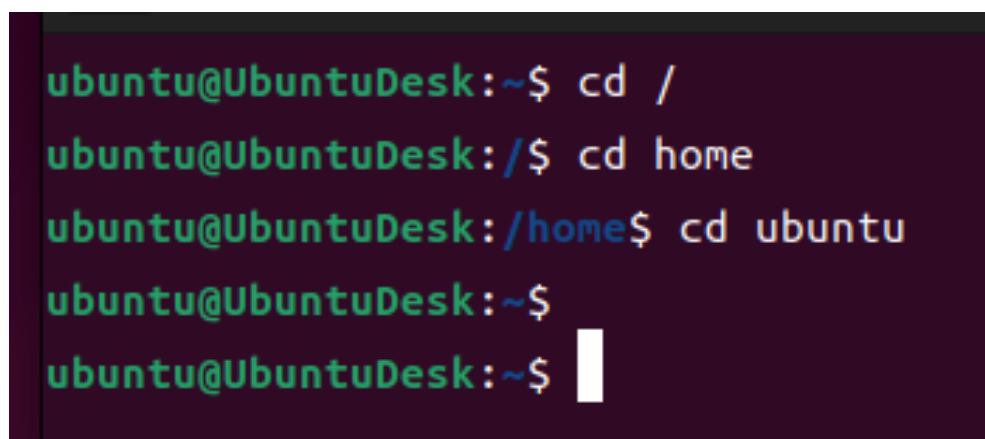
```
ubuntu@UbuntuDesk:~$ cd /  
ubuntu@UbuntuDesk:/$
```

프롬프트 모양이 바뀜

~ 심벌이 / 로 바뀜

~ : 자신의 홈 디렉토리

/: 홈 디렉토리 제외한 나머지 경로는 경로 위치 표기됨. 현재는 root 로 간거



```
ubuntu@UbuntuDesk:~$ cd /  
ubuntu@UbuntuDesk:/$ cd home  
ubuntu@UbuntuDesk:/home$ cd ubuntu  
ubuntu@UbuntuDesk:~$  
ubuntu@UbuntuDesk:~$
```

cd home

```
ubuntu@UbuntuDesk:~$ cd /etc/netplan
ubuntu@UbuntuDesk:/etc/netplan$ cd ..
ubuntu@UbuntuDesk:/etc$ cd ..
ubuntu@UbuntuDesk:/$ pwd
/
ubuntu@UbuntuDesk:/$
```

```
ubuntu@UbuntuDesk:/$ ls
bin    dev    lib    libx32  mnt    root   snap    sys    var
boot   etc    lib32  lost+found  opt    run    srv      tmp
cdrom  home   lib64  media    proc   sbin   swapfile  usr
ubuntu@UbuntuDesk:/$ cd var
```

색상으로 폴더 파일 구분하고 있는데
이것보다! 옵션

```
ubuntu@UbuntuDesk:/var$ ls -l
total 48
drwxr-xr-x  2 root root    4096  4월 18  2022 backups
drwxr-xr-x 17 root root    4096 12월 21  10:07 cache
drwxrwsrwt  2 root whoopsie 4096  8월  8  07:55 crash
drwxr-xr-x 68 root root    4096 12월 21  10:07 lib
drwxrwsr-x  2 root staff   4096  4월 18  2022 local
lrwxrwxrwx  1 root root      9 12월 21  09:52 lock -> /run/lock
drwxrwxr-x 13 root syslog  4096 12월 21  10:32 log
drwxrwsr-x  2 root mail    4096  8월  8  07:52 mail
drwxrwsrwt  2 root whoopsie 4096  8월  8  07:55 metrics
drwxr-xr-x  2 root root    4096  8월  8  07:52 opt
lrwxrwxrwx  1 root root      4 12월 21  09:52 run -> /run
drwxr-xr-x 12 root root    4096  8월  8  07:59 snap
drwxr-xr-x  7 root root    4096  8월  8  07:54 spool
drwxrwsrwt 12 root root    4096 12월 21  11:26 tmp
```

