

[리눅스] 08. 권한(permission)

학습 목표

- 가. Linux의 권한 설정에 대해 알아본다.
- 나. chmod의 명령어를 통해 권한을 변경할 수 있다.
- 다. 그룹의 개념에 대해 이해할 수 있다.

목차

08. 권한(permission)과 그룹

8-1 권한이 대상이 되는 File & Directory(파일과 디렉터리)

8-2 예제를 통한 권한 확인(읽기, 쓰기 권한)

8-3 예제를 통한 권한 확인(실행권한)

8-4 그룹에 대해 알아보기

8-5 groupadd에 실행해보기

8-6 Summary

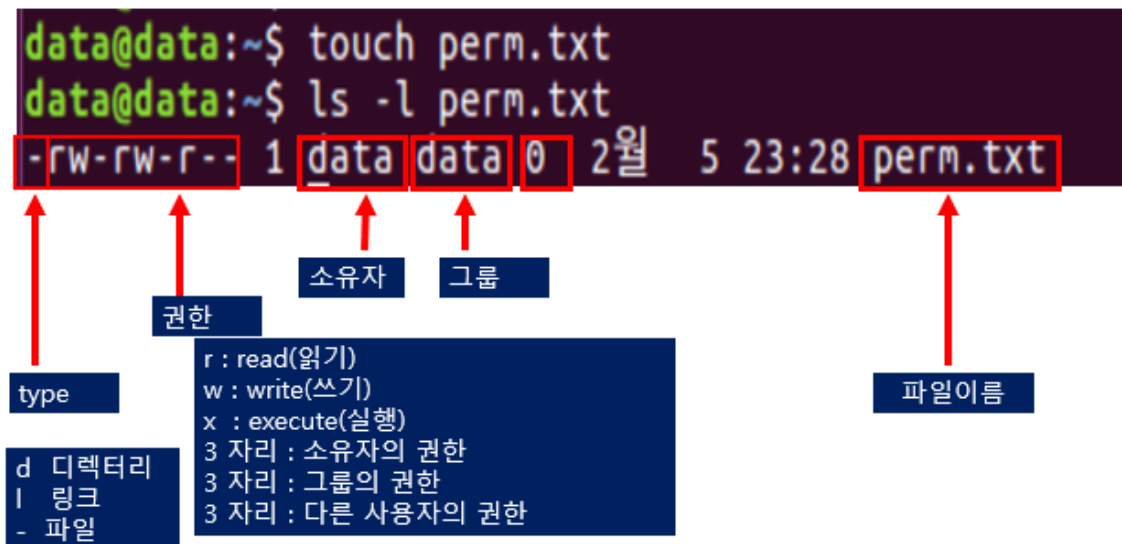
8-7 Quiz

8-1 권한이 대상이 되는 File & Directory(파일과 디렉터리)

파일과 디렉터리는 각각의 권한을 가지게 된다.

빈 파일을 만들고 perm.txt에 'hi'라는 출력을 넣어 출력시킨다.

```
$ touch perm.txt
$ ls -l perm.txt
$ echo 'hi' > perm.txt
$ cat perm.txt
```



8-2 예제를 통한 권한 확인

(가) 유저 추가 후, 파일 생성을 한다. 다른 유저에서 확인이 가능한가?

```
data~$ sudo useradd -m k1
data~$ echo "hello k1" > perm.txt
data~$ sudo passwd k1
```

[새로운 터미널을 연 후]

```
$ su - k1
$ cd /home/data
$ cat perm.txt # 파일 내용 확인
$ echo "me too" >> perm.txt
-su: perm.txt: 허가 거부
```

[결과 내용]

```
data@data:~$ su - k1
암호:
k1@data:~$ cd /home/data
k1@data:/home/data$ cat perm.txt
hello k1

k1@data:/home/data$ ls -l
```

합계 280

```
-rw-rw-r-- 1 data data 197546 2월 5 17:39 301757
```

```
-rw-rw-r-- 1 data data 85 2월 4 23:14 err.log
```

...

```
k1@data:/home/data$ echo "me too" >> perm.txt
```

-su: perm.txt: 허가 거부

[설명] : 허가 거부가 확인된다. 왜냐하면 다른 사용자에게 쓰기 권한이 없기 때문이다.

