

# 사용자와 파일권한 관리

# Host의 이해

네트워크에서

Host는 컴퓨터 주인을 뜻한다.

O / X ????

네트워크에 연결되어 있는 **장치, 컴퓨터**를  
Host 라고 한다.

리눅스는

다중 사용자 시스템으로 설계 되었다.

- 한 컴퓨터를 여러명에서 사용함

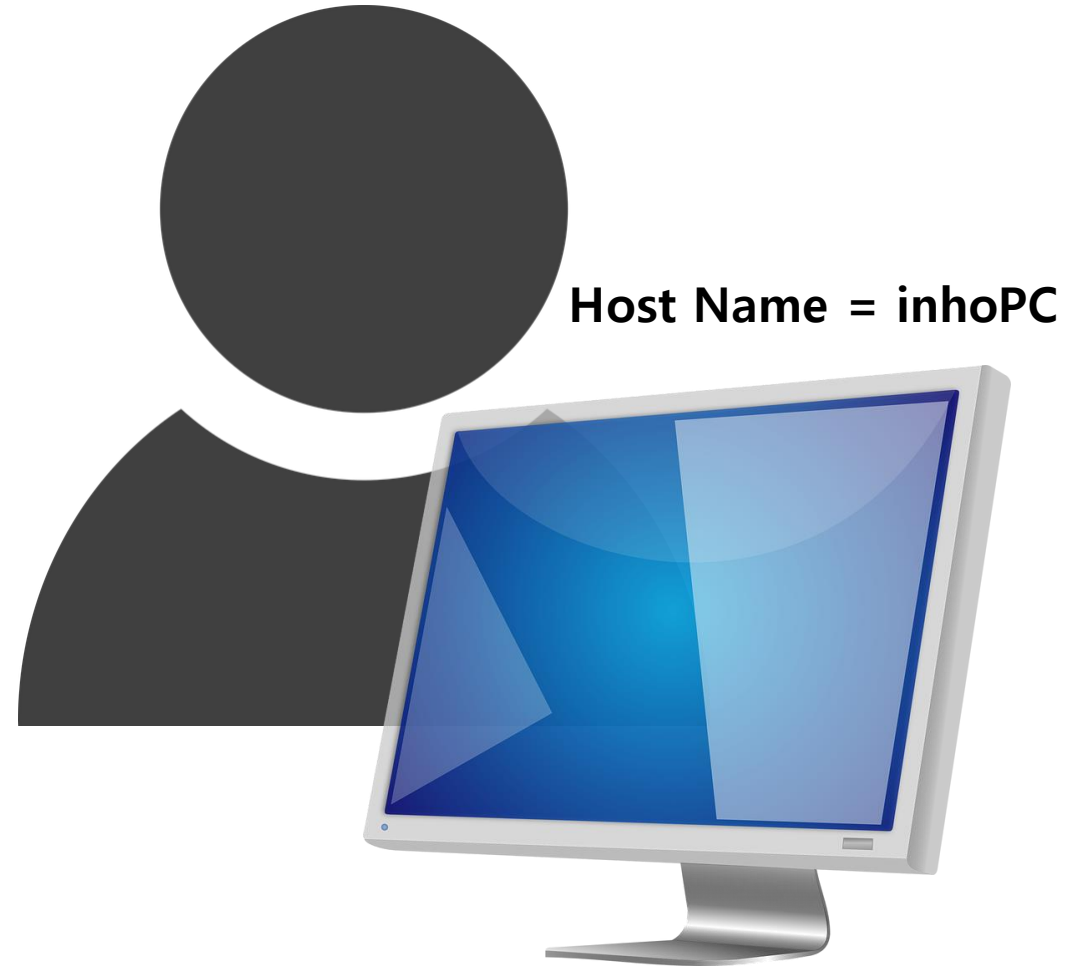
네트워크에 연결된 한 컴퓨터 개념으로

리눅스가 설치된 컴퓨터 한 대를 Host 라고 부른다.

- 여러분의 컴퓨터가 Host이다.

## 리눅스로 가능한 것

- 나 혼자 써야지



## 리눅스로 가능한 것 2

- 우리 가족이 전체 쓰는 컴퓨터
- 각자 계정으로 로그인 가능

## 관리자 지정 필요

- 공용 프로그램들  
설치 / 제거를 모두에게 허용할 순 없다.

User Name  
= inho



User Name  
= kim

User Name  
= dong

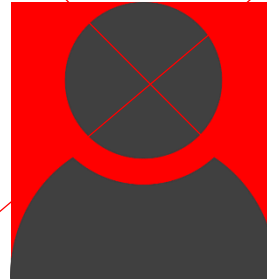


Host Name = inhoPC

## 관리자 지정 필요

- 공용 프로그램들  
설치 / 제거를 모두에게 허용할 순 없다.

**administator**  
User Name  
= inho



User Name  
= kim

User Name  
= dong



관리자는 이렇게 설정하는 되는 것이 아니다.



Host Name = inhoPC



## 공용 프로그램 관리

- 관리자만 건드리도록 해야한다.
- root 계정이 있어,  
root 계정이 프로그램 추가 / 삭제 가능

User Name  
= inho



User Name  
= kim

User Name  
= dong



root



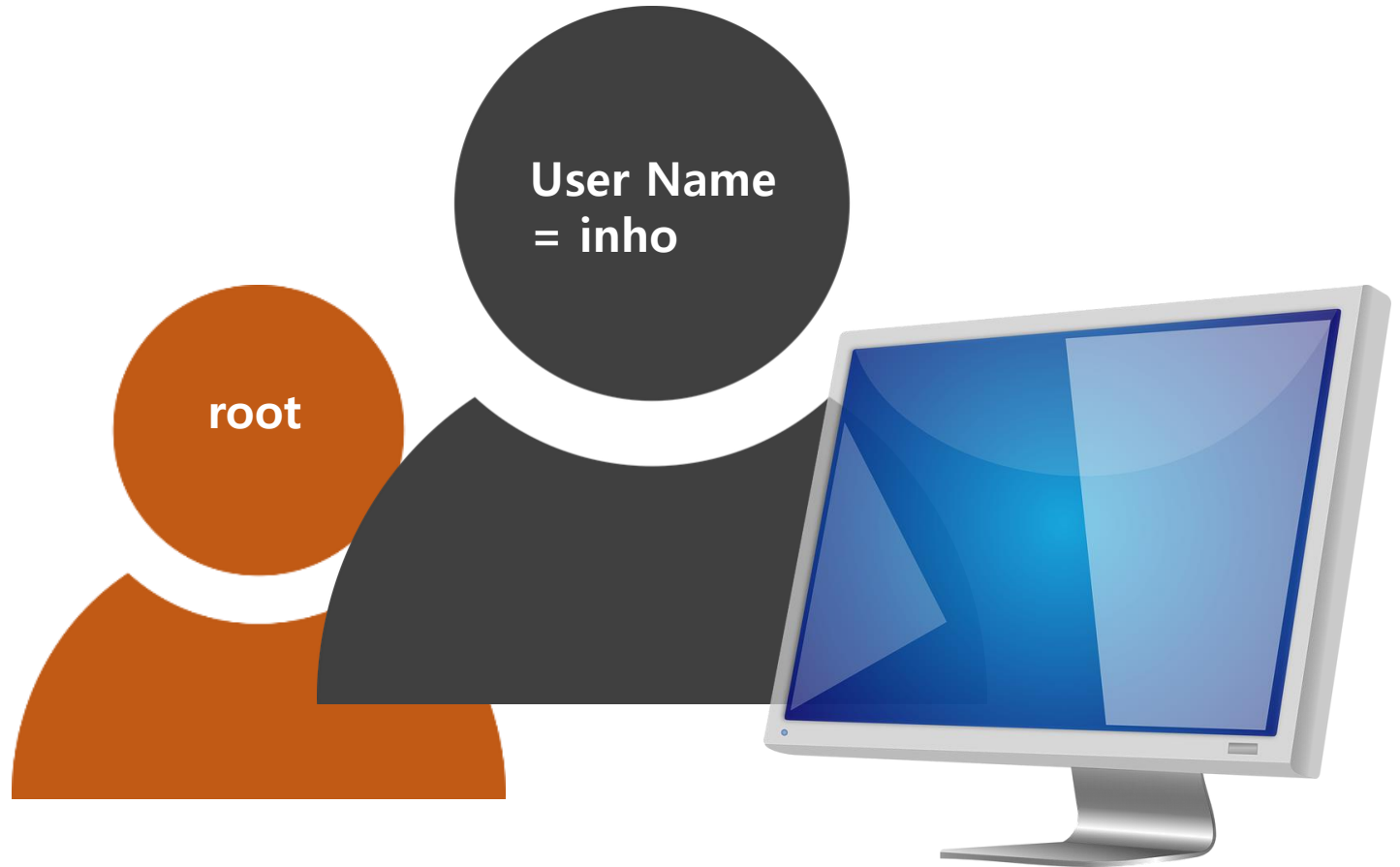
관리자는 root 이다.



Host Name = inhoPC

## 리눅스 처음 설치하면 만들어지는 계정

- Host Name : inhoPC
- User ID : inho
- root 도 자동 생성됨



inho / kim / dong 은  
각자만의 파일들을  
남이 건들지 못하게  
설정 가능하다.

- 파일 권한설정 이라고 한다.

User Name  
= inho



User Name  
= kim

User Name  
= dong



root



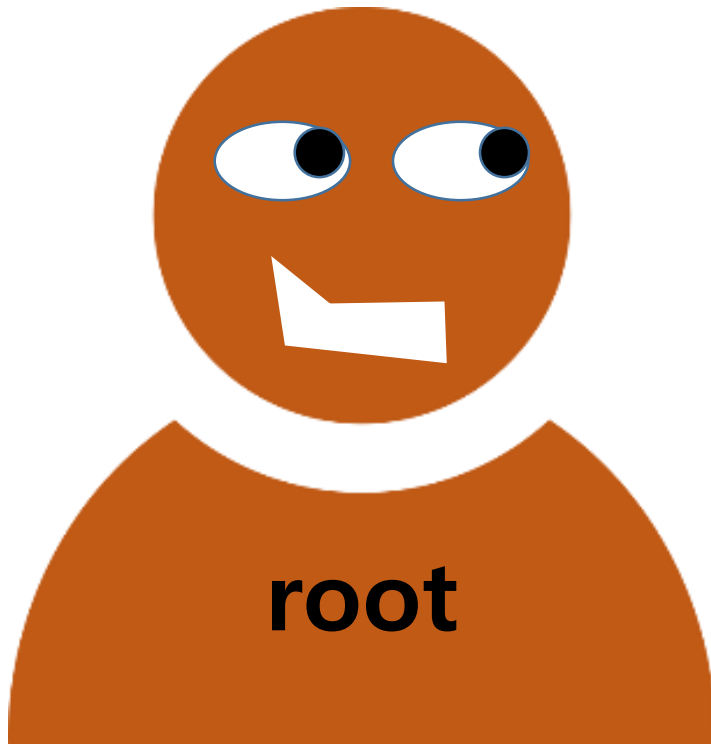
Host Name = inhoPC

# 그럼 Root 사용자는 모든 파일을 다 뒤적거릴 수 있는가?

Confidential

O / X ?????

- 아무리 User들이 권한 설정 해봤자 root 는 다 볼수있나?



User Name  
= inho



User Name  
= kim

User Name  
= dong



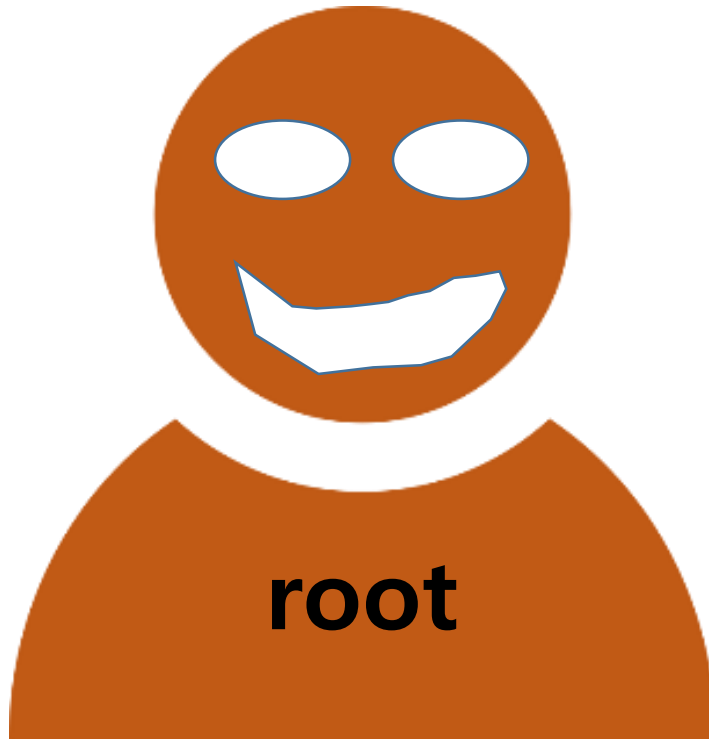
Host Name = inhoPC

# 그럼 Root 사용자는 모든 파일을 다 뒤적거릴 수 있는가?

Confidential

그렇다. 다 볼 수 있다.  
모든 권한을 가진다.

- root 로 아무나 접속을 못하게  
보안에 신경 많이 써야한다.



User Name  
= inho



User Name  
= kim

User Name  
= dong



Host Name = inhoPC

여기서 Host 는 누구 누구인가?



여기서 User계정은 누구 누구인가?



## 1. Host 설정하기

## 2. user 계정 만들기

- user 계정 만들어서 로그인 해보기
- group 계정 만들기

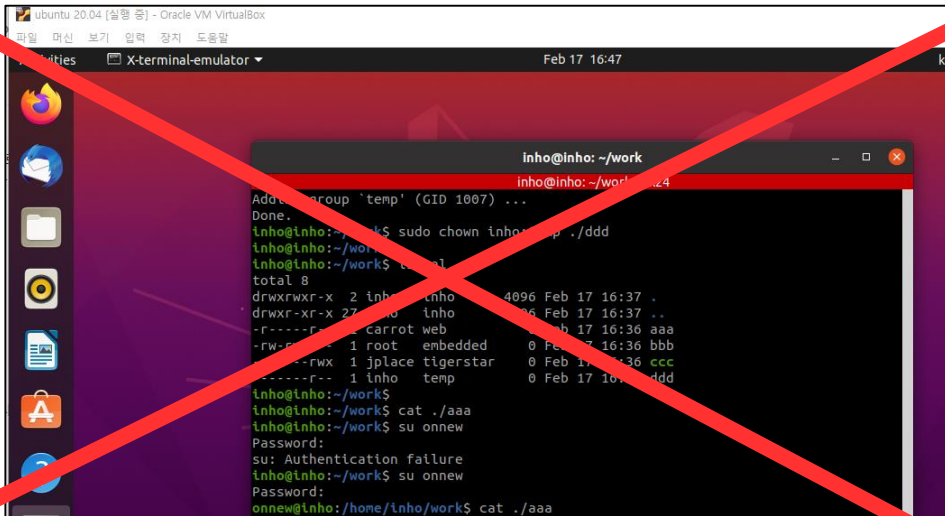
## 3. 파일 권한 설정하기



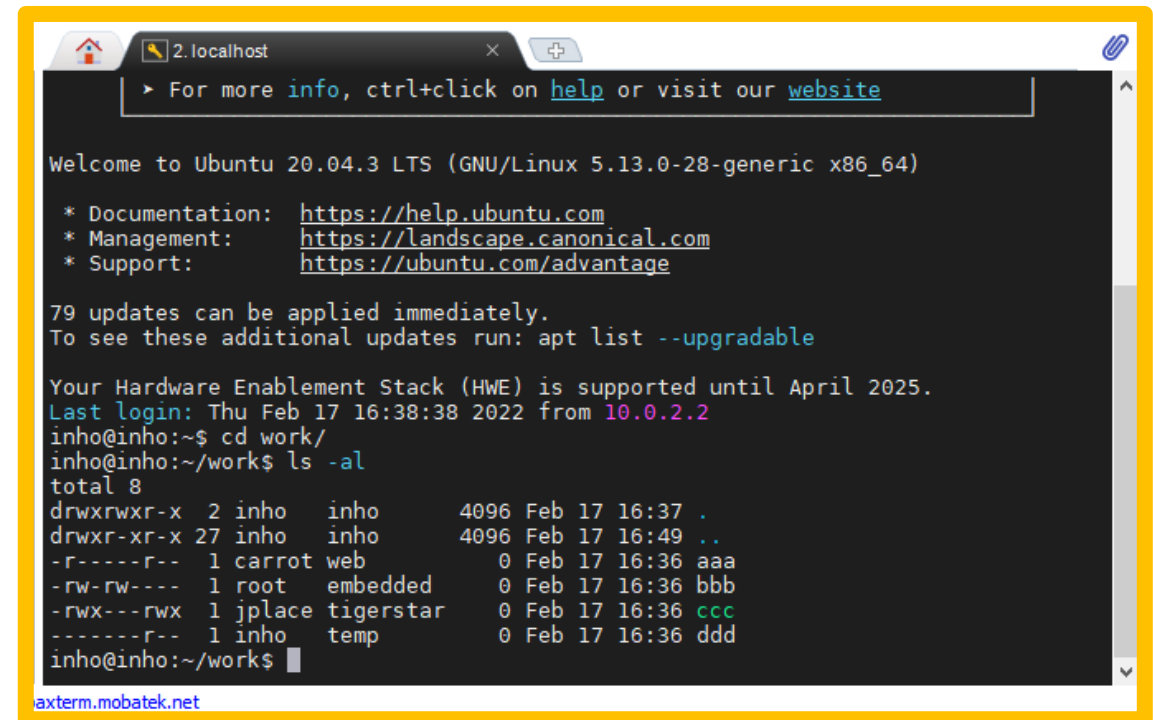
## Host 세팅 - 기본

## Linux 원격접속하여 실습을 한다.

- Linux 직접 실행이 아닌,  
원격 터미널 연결로 권한 테스트를 하자.
- Root권한 사용자에게 권한이 더 부여되어,  
실습시 결과가 달라지는 문제가 발생



```
ubuntu 20.04 [실행 중] - Oracle VM VirtualBox
X-terminal-emulator Feb 17 16:47
incho@incho: ~/work
incho@incho:~/work$ sudo chown incho:temp ./ddd
incho@incho:~/work$ ls -la
total 8
drwxrwxr-x 2 incho incho 4096 Feb 17 16:37 .
drwxr-xr-x 27 incho incho 4096 Feb 17 16:37 ..
-r-----r-- 1 carrot web 0 Feb 17 16:36 aaa
-rw-rw---- 1 root embedded 0 Feb 17 16:36 bbb
-rwx---rwx 1 jplace tigerstar 0 Feb 17 16:36 ccc
-----r-- 1 incho temp 0 Feb 17 16:36 ddd
incho@incho:~/work$
incho@incho:~/work$ cat ./aaa
incho@incho:~/work$ su onnew
Password:
su: Authentication failure
incho@incho:~/work$ su onnew
Password:
onnew@incho:/home/incho/work$ cat ./aaa
```



```
2. localhost
> For more info, ctrl+click on help or visit our website

Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.13.0-28-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

79 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Thu Feb 17 16:38:38 2022 from 10.0.2.2
incho@incho:~$ cd work/
incho@incho:~/work$ ls -la
total 8
drwxrwxr-x 2 incho incho 4096 Feb 17 16:37 .
drwxr-xr-x 27 incho incho 4096 Feb 17 16:49 ..
-r-----r-- 1 carrot web 0 Feb 17 16:36 aaa
-rw-rw---- 1 root embedded 0 Feb 17 16:36 bbb
-rwx---rwx 1 jplace tigerstar 0 Feb 17 16:36 ccc
-----r-- 1 incho temp 0 Feb 17 16:36 ddd
incho@incho:~/work$
```