첫 번째 줄의 하이픈

첫 번째 문자

-: 하이픈

d: 디렉토리

r:

l: 링크 (바로가기)

c: 입출력 장치 파일

b: 저장 장치 디바이스 파일

-
- root 디렉토리

→ 리눅스의 모든 디렉토리의 시작점

- bin 디렉토리 /bin

리눅스의 기본 명령어가 저장되어 있는 디렉토리

예) cp, rm, mv, ls 등

- boot 디렉토리 /boot

리눅스의 부트로더가 들어 있는 디렉토리

부팅 프로세스가 진행되는 과정에 필요한 파일만 있음

→ 이거는 절대로 건들면 안됨

- dev 디렉토리 /dev

시스템 장치 파일이 들어 있는 디렉토리

하드디스크 장치 파일 및 마우스 키보드 등의 파일

- etc 디렉토리 /etc

리눅스 시스템의 설치된 프로그램의 구성 정보 파일이 저장되는 디렉토리

추가적으로 /usr/etc 디렉토리를 사용하기도 함

예) 네트워크 매니저, sshd, apache 등

- home 디렉토리 /home

리눅스 사용자 디렉토리

리눅스 사용자를 생성하면 기본적으로 생성되어 사용되는 디렉토리

- lib 디렉토리 /lib

커널이 필요로 하는 커널 모듈 파일과 프로그램(c, python)에 필요한 라이브러리 파일

- mnt 디렉토리 /mnt

cd 및 dvd 또는 usb 장치들의 마운트 포인트로 사용되는 디렉토리

- opt 디렉토리 /opt

우분투에서 제공되지 않는 프로그램을 추가로 설치할 경우에 사용되는 디렉토리

- proc 디렉토리 /proc

메모리에 존재하는 모든 작업들이 파일 형태로 존재할 때 위치하는 디렉토리

- usr 디렉토리 /usr

리눅스를 실행하는 다양한 시스템에서 마운트 할 수 있도록 공유 가능한 읽기 전용 데이터를 보유하는 디렉토리

- var 디렉토리 /var

로그 파일과 같이 크기가 변경될 수 있는 파일이 포함되어 있는 디렉토리

/var/log 해당 디렉토리에 로그파일이 저장되어 있음.

마지막 줄에 있는 해당하는 로그를 확인해 볼 수 있음

- run 디렉토리 /run

Run-time variable data 관리하는 디렉토리

시스템 정보를 관리하는 디렉토리로 시스템 정보를 찾아볼 수 있음

- tmp 디렉토리 /tmp

공유 디렉토리로 모든 사용자가 사용할 수 있는 임시 디렉토리

재부팅 시 디렉토리 내용을 삭제되며, 정기적으로 10일 정도 기간을 두고 디렉토리 내용을 삭제한다

```
ubuntu@UbuntuDesk:/home$ cd home
bash: cd: home: No such file or directory
ubuntu@UbuntuDesk:/home$ cd /home
ubuntu@UbuntuDesk:/home$
```

/안붙였을때 상대경로

→ home 에서 home 인지 못함

/ 붙었을 때 절대경로

→ home 에서 home 인지

./ → 현재위치

../ → 상위 디렉토리

무조건 절대경로 사용하면 어디든지 갈 수는 있지만 이동 경로가 매우 길어지게썻(..??//

```
ubuntu@UbuntuDesk:/home$ cd /etc/netplan
ubuntu@UbuntuDesk:/etc/netplan$
```

절대경로

```
ubuntu@UbuntuDesk:/$ cd ~
ubuntu@UbuntuDesk:~$ ls
snap  공개  다운로드  문서  바탕화면  비디오  사진  음악  템플릿
ubuntu@UbuntuDesk:~$ mkdir test
ubuntu@UbuntuDesk:~$ ls
snap  test  공개  다운로드  문서  바탕화면  비디오  사진  음악  템플릿
ubuntu@UbuntuDesk:~$ rm test
rm: cannot remove 'test': Is a directory
ubuntu@UbuntuDesk:~$ rmdir test
ubuntu@UbuntuDesk:~$ ls
snap  공개  다운로드  문서  바탕화면  비디오  사진  음악  템플릿
ubuntu@UbuntuDesk:~$
```

mkdir → 디렉토리 만드는 거

rmdir → 디렉토리 삭제

touch → 파일 생성하는 거

rm → 파일 삭제

***rm -rf f1 : 강제로 지워라**

```
ubuntu@UbuntuDesk:~$ touch f1
ubuntu@UbuntuDesk:~$ ls
f1  snap  공개  다운로드  문서  바탕화면  비디오  사진  음악  템플릿
ubuntu@UbuntuDesk:~$ cat f1
ubuntu@UbuntuDesk:~$ echo message > f2
```