(실습 1)

(가) 빈 파일(a.txt)을 만들고 a.txt에 'hi hello'라는 출력을 넣는다. 그리고 cat 명령을 이용하여 확인해 보자.

(나) 새로운 유저 bigdata4를 만들고, 해당 유저로 접속 후, a.txt 파일에 새로운 내용 "I am student"를 추가해 보자.

(다) (나)가 예러가 발생한다. 자 그럼. chmod를 이용하여 권한을 부여한 후, 다시 (나)를 시도해 보자.

8-3 예제를 통한 권한 확인

(가) 만약 소유자와 그룹을 제외한 모든 사람들이 읽을 수 없도록 하려면 어떻게 해야 하나?

```
-rw-rw-r-- 1 data data 446 2월 4 23:02 result1.txt
```

```
$ chmod o-r result1.txt
```

(나) 만약 모든 사용자가 읽을 수 있도록 한다면 어떻게?

```
$ chmod o+r result1.txt
```

(다) 소유자에게 권한을 읽기 권한을 주고 읽기 권한을 해제한다.

```
chmod u+r result1.txt # 소유자에게 읽기 권한을 준다.
```

```
chmod u-r result1.txt # 소유자에게 읽기 권한을 해제한다.
```

```
chmod o+x result1.txt # 다른 사용자에게 실행 권한을 준다.
```

```
chmod o-x result1.txt # 다른 사용자에게 실행 권한을 해제한다.
```

(라) 쉘 스크립트를 만들고 실행 권한을 다른 유저에게 부여.

파일명 : hi.sh

```
#!/bin/bash
```

```
echo "hi hello good morning"
```

[다른 유저로 터미널을 열고 확인]

```
su - k1
```

```
cd /home/data
```

```
./hi.sh
```

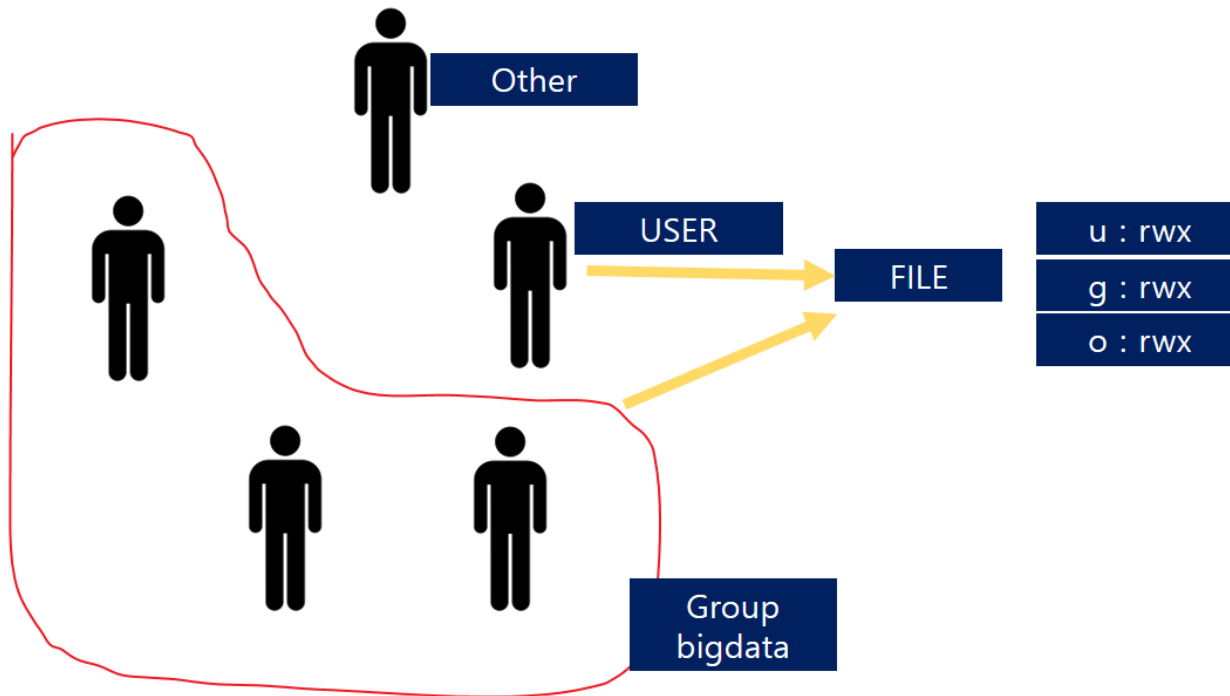
--> 허가 없다.