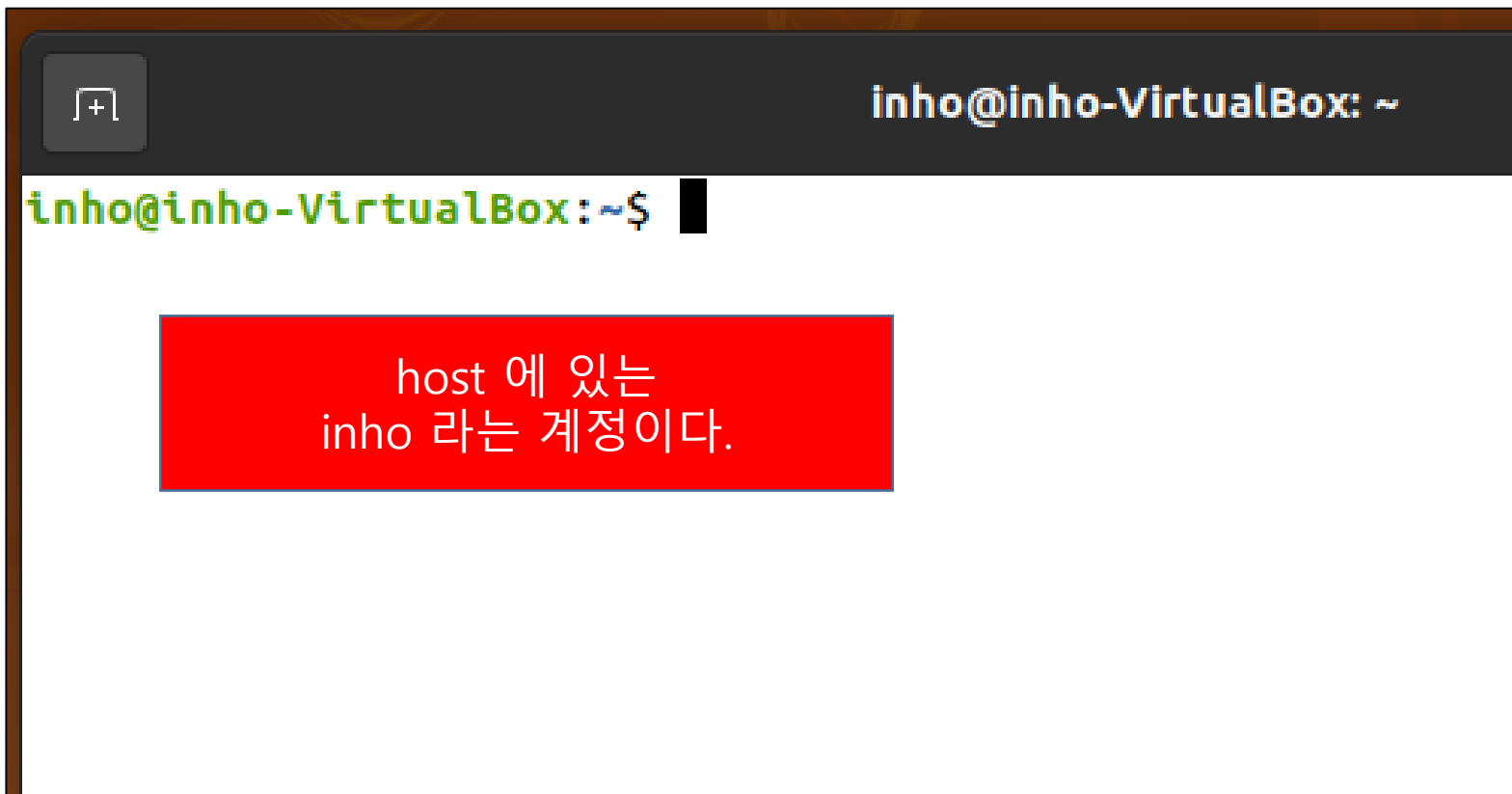
여기에는 host name과 user name이 적혀있다.

- host name과 user name을 구분해보자.

A terminal window with a dark gray title bar. On the left of the title bar is a square button with a white icon of a terminal window with a plus sign. On the right of the title bar is the text 'inho@inho-VirtualBox: ~' in a light gray font. The main area of the terminal is white and contains the text 'inho@inho-VirtualBox:~\$' in a green and blue monospace font, followed by a black cursor bar.

여기에는 host name과 user name이 적혀있다.

- host name과 user name을 구분해보자.



```
inho@inho-VirtualBox: ~$
```

host 에 있는  
inho 라는 계정이다.

확실히 구분하자.

user name

host name

```
inho@com:~$  
inho@com:~$  
inho@com:~$
```

해당 host에 로그인된  
사용자 이름

host name

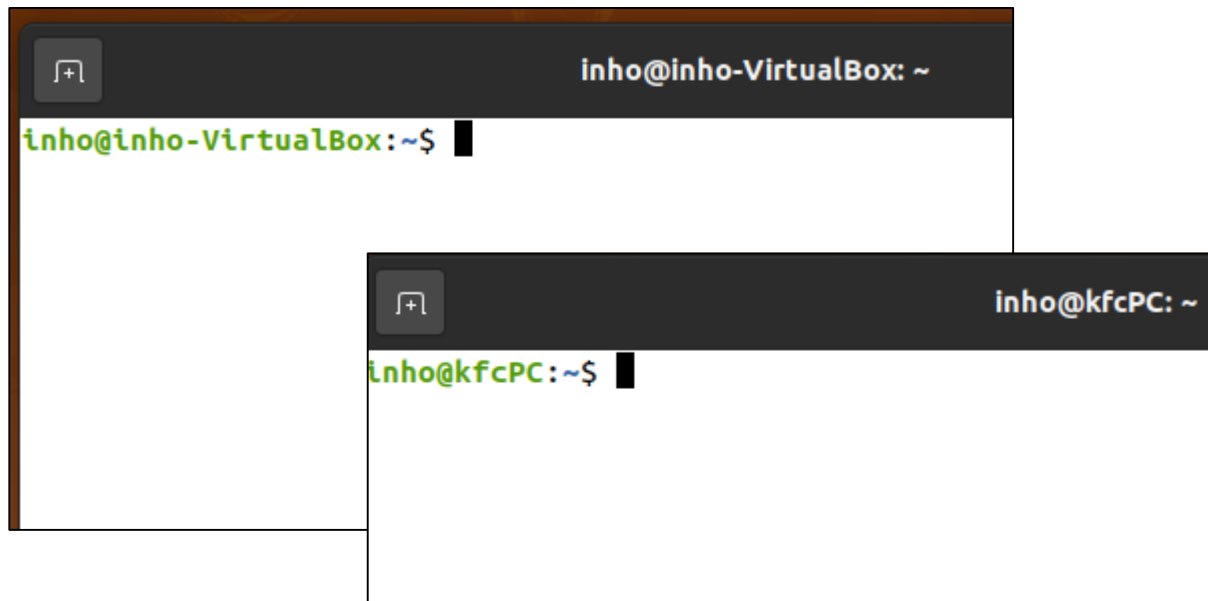
## Host Name 바꾸기

### 1. 관리자 권한으로 /etc/hostname 내용 편집 [암기 안해도 됨]

- `sudo vi /etc/hostname`
- 이름 바꾸고 저장

### 2. 리셋 명령어 [암기]

- `sudo reboot`



Host : 네트워크에 연결된 장치

Host name : 장치를 지목할 때 쓰는 이름이다.

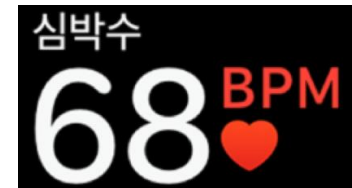
- 네트워크의 특정 PC를 지목하는 방법 1 : ip 주소
- 네트워크의 특정 PC를 지목하는 방법 2 : hostname

네트워크에서, 살아있는지 체크하는  
ping Test 를 해보자.

- 본인에게 맞는 host 네임과 ip주소를 입력하자.

```
jony@jonyPC:~$ ping jonyPC
PING jonyPC (10.0.2.15) 56(84) bytes of data.
64 bytes from jonyPC (10.0.2.15): icmp_seq=1 ttl=64 time=0.021 ms
64 bytes from jonyPC (10.0.2.15): icmp_seq=2 ttl=64 time=0.017 ms
64 bytes from jonyPC (10.0.2.15): icmp_seq=3 ttl=64 time=0.017 ms
```

```
jony@jonyPC:~$ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.013 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.017 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.017 ms
```





Host PC에서

지금 사용중인 User 의 비밀번호 바꾸어보자.

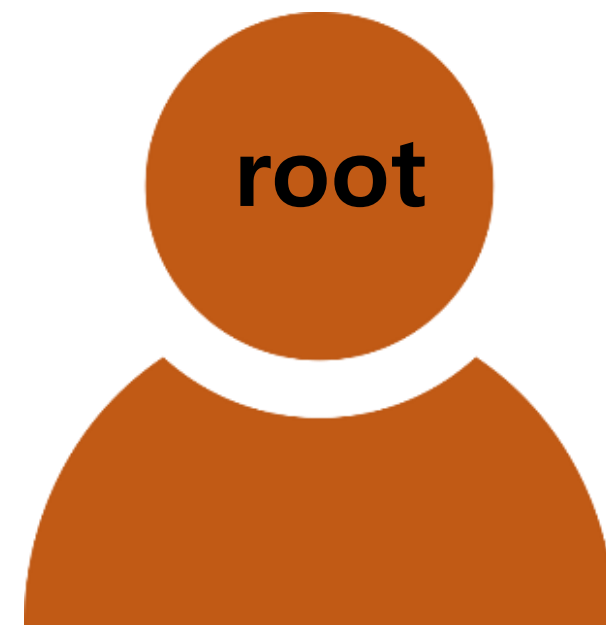
- User 의 이름도 바꿀 수 있다. (잘 안쓰이니 수업에서 생략한다.)
- User 의 비밀번호만 바꿔보자.

본인의 비밀번호를 바꾸어보자.

명령어 이름 : passwd

- 자신의 비밀번호를 asdf1234 으로 바꾼다.

리눅스 설치할 때 root 비밀번호 세팅한 적이 없다.  
그럼 root로 어떻게 로그인할까?



root로 로그인을 막아두었다!

- 보안을 위해 막아둠
- 설정 바꾸면 풀 수 있다.

먼저 일반 계정으로 접속하여

→ 사용자 변경을 통해 root 계정 사용 가능

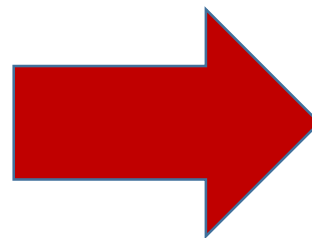
## 1. `sudo su` 입력

- 사용자 변경

## 2. `exit` 입력

- root를 빠져나와서, 다시 원래 user로 돌아온다.

User Name  
= inho



root



# 아무나 root로 변경 가능한 것이 아니다.

Confidential

## root 권한이 있는 사람

- root로 변신할 수 있는 user 계정을 지목할 수 있다.

User Name  
= inho

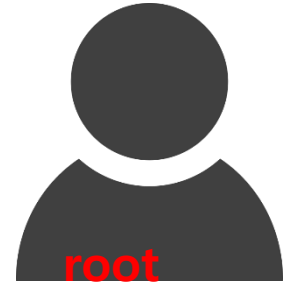


root  
변신 권한



User Name  
= kim

User Name  
= dong



root  
변신 권한



Host Name = inhoPC

root 권한을 가진 계정 이라고 표현한다.

Confidential

root 변신 권한이라고  
부르지 않고,

root 권한을 가진 계정  
이라고 표현한다.

User Name  
= inho



root 권한



User Name  
= kim

User Name  
= dong



root 권한



Host Name = inhoPC

# 그럼 sudo는 도대체 무엇인가?

Confidential

sudo

- 관리자의 힘으로 해당 명령어를 수행한다. 라는 의미
- root 권한으로 명령어 수행

관리자의 힘으로  
특정 명령어를 수행한다.





## 가장 많이 사용하는 것

- 프로그램 추가 / 삭제 하기
- 시스템 중요 파일, vi 편집하기

프로그램 설치할 때  
sudo apt install 하거나

그냥 root계정으로 변경해서,  
apt install 하면 된다.

## 그 밖에

- User 계정 추가
- User 계정 삭제
- 등등

root 계정 변경해서 프로그램을 설치한다.

- sl 이라는 프로그램 설치하기
- sl 프로그램 실행해보기

root 계정을 탈출하고, user 계정에서 프로그램 삭제한다.

- sl 이라는 프로그램 삭제하기

# 임베디드 시스템에서 계정이 필요할까?

Confidential

임베디드 리눅스시스템에서  
사용자 계정 필요할까?

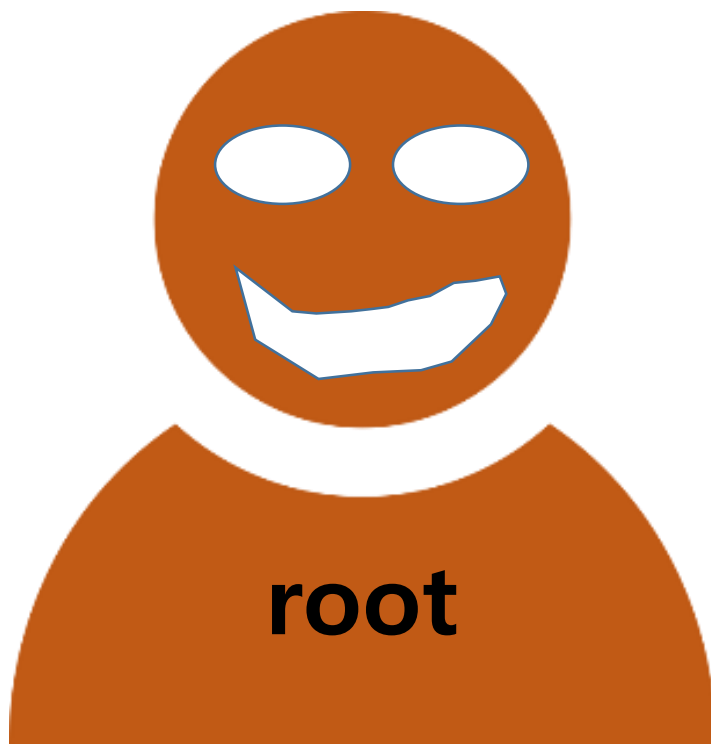
• O / X 퀴즈

User Name  
= inho



User Name  
= kim

User Name  
= dong



Host Name = inhoM

**매우 중요한 내용**, 다음 내용을 반드시 이해해야 한다.

임베디드 리눅스에서는 다음과 같이 동작되도록 한다.

1. 켜자마자 바로 root 로 자동 로그인 되도록 한다.
2. 자동으로 세탁기 App을 실행하도록 만든다.

→ 즉, 켜자마자 root 권한으로 자동 App 실행

Host Name = inhoPC



## Host 기본 세팅 - 시간

리눅스 컴퓨터 한대 세팅하면  
시간부터 세팅한다.

- 세팅 하는 이유??