f1 파일 만들고 내용 넣고 싶어

echo message > f2 → ? > → 는 redirect f2라는 파일로 돌려보낸 거

```
ubuntu@UbuntuDesk:~$ cat f1
ubuntu@UbuntuDesk:~$ echo message > f2
ubuntu@UbuntuDesk:~$ echo message
message
ubuntu@UbuntuDesk:~$ cat f2
message
ubuntu@UbuntuDesk:~$
```

f2 호출해도 message 나온다

redirection 파일을 임시로 f2 라고 했는데

```
ubuntu@UbuntuDesk:~$ mv f2 readme.md
ubuntu@UbuntuDesk:~$ cat readme.md
message
ubuntu@UbuntuDesk:~$
```

이름 바꾸려면 mv f2 readme.md

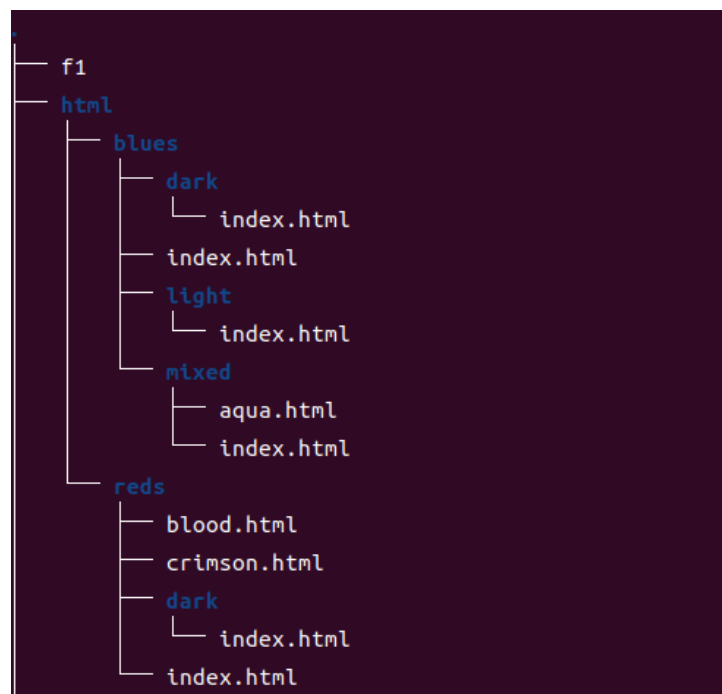
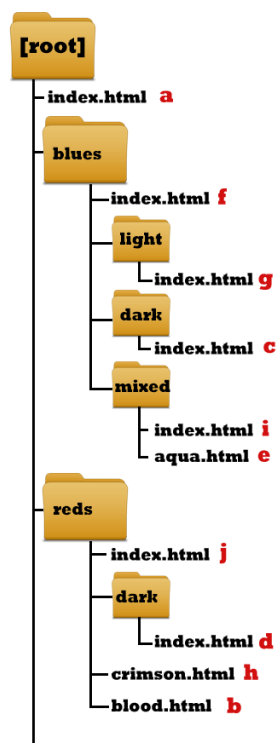
```
ubuntu@UbuntuDesk:~$ mkdir workspace
ubuntu@UbuntuDesk:~$ mv readme.md workspace
ubuntu@UbuntuDesk:~$ ls workspace
readme.md
ubuntu@UbuntuDesk:~$
```

이동 mv

cp

```
ubuntu@UbuntuDesk:~$ touch f1
ubuntu@UbuntuDesk:~$ ls
f1  snap  workspace  공개  다운로드  문서  바탕화면  비디오  사진  음악  템플릿
ubuntu@UbuntuDesk:~$ cp f1 workspace/index.html
ubuntu@UbuntuDesk:~$ cd workspace
ubuntu@UbuntuDesk:~/workspace$ ls
f1  index.html  readme.md
```

`cp f1 workspace/index.html` 명령어는 현재 디렉토리에 있는 `f1` 이라는 파일을 `workspace` 디렉토리로 복사하여 `index.html` 이라는 이름으로 저장하는 명령어입니다.