(실습 2)

(가) bigdata3 유저로 셸 스크립트 hi.sh를 작성 후, bigdata4유저로 접속 후, 셸 스크립트 hi.sh에 대한 권한을 부여해 보자.

(도전) 파일을 두개 이상 만들고, 이 파일들에 대한 새로운 내용을 추가하는 셸 스크립트를 만들고, 다른 유저에게도 실행권한을 부여해 보자.

8-4 그룹에 대해 알아보기



8-5 groupadd에 실행해보기

(가) 그룹으로 묶기 위한 유저 생성

```
sudo useradd -m u1
```

```
sudo useradd -m u2
```

```
sudo useradd -m u3
```

```
sudo passwd u1
sudo passwd u2
sudo passwd u3
```

확인

```
cat /etc/passwd | grep 계정명
```

```
data@data:~$ cat /etc/passwd | grep u1
u1:x:1003:1004::/home/u1:
data@data:~$ cat /etc/passwd | grep u2
u2:x:1004:1004::/home/u2:
data@data:~$ cat /etc/passwd | grep u3
u3:x:1005:1004::/home/u3:
```

(나) 각각의 유저로 터미널 접속

```
su - u1 # u1 유저로 접속
su - u2 # u2 유저로 접속
su - u3 # u3 유저로 접속
```

(다) 그룹을 생성 및 확인

```
sudo groupadd group1
```

그룹 확인

```
nano /etc/group
sambashare:x:128:data
vboxadd:x:1001:
vboxsf:x:999:data
duru:x:1002:
k1:x:1003:
mygroup1:x:1004:
group1:x:1005:
```

(라) 그룹에 유저 추가

```
sudo usermod -a -G group1 u1
sudo usermod -a -G group1 u2
sudo usermod -a -G group1 u3
```

이후에 로그인을 다시 해야 한다.

(실습 3)

(가) 유저 m1, m2, m3를 만들고 group2를 만들어 보자.

(도전) m1, m2, m3, u1, u2, u3를 묶는 그룹을 하나 만들어보자.

8-6 Summary

(가) 파일과 디렉터리는 권한을 **r, w, x**의 3가지 구분의 권한을 갖는다.

r (read) : 읽기

w (write) : 쓰기

x (execute) : 실행

(나) 맨 앞의 10자리는 다음의 구분을 갖는다.

d rwx rw- r-- :

맨 첫자리는 파일 또는 디렉터리 등의 구분을 나타낸다.

다음 세자리는 소유자의 권한이다.

다음 세자리는 그룹의 권한이다.

다음 세자리는 다른 사용자의 권한이다.

(다) chmod의 권한 정리

레퍼런스	클래스	설명
u	사용자	파일의 소유자
g	그룹	그 파일의 그룹 멤버인 사용자
o	다른 사람들	그 파일의 소유자나 혹은 그 그룹의 멤버가 아닌 사용자
a	모든 사람	위의 셋 모두, "ugo"와 같다

연산자	설명
+	지정된 모드들은 지정된 클래스들에 더한다
-	지정된 클래스들로부터 지정된 모드들은 지운다
=	지정된 클래스들을 위해서 지정된 모드들이 정확한 모드들로 만들어지게 된다