- 시스템

로그 메시지에 시간 정확히 찍으려고

## 로그메세지

- 사용 이력을 나타내는 메세지
- 에러 발생시  
이곳에 에러메세지들이 찍혀있다.
- 디버깅할 때
  - 필수적으로 분석해야 하는 정보
  - 시간 정보가 중요하다.

```
Dec 1 10:27:19 atlas kernel: [ 25.970847] [drm] DDC: 0x7e50 0x7e50 0
Dec 1 10:27:19 atlas kernel: [ 25.970849] [drm] Encoders:
Dec 1 10:27:19 atlas kernel: [ 25.970850] [drm] CRT2: INTERNAL_KLD
Dec 1 10:27:19 atlas kernel: [ 25.970851] [drm] DFP3: INTERNAL_LVT
Dec 1 10:27:19 atlas kernel: [ 25.981742] type=1400 audit(1291192037.5
Dec 1 10:27:19 atlas kernel: [ 25.982360] type=1400 audit(1291192037.5
Dec 1 10:27:19 atlas kernel: [ 25.982683] type=1400 audit(1291192037.5
Dec 1 10:27:19 atlas kernel: [ 26.054640] hda codec: ALC883: BIOS auto
Dec 1 10:27:19 atlas kernel: [ 26.117222] [drm] fb mappable at 0x000C0
Dec 1 10:27:19 atlas kernel: [ 26.117224] [drm] vram apper at 0x000000
Dec 1 10:27:19 atlas kernel: [ 26.117226] [drm] size 7258112
Dec 1 10:27:19 atlas kernel: [ 26.117227] [drm] fb depth is 24
Dec 1 10:27:19 atlas kernel: [ 26.117228] [drm] pitch is 6912
Dec 1 10:27:19 atlas kernel: [ 26.137222] Console: switching to colour
Dec 1 10:27:19 atlas kernel: [ 26.141579] fb0: radeondrmfb frame buffe
Dec 1 10:27:19 atlas kernel: [ 26.141581] drm: registered panic notifi
Dec 1 10:27:19 atlas kernel: [ 26.141583] Slow work thread pool: Start
Dec 1 10:27:19 atlas kernel: [ 26.141664] Slow work thread pool: Ready
Dec 1 10:27:19 atlas kernel: [ 26.141671] [drm] Initialized radeon 2.5
Dec 1 10:27:19 atlas kernel: [ 26.549625] EXT3-fs (dm-0): using intern
Dec 1 10:27:19 atlas kernel: [ 27.827432] type=1400 audit(1291192039.4
Dec 1 10:27:19 atlas kernel: [ 27.891496] type=1400 audit(1291192039.4
Dec 1 10:27:19 atlas kernel: [ 27.892104] type=1400 audit(1291192039.4
Dec 1 10:27:19 atlas kernel: [ 27.892427] type=1400 audit(1291192039.4
Dec 1 10:27:19 atlas kernel: [ 27.893149] type=1400 audit(1291192039.4
Dec 1 10:27:19 atlas kernel: [ 27.893892] type=1400 audit(1291192039.4
Dec 1 10:27:19 atlas kernel: [ 27.894028] type=1400 audit(1291192039.4
Dec 1 10:27:20 atlas kernel: [ 28.753956] ppdev: user-space parallel p
Dec 1 10:27:28 atlas kernel: [ 36.540046] eth0: no IPv6 routers presen
Dec 1 10:48:02 atlas kernel: [ 1271.135118] show_signal_msg: 21 callback
Dec 1 10:48:02 atlas kernel: [ 1271.135123] npviewer.bin[2912]: segfault
```

로그를 통한, 디버깅을 위해  
현재 시간이  
정확하게 되도록 맞춰주어야한다.

## 현재 시간을 확인할 수 있다.

- date 명령어

## KST

- Korea Standard Time
  - 세계 표준 시간인 UTC + 9시간
  - 국제 표준과 호환되는 시간이다.

```
inho@inho-VirtualBox:~$ date
2021. 03. 05. (금) 16:26:35 KST
inho@inho-VirtualBox:~$
```

```
inho@inho:~$ date
2022. 02. 15. (화) 11:53:02 KST
inho@inho:~$
inho@inho:~$
```



## rdate 설치

- apt 으로 설치하자.
- rdate = remote date

입력 : `sudo rdate time.bora.net`

- LG 유플러스가 운영하는 시간서버

## 이 프로그램이 하는 일

1. 시간 정보를 가지고 있는 **Time Server** 서버 접속
2. Time Server에서 시간 정보를 가져온다.
3. 내 Host에 Time 값을 적용한다.

```
inho@inho:~$ sudo rdate time.bora.net
Tue Feb 15 11:58:01 KST 2022
inho@inho:~$ date
2022. 02. 15. (화) 11:58:07 KST
inho@inho:~$
```

## Linux 세팅하면 현재 시간 부터, 정확히 맞춰주자.

- 클라우드 사용시, 시간이 안 맞춰져있는 경우가 많음
- 삼성전자 / SSAFY 사업장 / 강의장에서는  
Time Server가 접속이 막혀져 있어, rdate 명령어가 안먹히는 경우가 있음
  - 이럴 땐, 어쩔수 없이 Hot spot으로 연결해서 시간을 맞춰주어야 한다.

리눅스 user

root 권한을 가진 계정 이라고 표현한다.

Confidential

root 변신 권한이라고  
부르지 않고,

root 권한을 가진 계정  
이라고 표현한다.

User Name  
= inho



root 권한



User Name  
= kim

User Name  
= dong



root 권한



Host Name = inhoPC

## 새로운 user를 만들면

1. 해당 user 이름으로 로그인 가능하다.
  - root 권한이 없는, 새로운 계정이 만들어진다.
2. 해당 경로에, user 전용 디렉토리가 만들어진다.
  - /home/[user이름]

## adduser 명령어

- 동사가 먼저 나오고, 명사가 나온다.
- useradd 가 아니다. adduser 이다.

```
inho@inho:~$ sudo adduser kfc  
'kfc' 사용자를 추가 중...
```

kfc 계정 만들기

## 새로운 계정을 생성하는 명령어

- **adduser : 편리한 자동 설정**

- 필요한 모든 설정들을 Default 값으로 생성

- **useradd : 전문가용, 세부 옵션을 해주어야한다.**

- 홈 디렉토리, 계정설정, 사용자 UID 설정, 시작 셸 환경 등 세부 설정 가능



이것을 쓰자



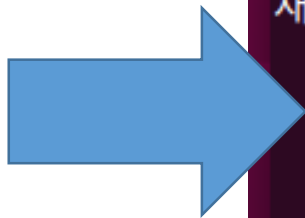
## 사용자를 추가하자

- adduser [사용자명]

```
inho@inho:~$ sudo adduser kfc  
'kfc' 사용자를 추가 중...
```

## 여러 묻는 질문에 **엔터**로 대답해주자.

- 엔터 = Default 값으로 세팅



```
passwd: 암호를 성공적으로 업데이트했습니다  
kfc의 사용자의 정보를 바꿉니다  
새로운 값을 넣거나, 기본값을 원하시면 엔터를 치세요  
이름 []:  
방 번호 []:  
직장 전화번호 []:  
집 전화번호 []:  
기타 []:  
정보가 올바릅니까? [Y/n]
```

사용자 계정을 만들면, 홈디렉토리에 자동 생성 된다.  
잘 만들어졌는지 확인하자.

```
inho@inho:~$ cd /home/  
inho@inho:/home$ ls  
inho  kfc  
inho@inho:/home$
```

exit 명령어를 사용해 로그아웃하고,  
kfc 로 로그인 해보자.

A terminal window titled '7. ubuntu desktop' with a search icon and window controls. It shows two lines of text: 'login as: kfc' and 'kfc@192.168.56.1's password: ' followed by a grey rectangular input field.

```
login as: kfc
kfc@192.168.56.1's password: 
```

원래 계정으로 돌아오기

다시 로그아웃 후,  
원래 계정으로 돌아오기

## 현재 사용자에서

- 잠시 다른 사용자로 이동하기
- 현재 사용자는 로그아웃 하지 않고, 잠시 다른 사용자로 접속한다.

## su [ user계정 이름 ]

- kfc 계정으로 접속하자.

## 탈출하기

- 명령어 exit 입력

파일(F) 편집(E) 보기(V) 검색(S)

```
inho@inho:~$ su kfc
```

암호:

```
kfc@inho:/home/inho$
```

su kfc 접속 후

sudo su 입력하기

- root 로 변경할 권한이 없다.
- adduser로 사용자 계정을 만들면, root 권한 없는 계정이 만들어지기 때문.

```
kfc@inho:/home/inho$ sudo su
[sudo] kfc의 암호:
kfc은(는) sudoers 설정 파일에 없습니다. 이 시도를 보고합니다.
kfc@inho:/home/inho$
kfc@inho:/home/inho$
```

/etc/sudoers 파일을 편집해서,  
kfc도 root 권한을 줄 수 있지만,  
자주쓰는 기능이 아니므로 수업에 생략

userdel이 아니라

**del**user [계정명]

- 동사 먼저 나오는 명령어

kfc 계정을 삭제하자.

home 디렉토리를 확인해보자.

- kfc 잔류 파일들이 남아있다.
- deluser가 깨끗하게 안지워졌다.

deluser의 정확한 사용법을  
찾아보자.

```
inho@inho:/home$ ls
inho  kfc
inho@inho:/home$
inho@inho:/home$
inho@inho:/home$
```



## man 명령어

- Linux 개발자들은 명령어 사용법 / 옵션을 자주 까먹는다.
- **man**ual 을 보고, 옵션을 기억해낸다.

## man deluser 입력하기

- delete 로 검색해보자.

```
--remove-home
    Remove the home directory of the user and its mailpool. If
    --backup is specified, the files are deleted after having per-
    formed the backup.

--remove-all-files
    Remove all files from the system owned by this user. Note:
    --remove-home does not have an effect any more. If --backup is
    specified, the files are deleted after having performed the
    backup.
```

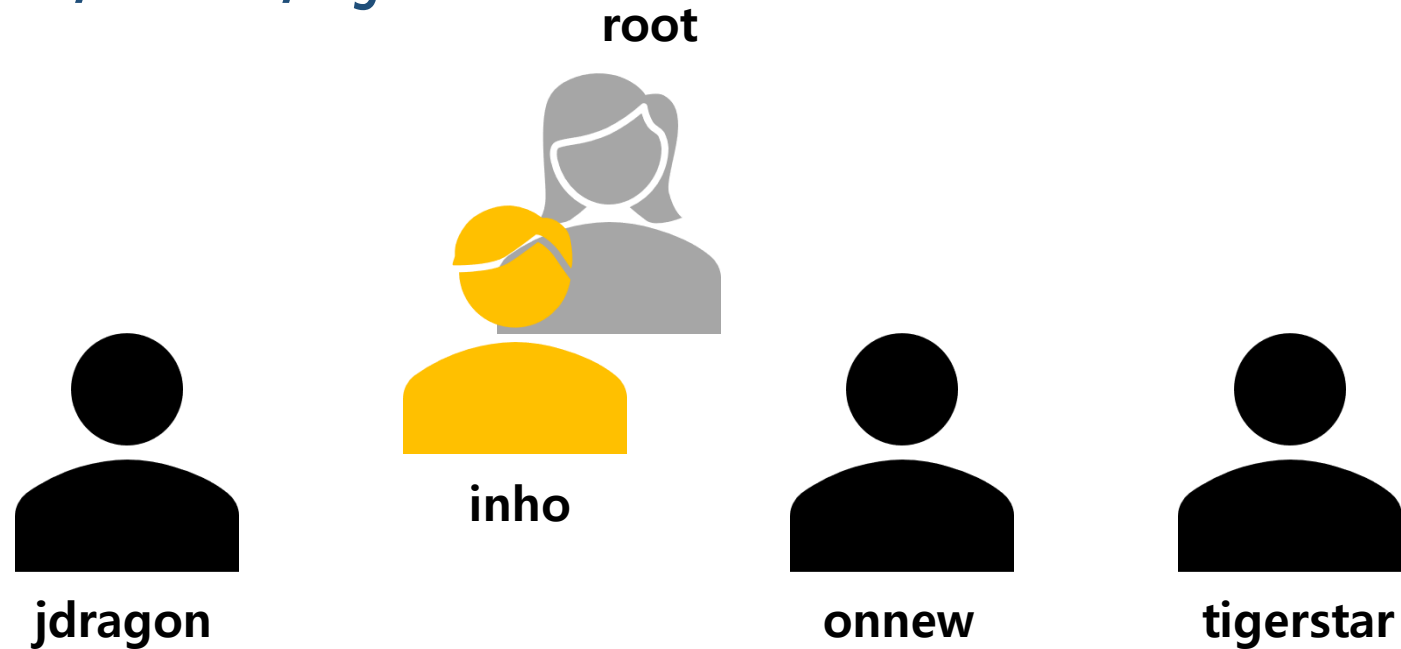
1. kfc 계정을 다시 만들어준다.
2. deluser 명령어 + 모든파일제거 옵션으로 kfc를 깔끔하게 제거한다.
3. Home directory에서 제거된 것을 확인한다.

```
--remove-home
Remove the home directory of the user and its mailpool. If
--backup is specified, the files are deleted after having per-
formed the backup.

--remove-all-files
Remove all files from the system owned by this user. Note:
--remove-home does not have an effect any more. If --backup is
specified, the files are deleted after having performed the
backup.
```

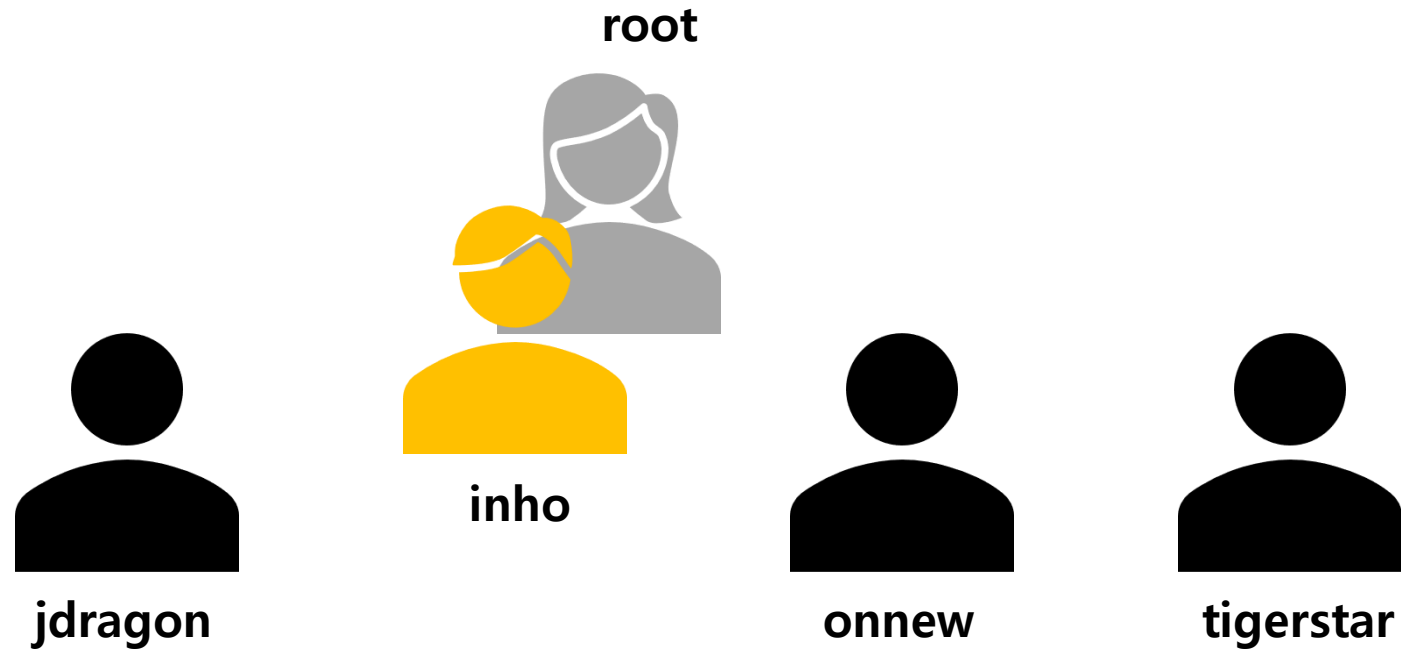
본인계정 외, 다른 기존의 계정을 모두 지우자!  
그리고 실습을 위해, 아래와 같은 사용자들을 만들자.

- 본인 계정 1개 (아래 이미지에서는 inho 라고 표현)
- 새로운 계정 3개 : jdragon / onnew / tigerstar



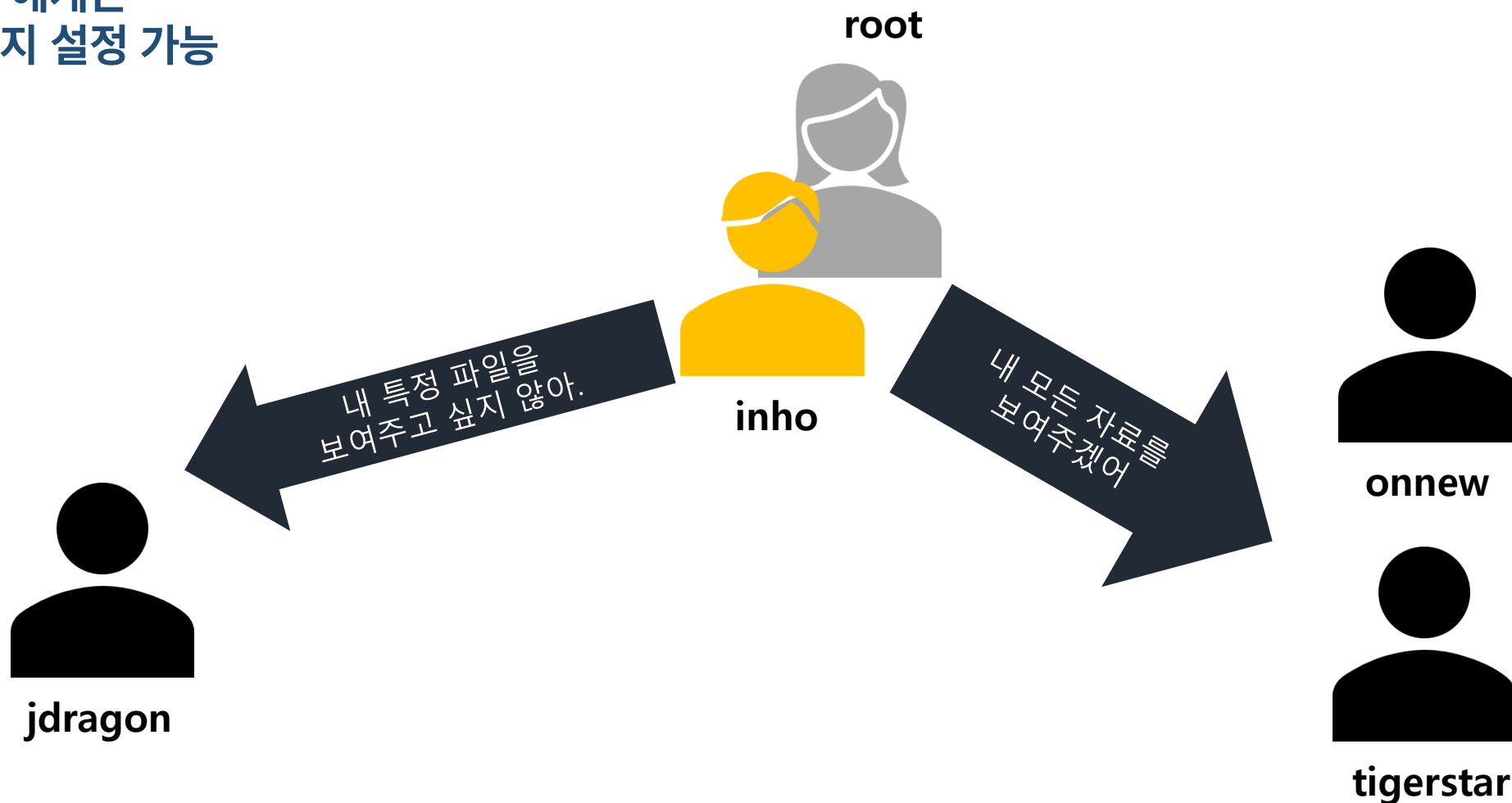
다음 명령어는 몇 번에서 에러가 발생할까?