[illegible]

```
ubuntu@UbuntuDesk:~$ su -
Password:
root@UbuntuDesk:~#
```

```
root@UbuntuDesk:~# apt get install tree
E: Invalid operation get
root@UbuntuDesk:~# apt get install tree
E: Invalid operation get
```

```
ubuntu@UbuntuDesk:~/html$ head -n 5 /var/log/gpu-manager.log
log_file: /var/log/gpu-manager.log
last_boot_file: /var/lib/ubuntu-drivers-common/last_gfx_boot
new_boot_file: /var/lib/ubuntu-drivers-common/last_gfx_boot
can't access /run/u-d-c-nvidia-was-loaded file
can't access /opt/amdgpu-pro/bin/amdgpu-pro-px
ubuntu@UbuntuDesk:~/html$
```

처음 5줄만 보겠다

```
ubuntu@UbuntuDesk:~/html$ tail -n 5 /var/log/gpu-manager.log
Has intel? no
Has nvidia? no
How many cards? 1
Has the system changed? No
Single card detected
ubuntu@UbuntuDesk:~/html$
```

마지막 5줄


```

ubuntu@UbuntuDesk:~/html$ more -n 5 /var/log/gpu-manager.log
log_file: /var/log/gpu-manager.log
last_boot_file: /var/lib/ubuntu-drivers-common/last_gfx_boot
new_boot_file: /var/lib/ubuntu-drivers-common/last_gfx_boot
can't access /run/u-d-c-nvidia-was-loaded file
can't access /opt/amdgpu-pro/bin/amdgpu-pro-px
Looking for nvidia modules in /lib/modules/6.2.0-39-generic/kernel
Looking for nvidia modules in /lib/modules/6.2.0-39-generic/updates/dkms
Looking for amdgpu modules in /lib/modules/6.2.0-39-generic/kernel
Looking for amdgpu modules in /lib/modules/6.2.0-39-generic/updates/dkms
Is nvidia loaded? no
Was nvidia unloaded? no
Is nvidia blacklisted? no
Is intel loaded? no
Is radeon loaded? no
Is radeon blacklisted? no
Is amdgpu loaded? no
Is amdgpu blacklisted? no
Is amdgpu versioned? no
--More-- (57%)

```

more 더보기

```

ubuntu@UbuntuDesk:~/html$ find ./ -name index.html
./reds/index.html
./reds/dark/index.html
./blues/mixed/index.html
./blues/index.html
./blues/light/index.html
./blues/dark/index.html

```

find

./ 루트에서부터

-name 파일명이

index.html 인 거 찾아라!~!

```
ubuntu@UbuntuDesk:~/html$ find ./ -name *.html
./reds/index.html
./reds/blood.html
./reds/crimson.html
./reds/dark/index.html
./blues/mixed/aqua.html
./blues/mixed/index.html
./blues/index.html
./blues/light/index.html
./blues/dark/index.html
```

디스크 정보 확인

```
ubuntu@UbuntuDesk:~/html$ df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
tmpfs	200644	1568	199076	1%	/run
/dev/sda3	25106692	14111384	9694624	60%	/
tmpfs	1003220	0	1003220	0%	/dev/shm
tmpfs	5120	4	5116	1%	/run/lock
/dev/sda2	524252	6220	518032	2%	/boot/efi
tmpfs	200644	120	200524	1%	/run/user/1000

```
ubuntu@UbuntuDesk:~/html$
```

```

ubuntu@UbuntuDesk:~/html$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
tmpfs           200644      1568    199076   1% /run
/dev/sda3       25106692 14111384   9694624  60% /
tmpfs           1003220        0    1003220   0% /dev/shm
tmpfs           5120         4        5116   1% /run/lock
/dev/sda2       524252     6220    518032   2% /boot/efi
tmpfs           200644      120    200524   1% /run/user/1000
ubuntu@UbuntuDesk:~/html$ du
4      ./reds/dark
8      ./reds
4      ./blues/mixed
4      ./blues/light
4      ./blues/dark
16     ./blues
28     .
ubuntu@UbuntuDesk:~/html$ ls -lh
total 8.0K
drwxrwxr-x 5 ubuntu ubuntu 4.0K 12월 21 14:12 blues
drwxrwxr-x 3 ubuntu ubuntu 4.0K 12월 21 14:15 reds

```

사용자 관리 쪽

- 계정추가