모드	이름	설명
r	읽기 (read)	파일을 읽거나 디렉터리 안 내용물의 리스트를 보여준다
w	쓰기 (write)	파일이나 디렉터리에 쓴다
x	실행하기 (excute)	파일을 실행하거나 디렉터리 트리로 되돌아간다
X	특별한 실행하기 (special excute)	이는 그 자체를 허가라고 보기보다는 <code>x</code> 대신에 사용될 수 있는 것으로 보아야 한다. 이 명령어는 디렉터리들의 현재 허가 상태와 관계없이 실행 허가를 디렉터리들에 적용하거나, (사용자, 그룹이나 다른 사람들에게) 벌써 최소한 한번의 실행 허가가 설정된 적이 있는 파일에 실행 허가를 적용한다. 이 명령어는 '+'와 함께 사용되는 경우와 일반적인 파일들 (텍스트 파일들과 같이)에서는 실행 허가를 설정하지 않고 큰 디렉터리 트리에 그룹이나 다른 사람들이 접근할 수 있게 해주는 <code>-R</code> 옵션과 결합하여 사용될 경우, 이 두 경우에만 유용하다. 두 번째 경우는 " <code>chmod -R a+rx</code> "를 막 사용한 경우에 주로 발생하며, " <code>chmod -R a+rx</code> " 대신 ' <code>X</code> '을 사용함으로써 같은 일을 할 수 있다.
s	셋유이드(setuid) / 기드(gid)	상세 설명은 특별한 모드들 섹션에 있다
t	스티키(sticky)	상세 설명은 특별한 모드들 섹션에 있다

기타 예제들

명령	설명
<code>chmod +r file</code>	모두에 대해 읽기를 추가한다
<code>chmod -x file</code>	모두에 대해 실행 허가를 제거한다
<code>chmod u=rw,go= file</code>	파일의 읽기와 쓰기를 그 파일의 소유자에게만 설정하고 그룹이나 다른 사람들에게 설정된 모든 허가를 지운다
<code>chmod +rw file</code>	모두가 읽고 쓸 수 있도록 file이라는 파일의 허가들을 바꾼다
<code>chmod -R u+w,go-w docs/</code>	사용자에게는 쓰기를 추가하고 다른 모든 사람들에게는 쓰기를 제거하기 위해서 docs라는 디렉터리와 그 안의 모든 내용물에 대한 허가들을 바꾼다.
<code>chmod 0 file</code>	모두의 모든 허가를 지운다
<code>chmod 666 file</code>	그 사용자, 그 그룹이나 다른 모든 이에 대해서 쓰거나 읽기 권한을 설정한다
<code>chmod 0755 file</code>	u=rwx (4+2+1)나 go=rx (4+1 & 4+1) 와 같다. 0 는 어떤 특별한 모드들도 지정하지 않는다.
<code>chmod 4755 file</code>	4 은 사용자 ID 설정을 지정하고 나머지는 u=rwx (4+2+1)나 go=rx (4+1 & 4+1)와 동일하다
<code>find path/ -type d -exec chmod a-x {} \;</code>	path/부터 시작하는 트리 안에 있는 모든 디렉터리들 (파일들의 목록은 볼 수 없다)의 실행 허가를 제거한다(파일들을 일치시키기 위해서 오직 '-type f'을 사용한다).
<code>find path/ -type d -exec chmod a+x {} \;</code>	만약 삼바(Samba) 쓰기에 대한 허가들을 재설정할 경우에 모든 사용자에게 대해서 디렉터리 열람을(예를 들어 ls) 허용한다.
<code>chmod -R u+rwx,g-rwx,o-rwx directory</code>	소유자 디렉터리에 대한 rwx(읽기, 쓰기, 실행)을, 소유자 파일들에 대해서는 rw(읽기, 쓰기)를, 그룹이나 다른 이들에 대해서는 ---를 설정하기 위해서 디렉터리 트리를 설정한다.
<code>chmod -R a-xX directory</code>	디렉터리 트리 안에 있는 모든 파일들에 대한 실행 허가를 제거한다. 반면 디렉터리 열람은 허용한다.

(라) 그룹과 관련된 명령어

그룹 추가 : groupadd

그룹 삭제 : groupdel

유저 추가 useradd

유저 삭제 userdel

(마) 그룹 확인

root@data:~# groups u1

u1 : mygroup1 group1

root@data:~# groups u2

u2 : mygroup1 group1

root@data:~# groups u3

u3 : mygroup1 group1

8-7 Quiz 및 실습

(가) r, w, x는 각각 어떤 권한인지 써보자.

(나) 아래 내용은 어떤 권한을 부여하는 것인지 써보자.

```
chmod +rw file
```

```
chmod -R u+w, go-w docs/
```

```
chmod 777 f1.txt
```

(다) 그룹을 추가하는 명령은 무엇인가요?