1. inho user로 로그인
2. su jdragon
3. cd /home/inho
4. ls ./
5. touch bbq.txt



각 사용자들은, 내 파일과 프로그램, 폴더들에 대한 권한 설정 가능!!

- 세세하게, 누구에게는  
얼마나 공개할지 설정 가능



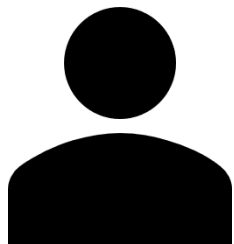
# user들을 묶어, 그룹 별 권한 설정도 가능하다!!

Confidential

## 그룹 단위로 권한 설정이 된다.

- User 들을 묶어 그룹을 만들 수 있다.
- 그룹 단위로, 파일 권한 설정 가능!!

보안팀



jdragon

root



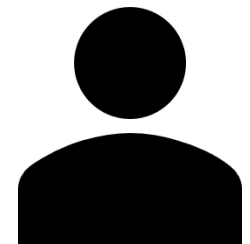
inho

시크릿팀

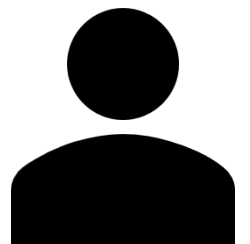


carrot

웹개발팀



onnew



tigerstar

# 그래서, 앞으로 이것을 배울 것이다!!

Confidential

## 1. 그룹 만들기

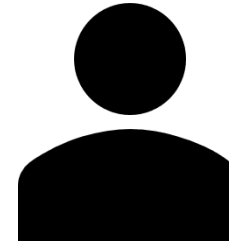
- 그룹을 만든다.
- 만든 그룹 안에, user 들을 포함시킨다.

## 2. 개인 / 그룹 단위, 권한세팅

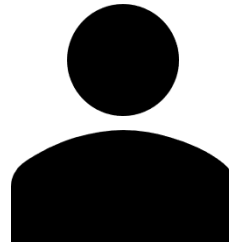
- 문서 파일 / 실행 파일 / 디렉토리에 대한 권한을 세세히 설정해본다.

바로 시작할 내용

웹개발팀



onnew



tigerstar

웹 개발팀 그룹은  
보안팀의 자료를 못보게 설정하자

# group 만들기



## 실습을 위한 세팅

- 아까 수업으로  
4명의 user가 있어야한다. 확인해보자.

1. 본인 계정
  - 우리 교안에서 inho 라고 써있는 것은  
전부 본인 계정명으로 바꾸어서 생각하자.
2. jdragon
3. onnew
4. tigerstar

```
inho@inho:~$  
inho@inho:~$  
inho@inho:~$ cd /home/  
inho@inho:/home$ ls  
inho jdragon onnew tigerstar  
inho@inho:/home$  
inho@inho:/home$
```

## 암기해야할 규칙 1

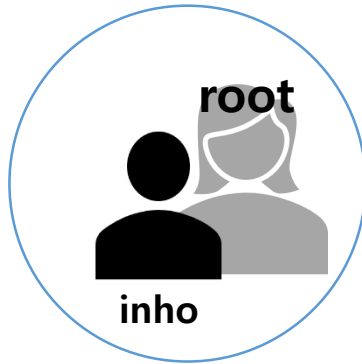
- 유저 계정을 하나 만들면,  
자동으로 그룹도 하나 만들어진다.
- 예시) inho 계정 adduser 시,  
**inho 라는 그룹도 만들어진다.**

User를 생성하면 그룹이 함께 생성됨

jdragon group



inho group



onnew group



tigerstar group



## 특정 user가 소속된 그룹 확인하기

- groupS [계정명]
- groups jdragon 라고 입력해보자.

```
inho@inho:/home$ groups jdragon
jdragon : jdragon
inho@inho:/home$
```

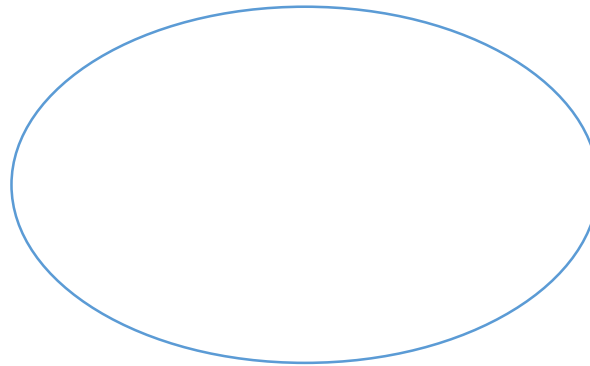
jdragon user

소속된 그룹 이름

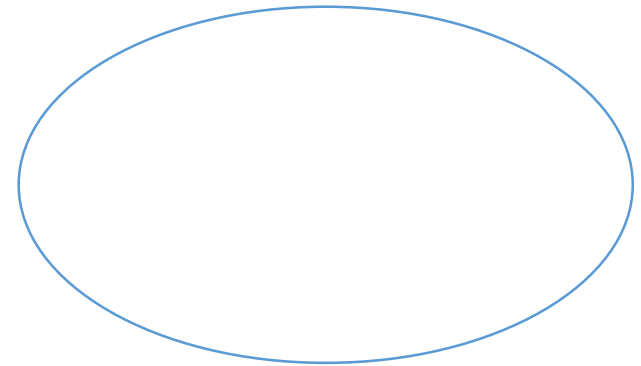
## 그룹 생성하기 명령어

- **addgroup** 명령어
  - 동사 먼저 나오고 명사가 나와야한다!!!!
- sudo **addgroup** embedded
- sudo **addgroup** web

embedded group



web group

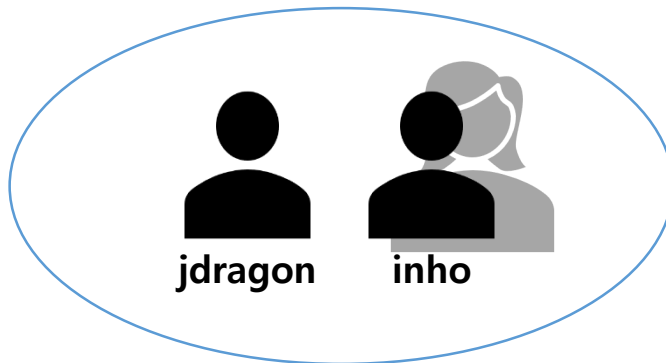


아래 그림과 같이 세팅하자.

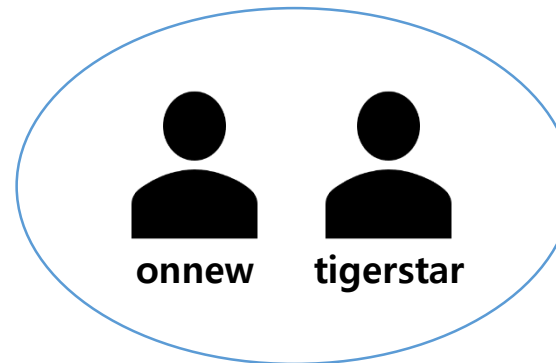
만든 그룹에, 사람들을 추가해보자.

- 그룹에 사용자 추가 : `sudo gpasswd -a inho embedded` (-a 옵션 : add)
- 그룹에 사용자 제거 : `sudo gpasswd -d inho embedded` (-d 옵션 : delete)

embedded Group



web Group



/etc/group 파일을 cat 으로 읽어보기

• 내용을 분석해볼것.

inho라는  
계정을 만들 때  
생성된 그룹

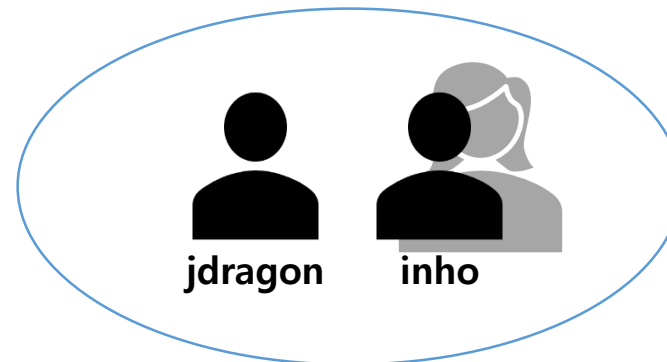
jdragon 이라는  
계정을 만들 때  
생성된 그룹

web 그룹,  
tigerstar과 onnew가 포함됨

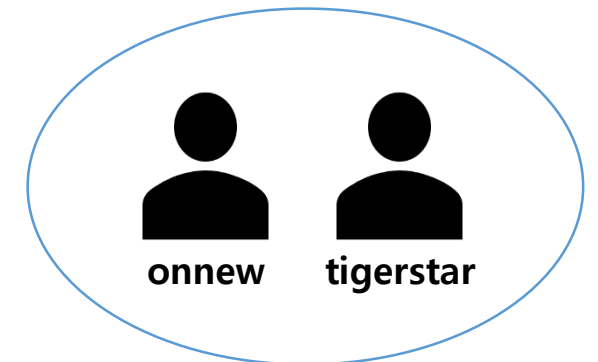
```
gdm:x:125:  
inho:x:1000:  
sambashare:x:126:inho  
vboxsf:x:999:  
rdma:x:127:  
jdragon:x:1001:  
onnew:x:1002:  
tigerstar:x:1003:  
embedded:x:1004:jdragon,inho  
web:x:1005:tigerstar,onnew  
inho@inho: /home$
```

onnew 가 소속된 그룹은 2개이다. O, X

embedded Group



web Group





답은 O, onnew는 2개 그룹에 소속되어 있다.

1. onnew 계정을 생성할 때, onnew 그룹 안에 onnew가 소속되어진다.
2. 그리고 onnew user는 web 그룹에 소속되어있다.

User를 생성하면 그룹이 함께 생성됨

jdragon group



inho group



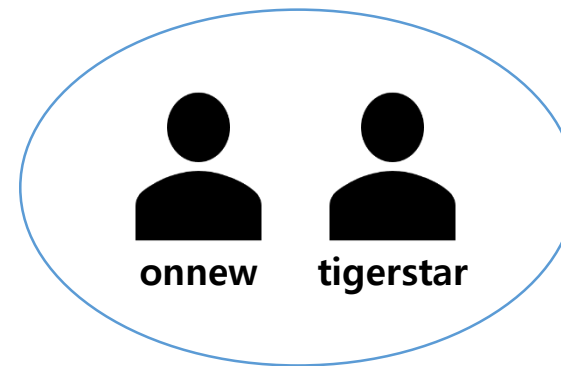
onnew group



tigerstar group



web Group

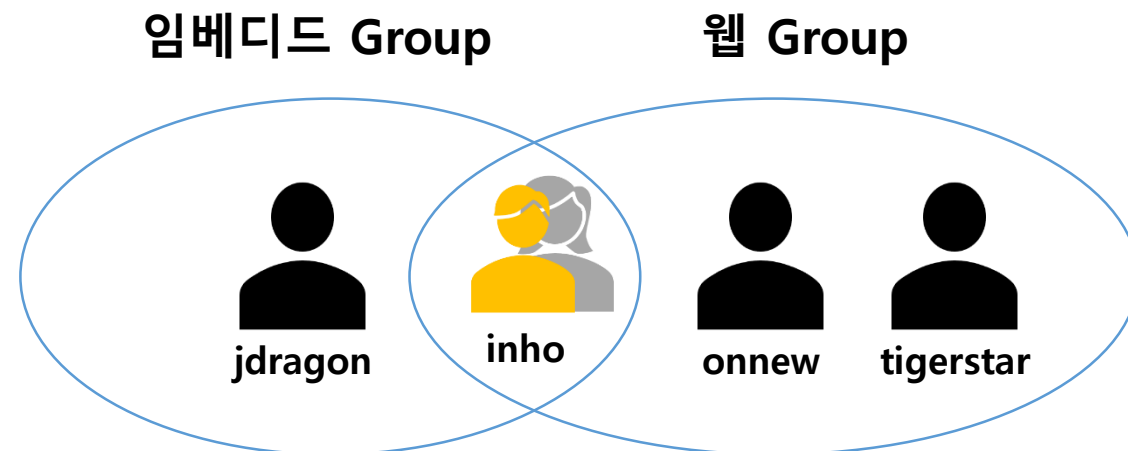


한 user는 여러개의 그룹에 참여 가능하다.

- 본계정(inho)을 두 그룹에 모두 포함시킨다.
  - `sudo gpasswd -a inho embedded`
  - `sudo gpasswd -a inho web`

결과를 확인해보자.

- `cat /etc/group`
- `groups inho`



그룹마다 어떤 사람들이 소속되어 있는지?

- `cat /etc/group`

user가 어떤 그룹에 소속되었는지?

- `groups [user계정명]`

```
inho@inho:/home$ groups onnew
onnew : onnew embedded web
inho@inho:/home$
```

리눅스에서는 Group 단위로, 권한을 단체 설정하려고

- 특정 Group만 읽고 쓸 수 있는 파일 설정 가능
- 특정 Group만 쓸 수 있는 디렉토리 설정 가능
- 특정 Group만 실행 시킬 수 있는 프로그램 설정 가능

실제 회사에서도 이런 방식으로 부서별 그룹을 지정한다.

- 회사 부서별 (그룹별) 파일 접속,  
디렉토리 접속 권한을 달리 부여함
- 타 부서 파일을 읽지 못하도록  
그룹 별 권한 설정 가능

delgroup [그룹명]

- 동사 이후 그룹명을 적으면 된다.

web 그룹만 지우고, 결과를 확인하자

- 결과 확인 : `cat /etc/group`

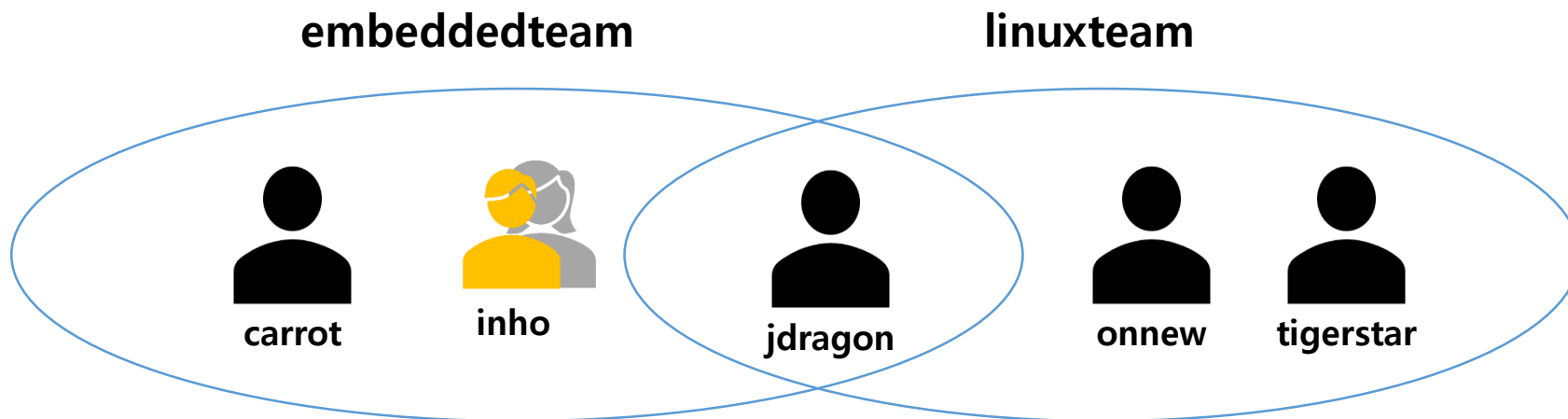
embedded 그룹 지우고, 결과를 확인하자

사용자를 지우자 - 깔끔하게 지운다.

- jdragon / onnew / tigerstar
- 옵션은 man을 참조하자.

다음 그림과 같이, 처음부터 다시 세팅하자

- history 명령어를 써서, 기존 이력을 참조해서 세팅하자.



## 1. 그룹 만들기 [완료]

- 그룹을 만든다.
- 만든 그룹 안에, user 들을 포함시킨다.

## 2. 개인 / 그룹 단위, 권한 세팅

- 문서 파일 / 실행 파일 / 디렉토리에 대한 권한을 세세히 설정해본다.