```

ubuntu@UbuntuDesk:~/html$ su -
Password:
root@UbuntuDesk:~# useradd user1 -m -s /bin/bash
root@UbuntuDesk:~# useradd user2
root@UbuntuDesk:~# ls -l

```

```

root@UbuntuDesk:~# password user1

```

이런 식으로 비번 설정

/bin/bash

/bin/sh

```

root@UbuntuDesk:/home/ubuntu/html# userdel user2
root@UbuntuDesk:/home/ubuntu/html# su user2
su: user user2 does not exist or the user entry does not contain all the required fields
root@UbuntuDesk:/home/ubuntu/html#

```

계정 삭제

사용자 정보 저장되

어 있음

시스템 계정 정보도 있음

```

root@UbuntuDesk:/# tail -n 5 /etc/passwd
gnome-initial-setup:x:126:65534::/run/gnome-initial-setup:/bin/false
hplip:x:127:7:HPLIP system user,,,:/run/hplip:/bin/false
gdm:x:128:134:Gnome Display Manager:/var/lib/gdm3:/bin/false
ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash
user1:x:1001:1001::/home/user1:/bin/bash
root@UbuntuDesk:/#

```

/etc/passwd → 사용자 정보가 저장된 파일, 시스템 계정 정보

```

user1:x:1001:1001::/home/user1:/bin/bash

```

```

root@UbuntuDesk:/# tail -n 5 /etc/shadow
gnome-initial-setup:!:19576:0:99999:7:::
hplip:!:19576:0:99999:7:::
gdm:!:19576:0:99999:7:::
ubuntu:$y$j9T$3ZiWb9k6AEeS0/Y5JChom.$W0DQTuLLq01oJJEBgNf10bNz2c40HHwQ0Gcjqoj4QF6:19712:0:99999:7:::
user1:$y$j9T$PBbOyyD9yBu1xvvrZ0RLi/$K03sq8QZj40JRS5WhG82w7YRAtrwtctyYve8whFrxm9:19712:0:99999:7:::
root@UbuntuDesk:/#

```

/etc/shadow : 암호가 저장된 파일

암호화 되어 있는 것

암호 생성 일자 : 변경 기간: 유효 기간: 만료경고일:만료후경과일자:만료일

```

root@UbuntuDesk:/# tail -n 5 /etc/group
gdm:x:134:
lxd:x:135:
ubuntu:x:1000:
smbashare:x:136:
user1:x:1001:

```

그룹 정보가 들어가 있다

```

root@UbuntuDesk:/# tail -n 5 /etc/gshadow
gdm:!::
lxd:!::
ubuntu:!::
smbashare:!::
user1:!::

```

그룹 비밀번호

tail -n 5 /etc/passwd:

tail -n 5 /etc/shadow:

tail -n 5 /etc/group:

tail -n 5 /etc/gpasswd:

```

user1@UbuntuDesk:~$ ls -la /home/
total 16
drwxr-xr-x  4 root    root    4096 12월 21 15:02 .
drwxr-xr-x 20 root    root    4096 12월 21 09:53 ..
drwxr-x--- 16 ubuntu  ubuntu  4096 12월 21 14:09 ubuntu
drwxr-x---  2 user1   user1   4096 12월 21 15:03 user1
user1@UbuntuDesk:~$

```

소유권, 소유자 (계정, 그룹)

따라서, `ls -la` 명령어를 실행하면 현재 디렉토리의 모든 파일과 디렉토리에 대한 자세한 정보가 나열되며, 숨겨진 파일도 포함됩니다. 출력은 각 파일 또는 디렉토리의 권한, 소유자, 그룹, 파일 크기, 수정일 등을 보여줍니다. 이 명령어는 파일 시스템의 구조를 이해하고 파일 및 디렉토리의 권한 및 속성을 확인하는 데 유용합니다.



rwX → 계정

r-x → 그룹

-- → other

r: read

w: write

x: execute

디렉토리 만들려면 write 권한이 필요한데 없다..... 그래서