바로 시작할 내용

## 권한 세팅을 위한 사전지식 파일의 종류



## 이번 챕터의 목적

권한 설정을 공부 하기 전,  
한가지 암기 해야할 지식이 있다.

→ 리눅스 파일의 종류

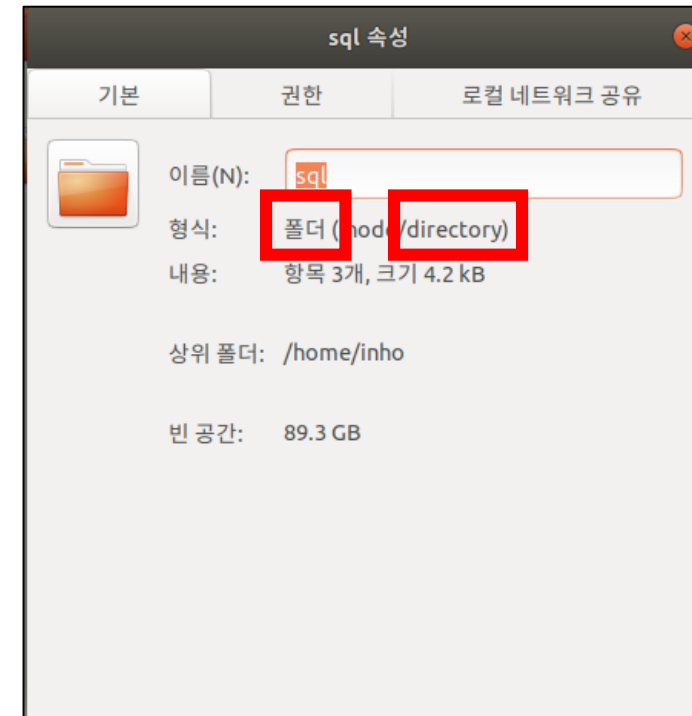
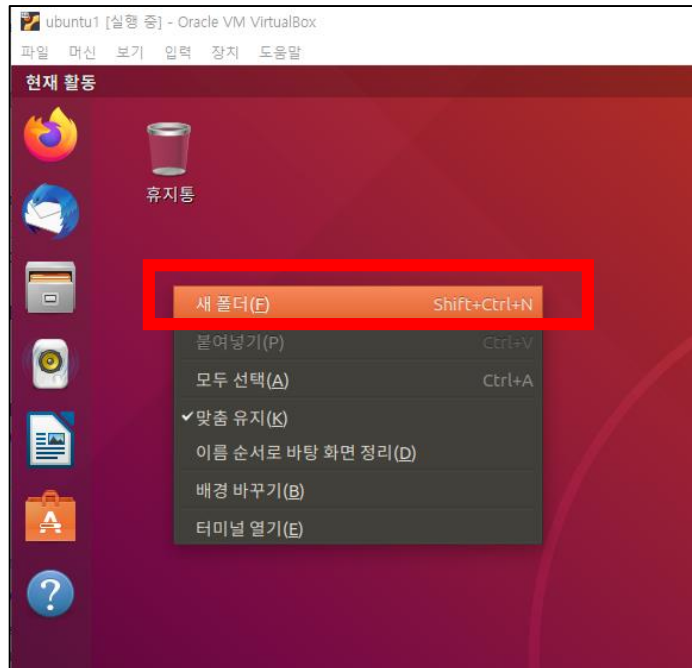
리눅스에서는  
폴더를 “**디렉토리**” 라고 불러야만 한다!!  
• O, X !?

# 답은 NO !!!

# Confidential

X 이다.

폴더 / 디렉토리 편한대로 부르자.



리눅스 커널 내부에서는  
폴더도 파일로 취급한다.?

- O / X

답은 O

특이한 리눅스 시스템,  
폴더도, 일반 파일도 모두 파일로 취급한다.

- 폴더
  - 리눅스 커널 내부에서는, 파일과 폴더를 똑같이 취급

## Regular File

- 일반 파일

## Directory File

- 리눅스 커널 내부에서는, 디렉토리도 파일로 취급

## Link File

- 윈도우의 **바로가기**와 비슷한 파일

## Device File

- 새로운 장치 (마우스 등) 연결하면, **상징물**과 같은 파일이 하나 생긴다.
- 이 파일 건드리면, 장치를 제어할 수 있다.

권한을 공부하기 전,  
암기하셔야 하는 내용

파일의 네 가지 종류!!



파일의 종류 네 가지를 말해보자.



권한은 ls -al 로 확인할 수 있다.

- /etc 디렉토리 내부를 ls -al 로 읽어보자.
- 이제 상세 내용을 배울 예정. 이 부분이 파일에 대한 권한이 나와있는 부분이다.

```
crwxr-xr-x 3 root root 4096 2월 10 03:52 update-manager
crwxr-xr-x 2 root root 4096 2월 10 03:52 update-motd.d
crwxr-xr-x 2 root root 4096 1월 21 01:46 update-notifier
-rw-r--r-- 1 root root 1523 2월 10 2020 usb_modeswitch.conf
crwxr-xr-x 2 root root 4096 2월 25 2020 usb_modeswitch.d
crwxr-xr-x 2 root root 4096 2월 10 03:47 vim
crwxr-xr-x 4 root root 4096 3월 1 13:30 vmware-tools
lrwxrwxrwx 1 root root 23 3월 1 13:12 vtrgb -> /etc/alternatives/vtrgb
crwxr-xr-x 5 root root 4096 2월 10 03:49 vulkan
-rw-r--r-- 1 root root 4942 7월 25 2019 wgetrc
crwxr-xr-x 2 root root 4096 3월 1 13:44 wpa_supplicant
-rw-r--r-- 1 root root 642 9월 24 2019 xattr.conf
crwxr-xr-x 6 root root 4096 2월 10 03:49 xdg
crwxr-xr-x 2 root root 4096 2월 10 03:52 xml
-rw-r--r-- 1 root root 477 10월 8 2019 zsh_command_not_found
inho@inho-VirtualBox:/etc$
```



## 파일의 종류를 나타낸다.

- - : 레귤러
- d : 디렉토리
- l : link - 윈도우의 바로가기 파일
- c, b : 장치파일

```
drwxr-xr-x 3 root root 4096 2월 10 03:52 update-manager
drwxr-xr-x 2 root root 4096 2월 10 03:52 update-motd.d
drwxr-xr-x 2 root root 4096 1월 21 01:46 update-notifier
-rw-r--r-- 1 root root 1523 2월 10 2020 usb_modeswitch.conf
drwxr-xr-x 2 root root 4096 2월 25 2020 usb_modeswitch.d
drwxr-xr-x 2 root root 4096 2월 10 03:47 vim
drwxr-xr-x 4 root root 4096 3월 1 13:30 vmware-tools
lrwxrwxrwx 1 root root 23 3월 1 13:12 vtrgb -> /etc/alternatives/vtrgb
drwxr-xr-x 5 root root 4096 2월 10 03:49 vulkan
-rw-r--r-- 1 root root 4942 7월 25 2019 wgetrc
drwxr-xr-x 2 root root 4096 3월 1 13:44 wpa_supplicant
-rw-r--r-- 1 root root 642 9월 24 2019 xattr.conf
drwxr-xr-x 6 root root 4096 2월 10 03:49 xdg
drwxr-xr-x 2 root root 4096 2월 10 03:52 xml
-rw-r--r-- 1 root root 477 10월 8 2019 zsh_command_not_found
incho@incho-VirtualBox: /etc$
```

ls -al /dev

- c와 b를 확인하자.

```
inho@inho: ~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
inho@inho:~$ ls /dev -al  
합계 4  
drwxr-xr-x 19 root root      4300 2월 17 00:25 .  
drwxr-xr-x 24 root root      4096 2월 17 00:25 ..  
crw-r--r--  1 root root    10, 235 2월 17 00:10 autofs  
drwxr-xr-x  2 root root      540 2월 17 00:10 block  
drwxr-xr-x  2 root root       80 2월 17 00:10 bsg  
crw-rw----  1 root disk   10, 234 2월 17 00:25 btrfs-control  
drwxr-xr-x  3 root root       60 2월 17 00:10 bus  
lrwxrwxrwx  1 root root       3 2월 17 00:10 cdrom -> sr0  
drwxr-xr-x  2 root root     3720 2월 17 00:25 char  
crw-----  1 root root       5,  1 2월 17 00:10 console  
lrwxrwxrwx  1 root root      11 2월 17 00:10 core -> /proc/kcore  
drwxr-xr-x  6 root root      120 2월 17 00:25 cpu  
crw-----  1 root root    10,  59 2월 17 00:10 cpu_dma_latency  
crw-----  1 root root    10, 203 2월 17 00:10 cuse  
drwxr-xr-x  6 root root      120 2월 17 00:10 disk  
drwxr-xr-x  3 root root       80 2월 17 00:10 dri  
lrwxrwxrwx  1 root root       3 2월 17 00:10 dvd -> sr0  
crw-----  1 root root    10,  62 2월 17 00:10 ecryptfs  
crw-rw----  1 root video  29,   0 2월 17 00:10 fb0  
lrwxrwxrwx  1 root root      13 2월 17 00:10 fd -> /proc/self/fd/
```

Device 파일이 모여 사는 곳 : /dev



# Is -ai 권한 표시

이번 챕터에서  
ls -al 가 표현하는

권한 관련 정보에 대해 **암기하는 시간**을 갖는다.

- ls -al /etc 를 수행하자.

```
drwxr-xr-x 3 root root 4096 2월 10 03:52 update-manager
drwxr-xr-x 2 root root 4096 2월 10 03:52 update-motd.d
drwxr-xr-x 2 root root 4096 1월 21 01:46 update-notifier
-rw-r--r-- 1 root root 1523 2월 10 2020 usb_modeswitch.conf
drwxr-xr-x 2 root root 4096 2월 25 2020 usb_modeswitch.d
drwxr-xr-x 2 root root 4096 2월 10 03:47 vim
drwxr-xr-x 4 root root 4096 3월 1 13:30 vmware-tools
lrwxrwxrwx 1 root root 23 3월 1 13:12 vtrgb -> /etc/alternatives/vtrgb
drwxr-xr-x 5 root root 4096 2월 10 03:49 vulkan
-rw-r--r-- 1 root root 4942 7월 25 2019 wgetrc
drwxr-xr-x 2 root root 4096 3월 1 13:44 wpa_supplicant
-rw-r--r-- 1 root root 642 9월 24 2019 xattr.conf
drwxr-xr-x 6 root root 4096 2월 10 03:49 xdg
drwxr-xr-x 2 root root 4096 2월 10 03:52 xml
-rw-r--r-- 1 root root 477 10월 8 2019 zsh_command_not_found
inho@inho-VirtualBox:/etc$
```

## 파일의 종류

```
d-rwxr-xr-x  3 root root    4096 2월 10 03:52 update-manager
d-rwxr-xr-x  2 root root    4096 2월 10 03:52 update-motd.d
d-rwxr-xr-x  2 root root    4096 1월 21 01:46 update-notifier
-rw-r--r--   1 root root    1523 2월 10 2020 usb_modeswitch.conf
d-rwxr-xr-x  2 root root    4096 2월 25 2020 usb_modeswitch.d
d-rwxr-xr-x  2 root root    4096 2월 10 03:47 vim
d-rwxr-xr-x  4 root root    4096 3월  1 13:30 vmware-tools
lrwxrwxrwx   1 root root      23 3월  1 13:12 vtrgb -> /etc/alternatives/vtrgb
d-rwxr-xr-x  5 root root    4096 2월 10 03:49 vulkan
-rw-r--r--   1 root root   4942 7월 25 2019 wgetrc
d-rwxr-xr-x  2 root root    4096 3월  1 13:44 wpa_supplicant
-rw-r--r--   1 root root    642 9월 24 2019 xattr.conf
d-rwxr-xr-x  6 root root    4096 2월 10 03:49 xdg
d-rwxr-xr-x  2 root root    4096 2월 10 03:52 xml
-rw-r--r--   1 root root    477 10월  8 2019 zsh_command_not_found
inho@inho-VirtualBox:/etc$
```

r,w,x

- r : 파일 read 권한
- w : 파일 write 권한
- x : 파일 실행(execution) 권한

암기하자

```
drwxr-xr-x 3 root root 4096 2월 10 03:52 update-manager
drwxr-xr-x 2 root root 4096 2월 10 03:52 update-motd.d
drwxr-xr-x 2 root root 4096 1월 21 01:46 update-notifier
-rw-r--r-- 1 root root 1523 2월 10 2020 usb_modeswitch.conf
drwxr-xr-x 2 root root 4096 2월 25 2020 usb_modeswitch.d
drwxr-xr-x 2 root root 4096 2월 10 03:47 vim
drwxr-xr-x 4 root root 4096 3월 1 13:30 vmware-tools
lrwxrwxrwx 1 root root 23 3월 1 13:12 vtrgb -> /etc/alternatives/vtrgb
drwxr-xr-x 5 root root 4096 2월 10 03:49 vulkan
-rw-r--r-- 1 root root 4942 7월 25 2019 wgetrc
drwxr-xr-x 2 root root 4096 3월 1 13:44 wpa_supplicant
-rw-r--r-- 1 root root 642 9월 24 2019 xattr.conf
drwxr-xr-x 6 root root 4096 2월 10 03:49 xdg
drwxr-xr-x 2 root root 4096 2월 10 03:52 xml
-rw-r--r-- 1 root root 477 10월 8 2019 zsh_command_not_found
inho@inho-VirtualBox:/etc$
```

## 파일의 User와 Group

- 한 파일을 누가 사용하고 어느 그룹이 사용하는지가 명시됨

## 우리 수업에서는

- Owner를 오너 라고 부르겠음
- Group Ownership을 오너 그룹 이라고 부르겠음

O.Group (Owner Group)

```
drwxr-xr-x 3 root root 4096 2월 10 03:52 update-manager
drwxr-xr-x 2 root root 4096 2월 10 03:52 update-motd.d
drwxr-xr-x 2 root root 4096 1월 21 01:46 update-notifier
-rw-r--r-- 1 root root 1523 2월 10 2020 usb_modeswitch.conf
drwxr-xr-x 2 root root 4096 2월 25 2020 usb_modeswitch.d
drwxr-xr-x 2 root root 4096 2월 10 03:47 vim
drwxr-xr-x 4 root root 4096 3월 1 13:30 vmware-tools
lrwxrwxrwx 1 root root 23 3월 1 13:12 vtrgb -> /etc/alternatives/vtrgb
drwxr-xr-x 5 root root 4096 2월 10 03:49 vulkan
-rw-r--r-- 1 root root 4942 7월 25 2019 wgetrc
drwxr-xr-x 2 root root 4096 3월 1 13:44 wpa_supplicant
-rw-r--r-- 1 root root 642 9월 24 2019 xattr.conf
drwxr-xr-x 6 root root 4096 2월 10 03:49 xdg
drwxr-xr-x 2 root root 4096 2월 10 03:52 xml
-rw-r--r-- 1 root root 477 10월 8 2019 zsh_command_not_found
```

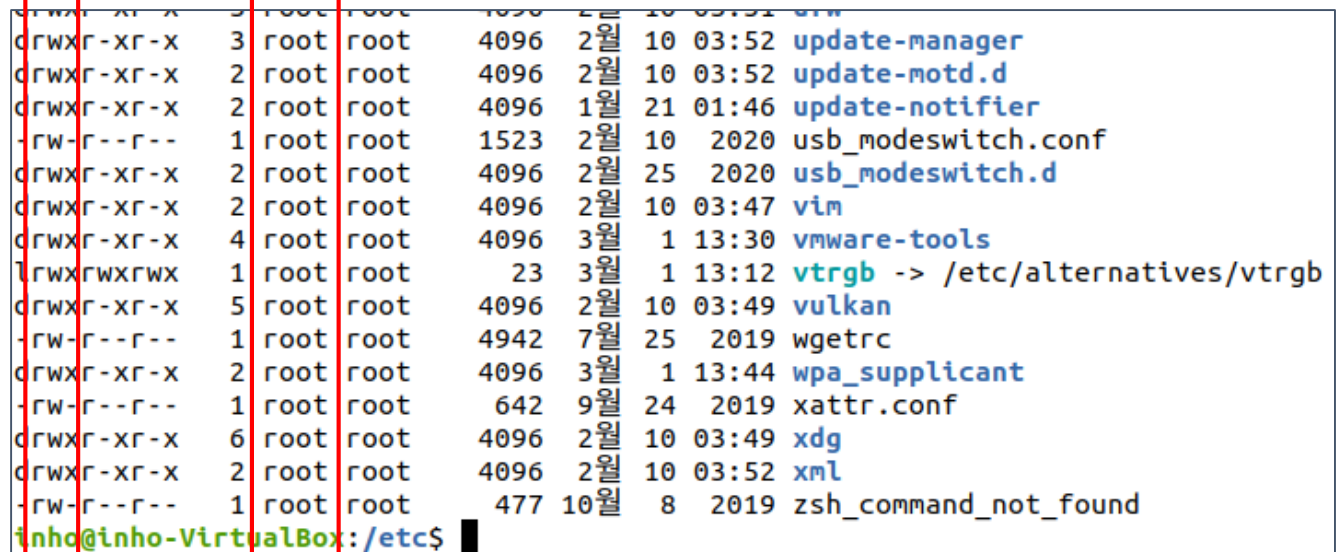
inho@inho-VirtualBox:/etc\$

Owner



## Owner 의 r, w, x 권한

- Owner 정보가 나와있는 위치를 암기하자



The screenshot shows the output of the command `ls -al` in a terminal. Two red boxes are drawn around the first two columns of the output: the permissions column (e.g., `-rwxr-xr-x`) and the owner column (e.g., `root`). Red arrows point from the text 'Owner 정보가 나와있는 위치를 암기하자' to these two boxes. The terminal output is as follows:

Permissions	Link Count	Owner	Group	Size	Month	Day	Time	File Name
-rwxr-xr-x	3	root	root	4096	2월	10	03:52	update-manager
-rwxr-xr-x	2	root	root	4096	2월	10	03:52	update-motd.d
-rwxr-xr-x	2	root	root	4096	1월	21	01:46	update-notifier
-rw-r--r--	1	root	root	1523	2월	10	2020	usb_modeswitch.conf
-rwxr-xr-x	2	root	root	4096	2월	25	2020	usb_modeswitch.d
-rwxr-xr-x	2	root	root	4096	2월	10	03:47	vim
-rwxr-xr-x	4	root	root	4096	3월	1	13:30	vmware-tools
lrwxrwxrwx	1	root	root	23	3월	1	13:12	vtrgb -> /etc/alternatives/vtrgb
-rwxr-xr-x	5	root	root	4096	2월	10	03:49	vulkan
-rw-r--r--	1	root	root	4942	7월	25	2019	wgetrc
-rwxr-xr-x	2	root	root	4096	3월	1	13:44	wpa_supplicant
-rw-r--r--	1	root	root	642	9월	24	2019	xattr.conf
-rwxr-xr-x	6	root	root	4096	2월	10	03:49	xdg
-rwxr-xr-x	2	root	root	4096	2월	10	03:52	xml
-rw-r--r--	1	root	root	477	10월	8	2019	zsh_command_not_found

inho@inho-VirtualBox: /etc\$

Owner



## O.Group의 r, w, x 권한

- O.Group = Owner Group

```
drwxr-xr-x  3 root root    4096 2월 10 03:52 update-manager
drwxr-xr-x  2 root root    4096 2월 10 03:52 update-motd.d
drwxr-xr-x  2 root root    4096 1월 21 01:46 update-notifier
-rw-r--r--  1 root root   1523 2월 10  2020 usb_modeswitch.conf
drwxr-xr-x  2 root root    4096 2월 25  2020 usb_modeswitch.d
drwxr-xr-x  2 root root    4096 2월 10 03:47 vim
drwxr-xr-x  4 root root    4096 3월  1 13:30 vmware-tools
lrwxrwxrwx  1 root root     23 3월  1 13:12 vtrgb -> /etc/alternatives/vtrgb
drwxr-xr-x  5 root root    4096 2월 10 03:49 vulkan
-rw-r--r--  1 root root   4942 7월 25  2019 wgetrc
drwxr-xr-x  2 root root    4096 3월  1 13:44 wpa_supplicant
-rw-r--r--  1 root root    642 9월 24  2019 xattr.conf
drwxr-xr-x  6 root root    4096 2월 10 03:49 xdg
drwxr-xr-x  2 root root    4096 2월 10 03:52 xml
-rw-r--r--  1 root root    477 10월  8  2019 zsh_command_not_found
inho@incho-VirtualBox: /etc$
```

Owner Group

other는 guest가 아니다.

- Owner와 O.Group 제외한 사람들을 other라고 한다.

```
drwxr-xr-x 3 root root 4096 2월 10 03:52 update-manager
drwxr-xr-x 2 root root 4096 2월 10 03:52 update-motd.d
drwxr-xr-x 2 root root 4096 1월 21 01:46 update-notifier
-rw-r--r-- 1 root root 1523 2월 10 2020 usb_modeswitch.conf
drwxr-xr-x 2 root root 4096 2월 25 2020 usb_modeswitch.d
drwxr-xr-x 2 root root 4096 2월 10 03:47 vim
drwxr-xr-x 4 root root 4096 3월 1 13:30 vmware-tools
lrwxrwxrwx 1 root root 23 3월 1 13:12 vtrgb -> /etc/alternatives/vtrgb
drwxr-xr-x 5 root root 4096 2월 10 03:49 vulkan
-rw-r--r-- 1 root root 4942 7월 25 2019 wgetrc
drwxr-xr-x 2 root root 4096 3월 1 13:44 wpa_supplicant
-rw-r--r-- 1 root root 642 9월 24 2019 xattr.conf
drwxr-xr-x 6 root root 4096 2월 10 03:49 xdg
drwxr-xr-x 2 root root 4096 2월 10 03:52 xml
-rw-r--r-- 1 root root 477 10월 8 2019 zsh_command_not_found
inho@inho-VirtualBox:/etc$
```

user와 group을 제외

## 누구인가 Quiz

모든 파일에는 다음 항목이 명시된다.

- 파일의 Onwer
- 파일의 Owner Group

모든 파일에는 다음 권한이 명시된다.

- 파일의 Onwer 의 R,W,X 권한
- 파일의 Owner Group의 R,W,X 권한
- 파일 **Other(그외)**의 R,W,X 권한

위 내용을 이해했다면, “누구인가” 퀴즈를 시작한다.



Other는 의미를 적으시오.

```
drwxr-xr-x 3 root root 4096 2월 10 03:52 update-manager
drwxr-xr-x 2 root root 4096 2월 10 03:52 update-motd.d
drwxr-xr-x 2 root root 4096 1월 21 01:46 update-notifier
-rw-r--r-- 1 root root 1523 2월 10 2020 usb_modeswitch.conf
drwxr-xr-x 2 root root 4096 2월 25 2020 usb_modeswitch.d
drwxr-xr-x 2 root root 4096 2월 10 03:47 vim
drwxr-xr-x 4 root root 4096 3월 1 13:30 vmware-tools
lrwxrwxrwx 1 root root 23 3월 1 13:12 vtrgb -> /etc/alternatives/vtrgb
drwxr-xr-x 5 root root 4096 2월 10 03:49 vulkan
-rw-r--r-- 1 root root 4942 7월 25 2019 wgetrc
drwxr-xr-x 2 root root 4096 3월 1 13:44 wpa_supplicant
-rw-r--r-- 1 root root 642 9월 24 2019 xattr.conf
drwxr-xr-x 6 root root 4096 2월 10 03:49 xdg
drwxr-xr-x 2 root root 4096 2월 10 03:52 xml
-rw-r--r-- 1 root root 477 10월 8 2019 zsh_command_not_found
inho@inho-VirtualBox:/etc$
```

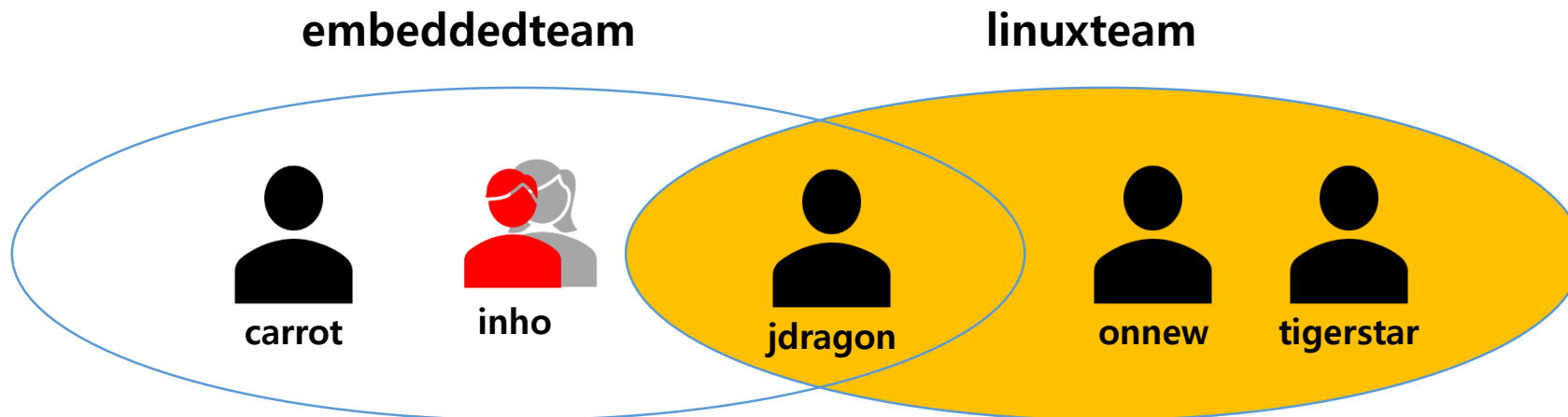
## Other

- owner도 아니고, 동시에 owner group도 아닌 사람들

예시.

- 파일 이름 : kfc.c
- 이 파일의 owner : inho
- 이 파일의 owner group : linux Team

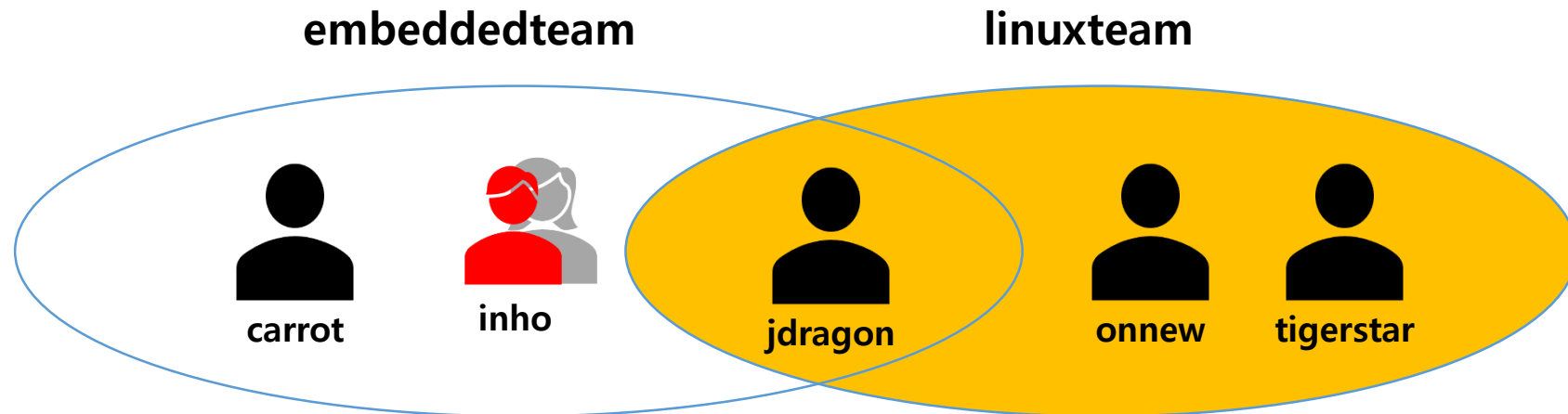
owner는, owner group 에 소속되지 않아도 된다.



여기서 Other 는 누구인가?

- 파일의 owner : **inho**
- 파일의 owner group : **linux Team**

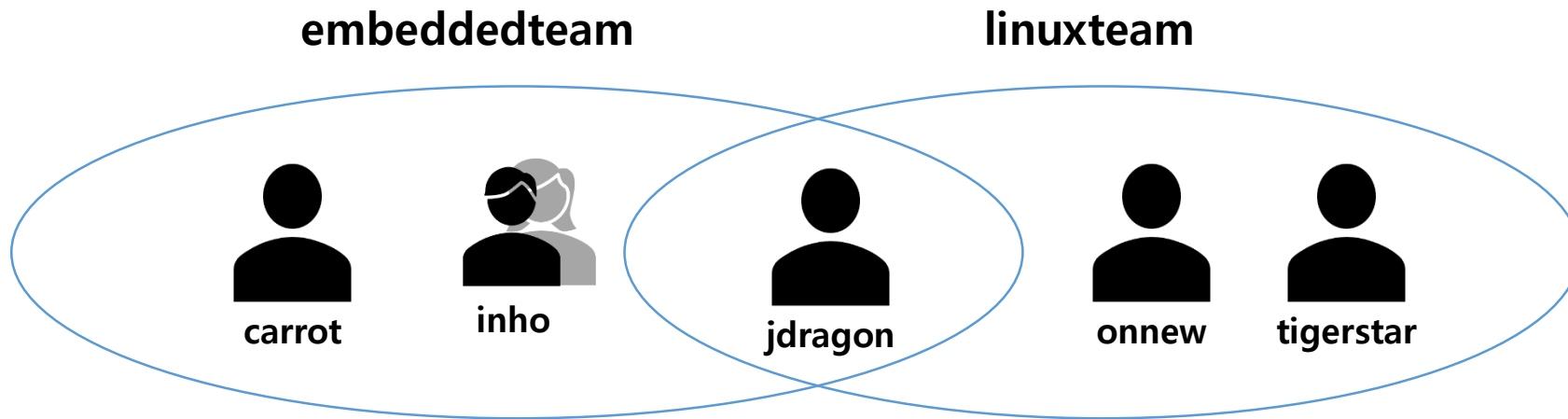
힌트 : other : owner / o.group이 아닌 모든 사람들





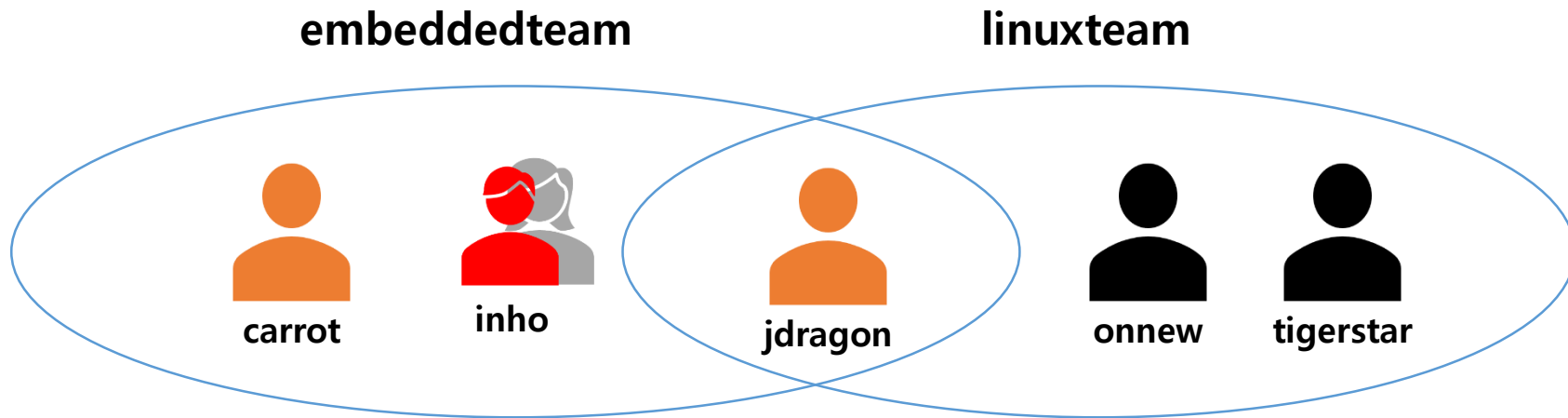
아래 내용에서, Other는 누구 누구인가?

- 파일명 : bbq.txt
- Owner : inho
- Group Owner : embeddedteam



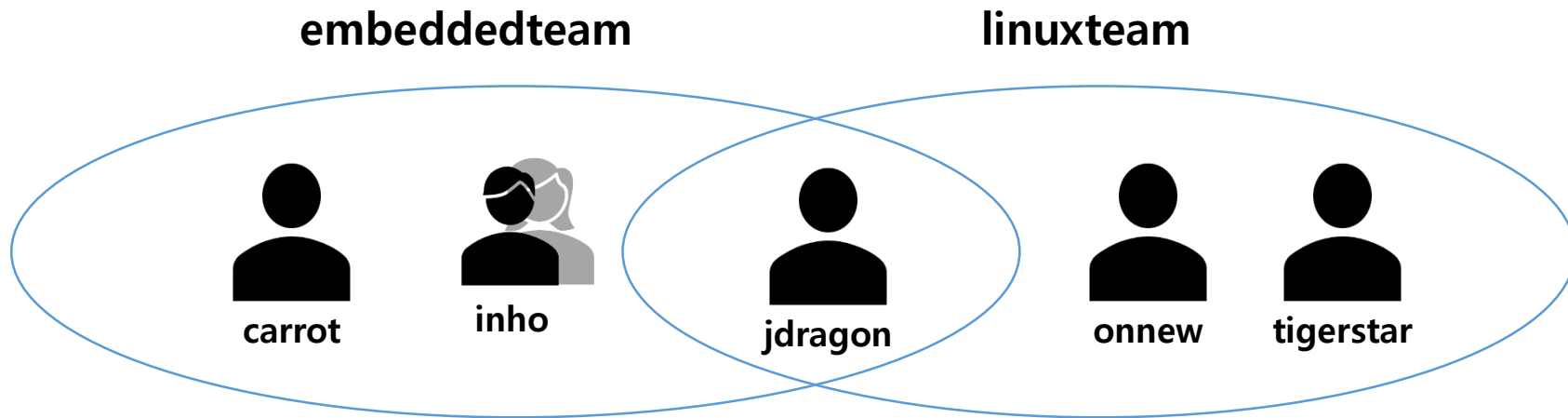
other는 onnew / tigerstar 이다.

- 파일명 : bbq.txt
- Owner : inho
- Group Owner : embeddedteam



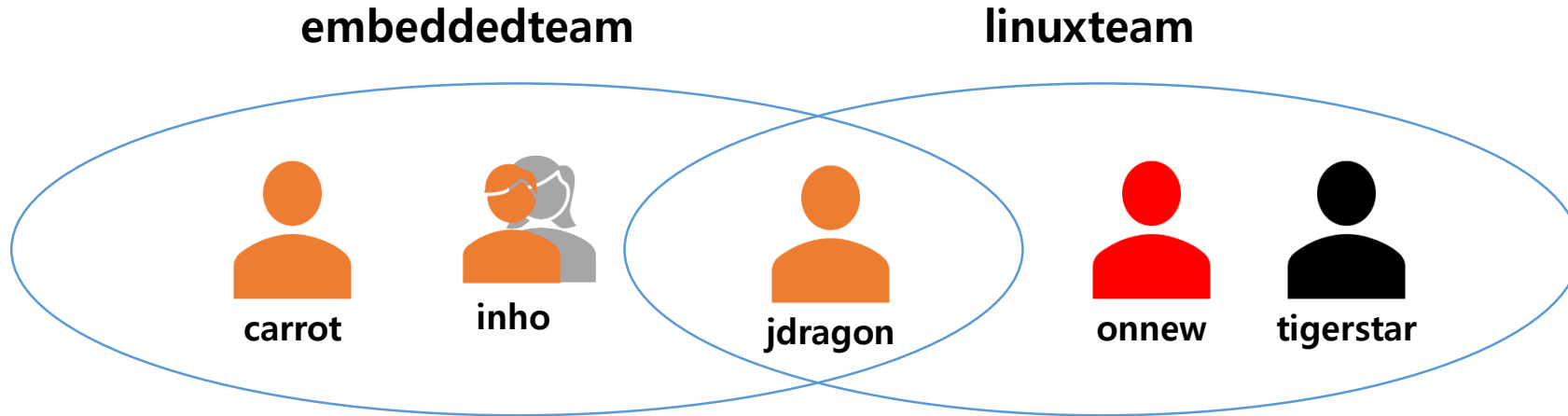
아래 내용에서, Other는 누구 누구인가?