```
user1@UbuntuDesk:~$ mkdir /home/ubuntu/f1
mkdir: cannot create directory '/home/ubuntu/f1': Permission denied
user1@UbuntuDesk:~$ ls -la /home/
```

permission denied 라는 경고가 뜨는 것

```
ubuntu@UbuntuDesk:~$ chmod o+rwX ./
ubuntu@UbuntuDesk:~$ su user1
```

→ 권한을 주는 걸로

```
user1@UbuntuDesk:/home/ubuntu$ mkdir /home/ubuntu/fa
user1@UbuntuDesk:/home/ubuntu$ ls
'♣'$'\213'$'\232'$'\234'$'\223\234'  html
'♣'$'\204'$'\224\224'$'\230'$'        '♣'$'\202'$'\204'
'공♣'$'\234'                          '♣'$'\235\214'$'\225\205'
'문♣'$'\204\234'                      '♣'$'\205\234'$'\224\214'$'\205'
'♣'$'\224'$'\203\225'$'\231\224'$'\205'  snap
f1                                       workspace
fa
user1@UbuntuDesk:/home/ubuntu$
```

다시 생성 가능해짐

파일

r: 파일 읽기 권한 (cat, head, tail ...)

w : 파일 쓰기 권한 (touch, echo message > filename, ...)

x: 실행권한 (.sh, .py등의 스크립트 실행 파일)

디렉토리

r: 디렉토리 내부의 파일 및 디렉토리 정보를 읽기 위한 권한 (ls)

w: 디렉터리 내부의 파일 생성 및 삭제를 위한 권한 (mkdir, touch, rm, mv ...)

x: 디렉터리 접근과 관련된 권한

```
..... rwx rwx rwx
777 (111 111 111) 2
666 (110 110 110) 2
555 (101 101 101)2
444 (100 100 100) 2
000
```

root 계정은 필요할 때만

임시로 하는 거 SUDO : 관리자 권한으로 실행해라

```
ubuntu@UbuntuDesk:~$ sudo apt-get install tree
[sudo] password for ubuntu:
ubuntu is not in the sudoers file. This incident will be reported.
ubuntu@UbuntuDesk:~$
```

sudo 를 사용할 수 있도록 또 한 번 권한을 줘야함

sudoers file 이걸 수정해야해

su, sudo diff

`su` 와 `sudo` 는 둘 다 사용자가 슈퍼유저(root) 권한을 사용하여 명령을 실행할 수 있게 하는 명령어입니다. 그러나 두 명령어는 사용 방식과 동작에서 차이가 있습니다.

1. `su` (Switch User):

- **용도:** 다른 사용자로 전환하기 위한 명령어입니다.
- **사용법:** `su` 명령어 다음에 전환할 사용자 계정을 지정합니다.

```
bashCopy code
su <사용자 계정>
```

- **동작:** `su` 를 사용하면 현재 사용자가 다른 사용자로 전환되어 해당 사용자의 환경과 권한을 사용하게 됩니다. 비밀번호를 입력하여 전환할 수 있습니다.
- **예시:**

```
bashCopy code
su user2
```

2. `sudo` (Superuser Do):

- **용도:** 현재 사용자에게 슈퍼유저 권한을 일시적으로 부여하여 특정 명령어를 실행하게 하는 명령어입니다.
- **사용법:** `sudo` 명령어 다음에 실행하려는 명령어를 지정합니다.

```
bashCopy code
sudo <실행할 명령어>
```

- **동작:** `sudo` 를 사용하면 현재 사용자에게 일시적으로 슈퍼유저 권한이 부여되어 지정된 명령어를 실행할 수 있습니다. 사용자의 비밀번호를 입력하여 권한을 얻을 수 있습니다.
- **예시:**

```
bashCopy code
sudo apt-get update
```

차이점 요약:

- `su` 는 사용자를 전환하는 데 사용되며 전환된 사용자의 환경을 사용합니다.
- `sudo` 는 현재 사용자에게 일시적으로 슈퍼유저 권한을 부여하여 특정 명령어를 실행하게 합니다.
- `su` 는 전환할 사용자의 비밀번호를 요구하고, `sudo` 는 현재 사용자의 비밀번호를 요구합니다.
- `sudo` 는 `/etc/sudoers` 파일에서 지정된 권한에 따라 특정 사용자가 특정 명령어를 실행할 수 있는 권한을 제어할 수 있습니다.