- 파일명 : bbq.txt
- Owner : onnew
- Group Owner : embeddedteam



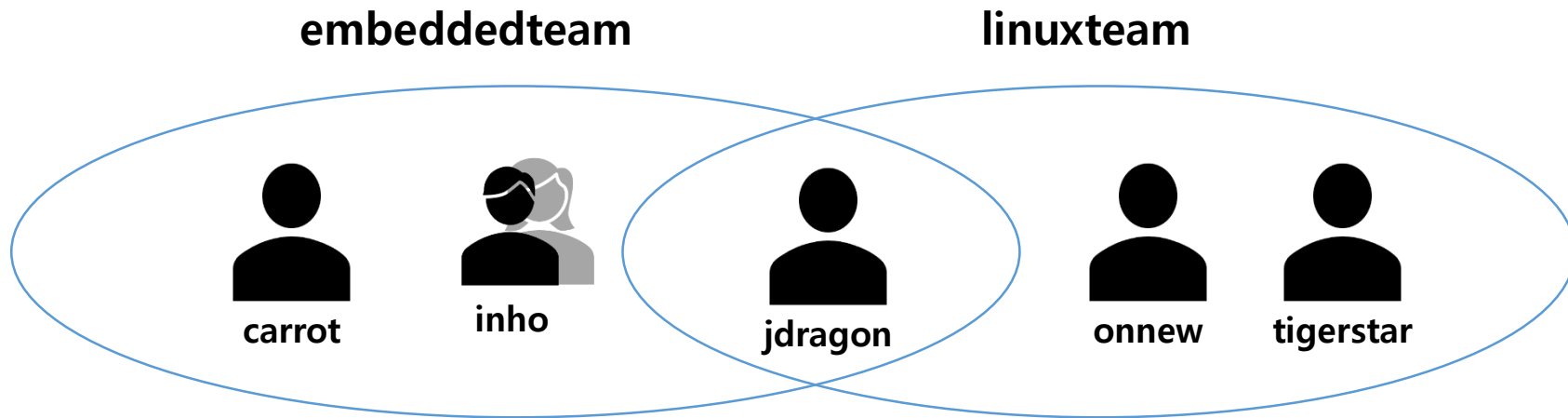
여기서 other는 tigerstar 이다.

- 파일명 : bbq.txt
- Owner : **onnew**
- Group Owner : **embeddedteam**



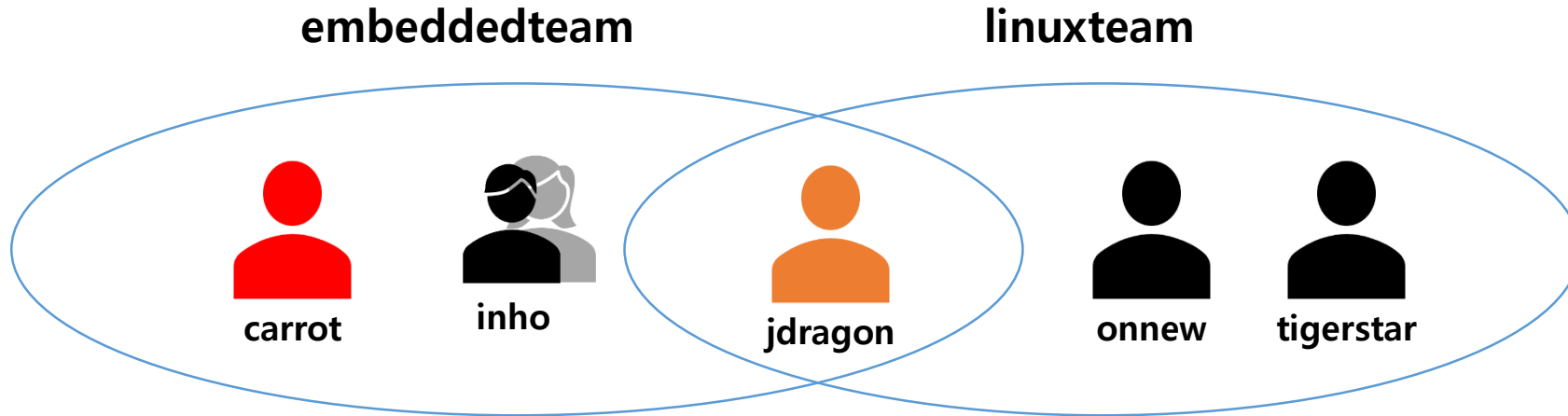
아래 내용에서, Other는 누구 누구인가?

- 파일명 : bbq.txt
- Owner : carrot
- Group Owner : jdragon



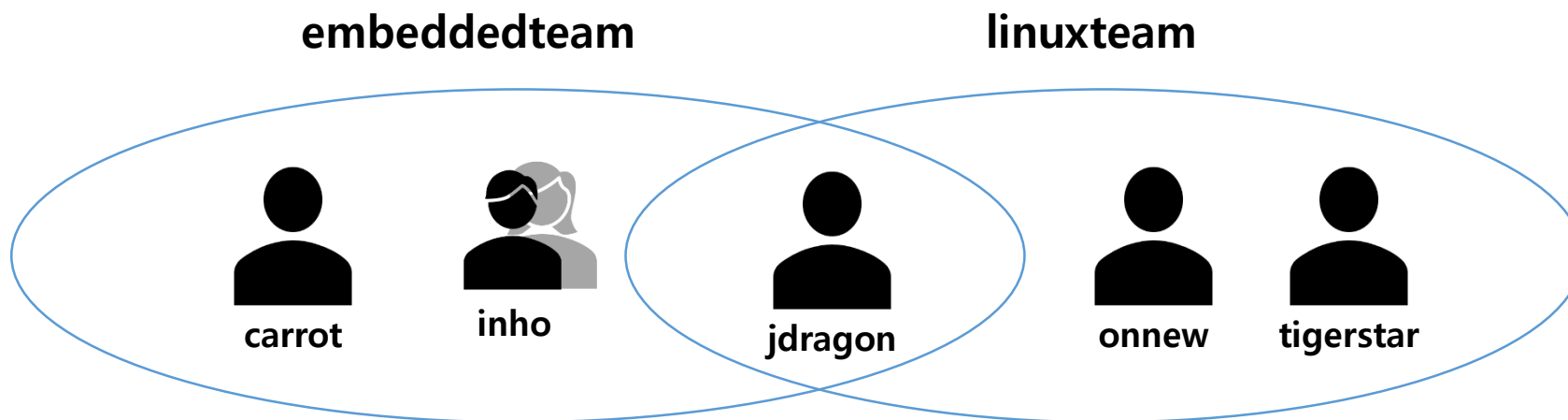
inho / onnew / tigerstar 가 Other이다.

- 파일명 : bbq.txt
- Owner : carrot
- Group Owner : jdragon



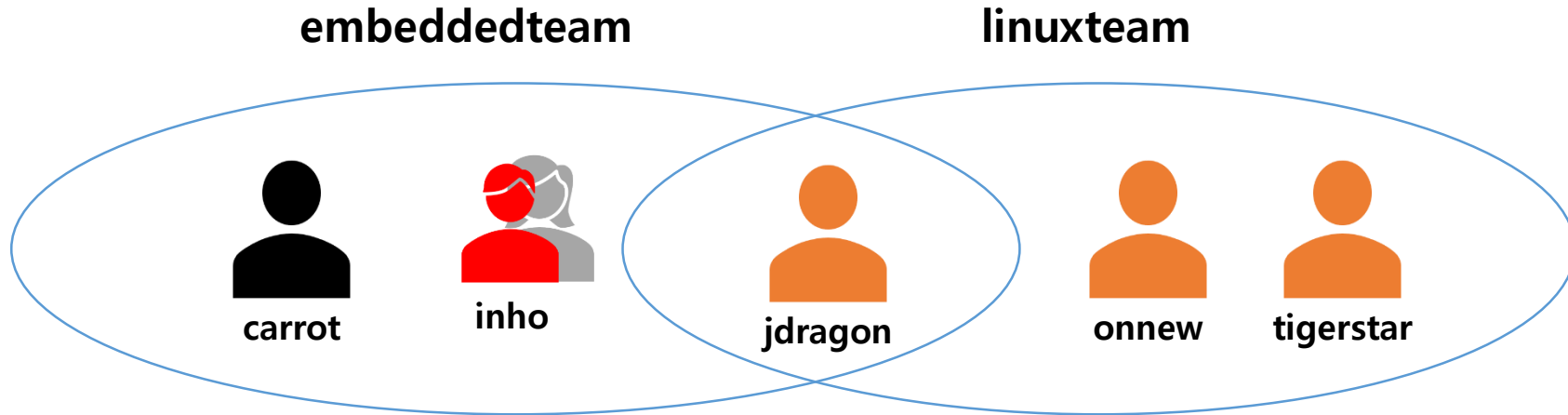
아래 내용에서, Read 권한이 있는 사람은 누구인가?

- 파일명 : bbq.txt
- 권한 : --- **r-X** r--
- Owner : inho
- Group Owner : linuxteam



## inho 빼고 전부

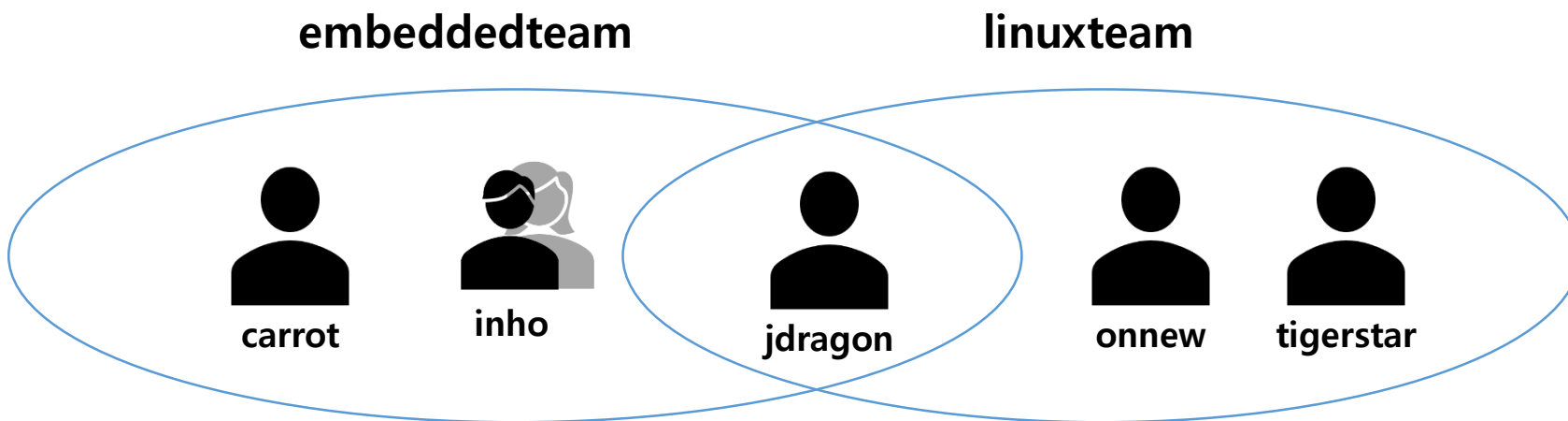
- 파일명 : `bbq.txt`
- 권한 : `--- r-X r--`
- Owner : `inho`
- Group Owner : `linuxteam`





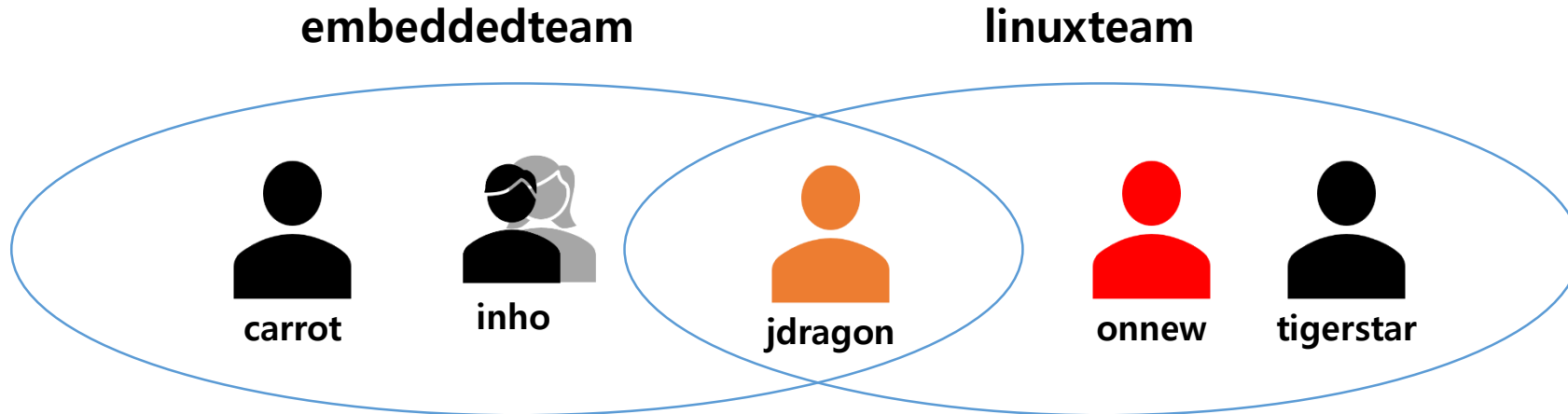
아래 내용에서, Write 권한이 있는 사람은 누구인가?

- 파일명 : bbq.txt
- 권한 : **-W- --- -W-**
- Owner : onnew
- Group Owner : jdragon



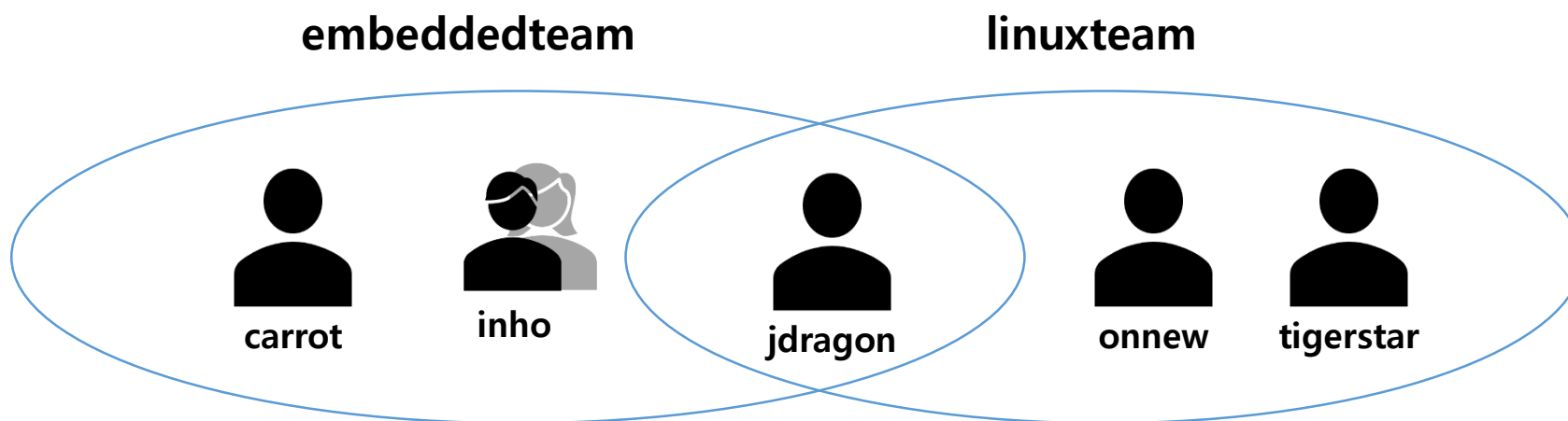
## jdragon 빼고 전부

- 파일명 : bbq.txt
- 권한 : **-W- --- -W-**
- Owner : **onnew**
- Group Owner : **jdragon**



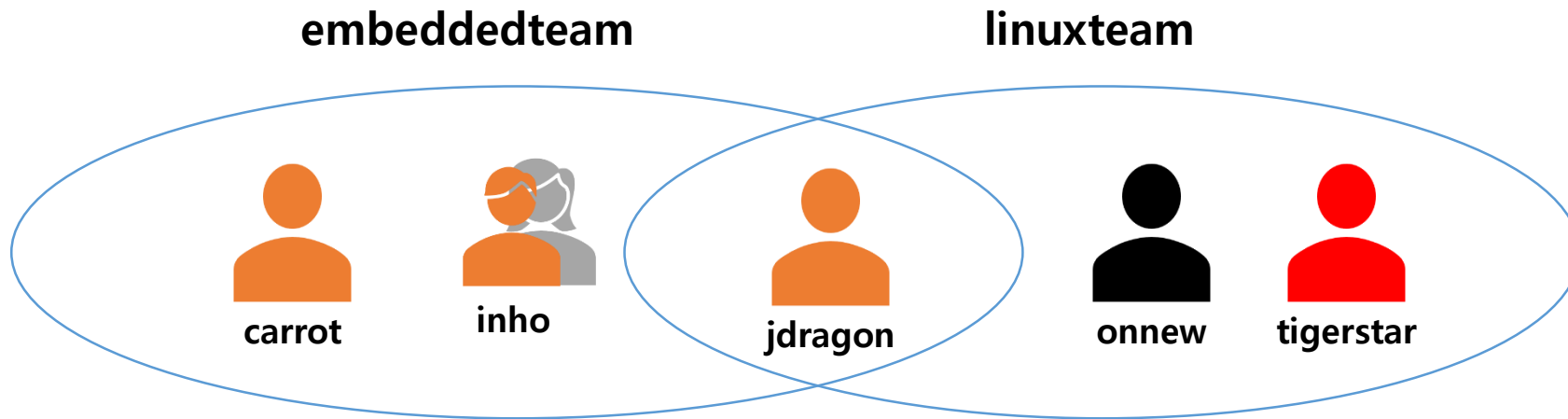
아래 내용에서, 실행 권한이 있는 사람은 누구인가?

- 파일명 : kfc.exe
- 권한 : **-WX --- rWX**
- Owner : tigerstar
- Group Owner : embeddedteam



## onnew / tigerstar

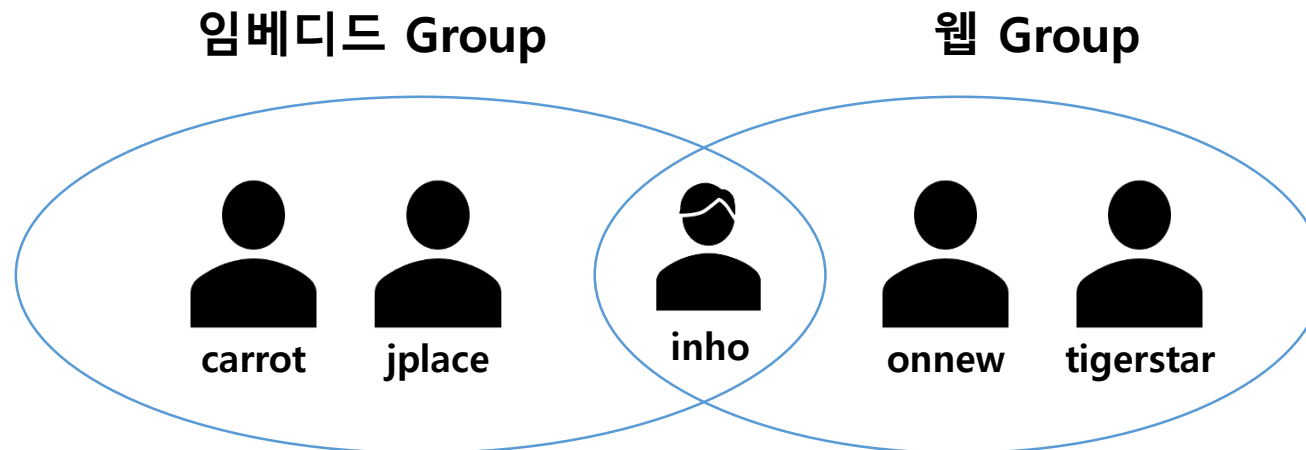
- 파일명 : kfc.exe
- 권한 : **-WX --- rWX**
- Owner : **tigerstar**
- Group Owner : **embeddedteam**



## Quiz. 권한 부여된 사람 수 맞추기 (root는 제외)

1. aaa 파일, read 권한 가진 user 수 ?
2. bbb 파일, write 권한 가진 user 수 ?
3. ccc 파일, 실행 권한 가진 user 수 ?
4. ddd 파일 read 권한 가진 user 수?

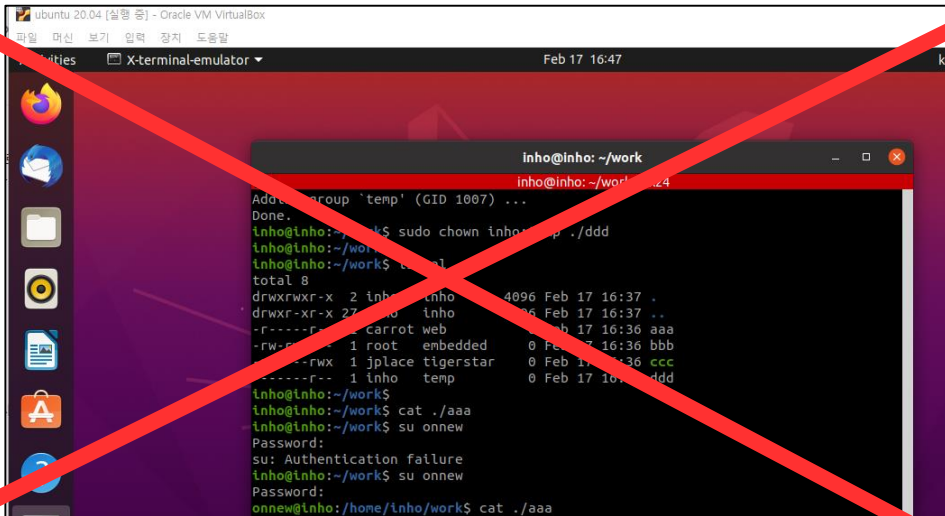
```
inho@inho-VirtualBox:~/work$ ls -al
합계 24
drwxrwxr-x  2 inho  inho    4096  3월  8 13:39 .
drwxr-xr-x 24 inho  inho    4096  3월  8 13:10 ..
-r-----r-- 1 carrot web      6  3월  8 13:37 aaa
-rw-rw---- 1 root  embedded  6  3월  8 13:37 bbb
-rwx---rwx 1 jplace tigerstar  6  3월  8 13:37 ccc
-----r-- 1 inho  temp      6  3월  8 13:40 ddd
```



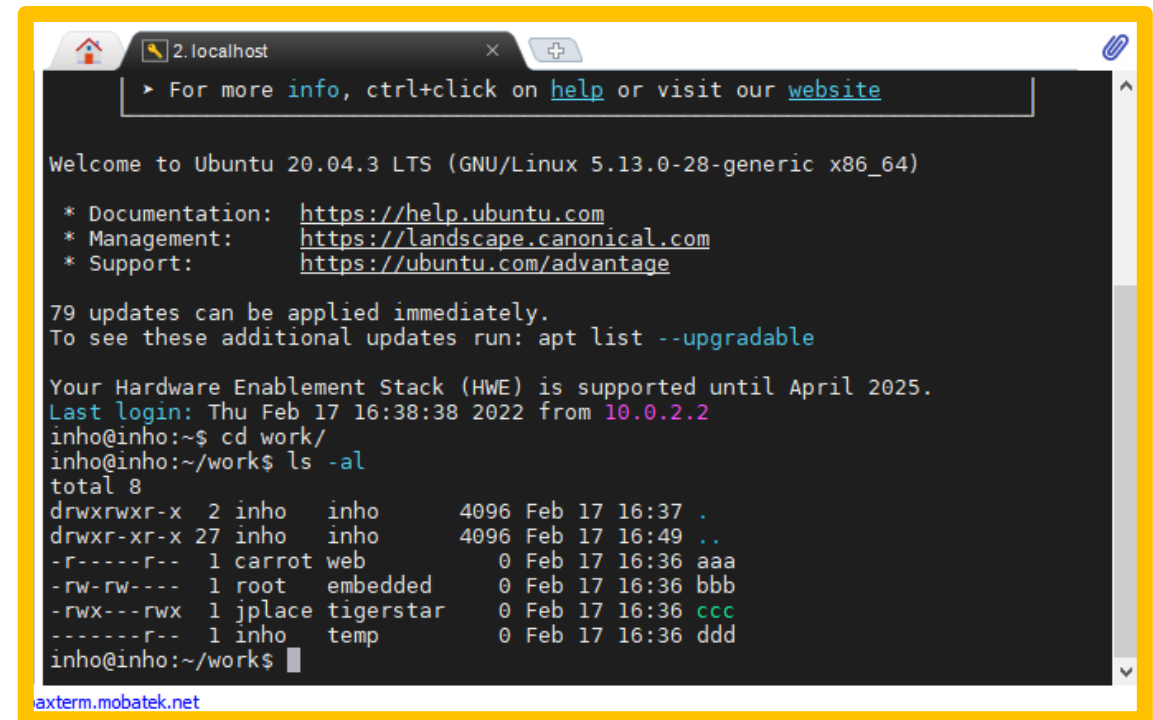
## 권한 설정 바꾸기

## Linux 원격접속하여 실습을 한다.

- Linux 직접 실행이 아닌,  
원격 터미널 연결로 권한 테스트를 하자.
- Root권한 사용자에게 권한이 더 부여되어,  
실습시 결과가 달라지는 문제가 발생



```
ubuntu 20.04 [실행 중] - Oracle VM VirtualBox
X-terminal-emulator Feb 17 16:47
incho@incho: ~/work
incho@incho:~/work$ sudo chown incho:temp ./ddd
incho@incho:~/work$ ls -la
total 8
drwxrwxr-x 2 incho incho 4096 Feb 17 16:37 .
drwxr-xr-x 27 incho incho 4096 Feb 17 16:37 ..
-r-----r-- 1 carrot web 0 Feb 17 16:36 aaa
-rw-rw---- 1 root embedded 0 Feb 17 16:36 bbb
-rwx---rwx 1 jplace tigerstar 0 Feb 17 16:36 ccc
-----r-- 1 incho temp 0 Feb 17 16:36 ddd
incho@incho:~/work$
incho@incho:~/work$ cat ./aaa
incho@incho:~/work$ su onnew
Password:
su: Authentication failure
incho@incho:~/work$ su onnew
Password:
onnew@incho:/home/incho/work$ cat ./aaa
```



```
2. localhost
> For more info, ctrl+click on help or visit our website

Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.13.0-28-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

79 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Thu Feb 17 16:38:38 2022 from 10.0.2.2
incho@incho:~$ cd work/
incho@incho:~/work$ ls -la
total 8
drwxrwxr-x 2 incho incho 4096 Feb 17 16:37 .
drwxr-xr-x 27 incho incho 4096 Feb 17 16:49 ..
-r-----r-- 1 carrot web 0 Feb 17 16:36 aaa
-rw-rw---- 1 root embedded 0 Feb 17 16:36 bbb
-rwx---rwx 1 jplace tigerstar 0 Feb 17 16:36 ccc
-----r-- 1 incho temp 0 Feb 17 16:36 ddd
incho@incho:~/work$
```

## 작업폴더 work 에서 진행

- abc.txt 파일을 하나 만든다.

```
inho@inho:~$ mkdir ~/work
inho@inho:~$ cd work/
inho@inho:~/work$ ls
inho@inho:~/work$
inho@inho:~/work$ touch abc.txt
inho@inho:~/work$ ls -al
합계 8
drwxrwxr-x  2 inho inho 4096  2월 17 12:40 .
drwxr-xr-x 19 inho inho 4096  2월 17 12:40 ..
-rw-rw-r--  1 inho inho   0  2월 17 12:40 abc.txt
inho@inho:~/work$
```



## Owner User와 Group 변경하기

- 명령어 : **chown** (Change Owner)
- **chwon User이름:Group이름 파일명**
  - 관리자 권한으로 실행

## abc.txt 파일 Owner 정보를 바꾸자

- Owner : jdragon
- Owner Group : embeddedteam

## Mode 변경

- user, group, other 들의 권한을 File Mode 라고 부른다.
- change mode 명령어

## sudo **chmod** [변경할 mode] 파일명

1. **user**를 **rw**x로 바꾸기
  - ex) sudo chmod **u=rwx** ./aaa
2. **group**을 **rw-** 로 바꾸기
  - ex) sudo chmod **g=rw** ./bbb
3. **other**를 **r-x** 로 바꾸기
  - ex) sudo chmod **o=r-x** ./ccc

하나씩 실험해봐야한다.

Linux 사용자마다 Mode를 바꾸는 명령어를 다양하게 쓴다.

→ 남들이 만든 매뉴얼들을 이해하기 위해, **모두 알고 있어야 한다.**

하나씩 실험해봐야한다.

```
chmod u+r ./aaa
• user에 r권한 추가

chmod o-w ./bbb
• other에 w권한 제외

chmod a+w ./aaa
• a : all
• u,g,o 모두 w권한 추가
```

2진수로 환산하여 사용

```
--x --x -rw  = 113
rwx rwx rwx  = 777
rw- rw- ---  = 660
```

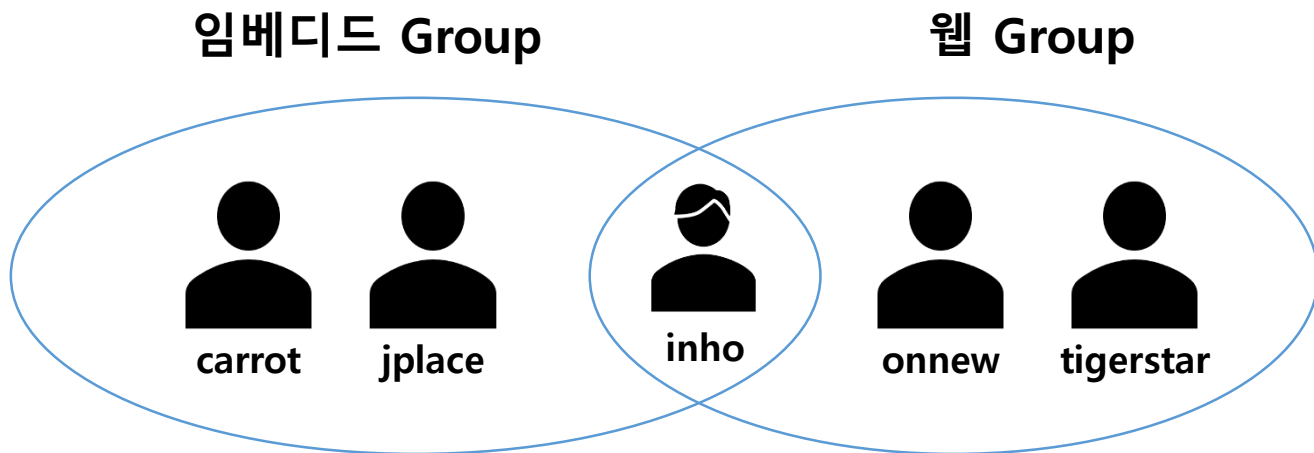
```
chmod 113 ./aaa
chmod 777 ./bbb
chmod 660 ./ccc
```

# [도전] 다음과 같이 세팅하자.

Confidential

1. 기존 사용자 지우기 (본인 계정 외)
2. User 만들기
3. Group 만들고, 세팅하기
4. 파일 만들기
5. Mode 변경 / Owner 변경

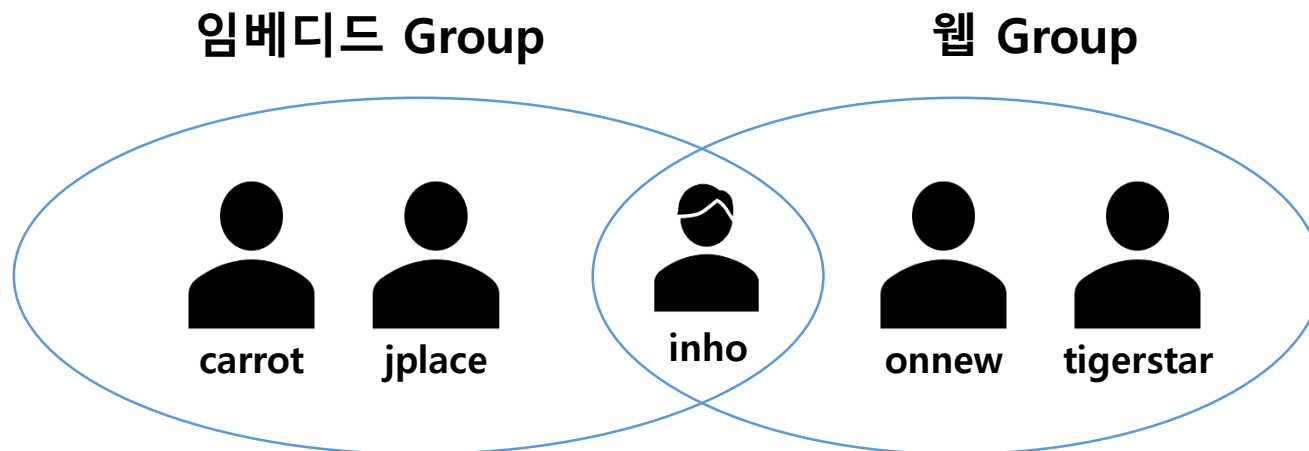
```
inho@inho-VirtualBox:~/work$ ls -al
합계 24
drwxrwxr-x  2 inho  inho    4096  3월  8 13:39 .
drwxr-xr-x 24 inho  inho    4096  3월  8 13:10 ..
-r-----r-- 1 carrot web      6  3월  8 13:37 aaa
-rw-rw---- 1 root  embedded  6  3월  8 13:37 bbb
-rwx---rwx 1 jplace tigerstar  6  3월  8 13:37 ccc
-----r-- 1 inho  temp      6  3월  8 13:40 ddd
```



접속해서 read Test를 수행한다.

1. aaa 파일이 읽혀야 하는 사람이 누군지 생각해보자.
2. 한 명씩 su로 접속해서 read를 수행한다.

```
inho@inho-VirtualBox:~/work$ ls -al
합계 24
drwxrwxr-x  2 inho  inho    4096  3월  8 13:39 .
drwxr-xr-x 24 inho  inho    4096  3월  8 13:10 ..
-r-----r-- 1 carrot web      6  3월  8 13:37 aaa
-rw-rw---- 1 root  embedded  6  3월  8 13:37 bbb
-rwx---rwx 1 jplace tigerstar  6  3월  8 13:37 ccc
-----r-- 1 inho  temp      6  3월  8 13:40 ddd
```



# 내일 방송에서 만나요!

삼성 청년 SW 아카데미