

# 파일과 디렉토리 관리 1

❖ 파일의 허가권의 종류 (개별 파일마다 독립적으로 설정할 수 있는 개별 허가권)

1. 퍼미션(Permission)
2. SetUID, SeGID, StickyBit
3. 속성 (attribution)

변위 : 소유자정보(파일과 디렉토리의 소유자 및 소유그룹 정보)

## 파일과 디렉토리의 퍼미션

### 파일과 디렉토리의 퍼미션(권한,Permission)

#### 종류

퍼미션	일반적인의미	파일	디렉토리
r	읽기(read) 권한	파일일 읽을 수 있음	디렉토리의 내용을 볼 수 있음
w	쓰기(write) 권한	파일에 저장, 삭제할 수 있음	디렉토리에 파일저장, 디렉토리이름변경, 삭제등 가능
x	실행(execution) 권한	파일을 실행할 수 있음	디렉토리에 접근할 수 있음(cd 로 접근가능, ls 로 리스트는 불가)
s	SetUID, SetGID 권한	파일소유자(SetUID), 그룹소유자(SetGID)의 권한으로 실행함	
t	Sticky Bit 권한	공유디렉토리로 사용됨	

#### 8 진수와 퍼미션의 표시관계 (r→4, w→2, x→ 1)

8 진수	표시 퍼미션	의미
0	---	아무 권한 없음
1	--x	실행(execution) 권한만 있음
2	-w-	쓰기(write) 권한만 있음
3	-wx	쓰기와 실행권한만 있음

4	r--	읽기(Read) 권한만 있음
5	r-x	읽기와 실행권한만 있음
6	rw-	읽기와 쓰기 권한만 있음
7	rx	읽기, 쓰기, 실행권한이 있음

## 퍼미션 설명

```
# mkdir -m 755 /test07
```

```
# ls -l / | grep test07
```

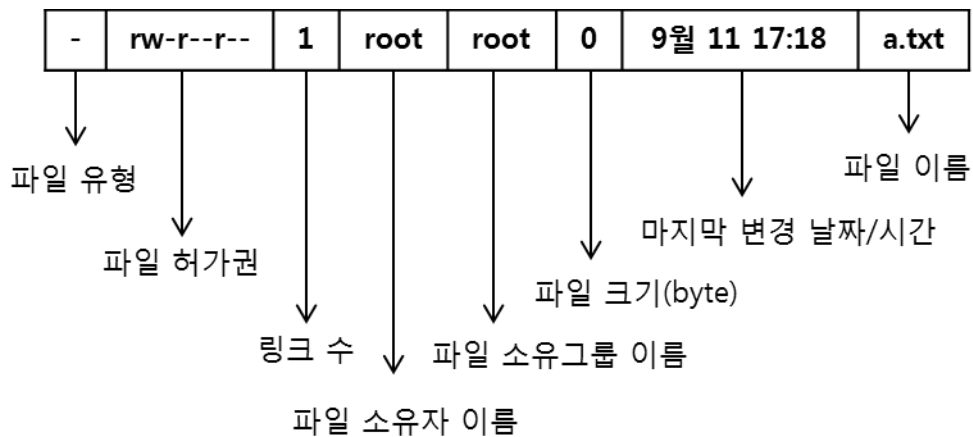
```
drwxr-xr-x. 2 root root 6 3 월 6 16:42 test07
```

```
# cd test07
```

```
# touch a.txt
```

```
# ls -l a.txt
```

```
-rw-r--r-- 1 root root 0 9 월 11 17:18 a.txt
```



### ❖ 퍼미션

-      r w    -    r    -    -    r    -    -  
 A.파일유형      B.소유자권한      C.그룹소속자권한      D.다른사용자권한

#### A. 파일유형

- : 일반파일을 의미

**b** : 블록구조의 특수파일(block Special File)을 의미함. (예, /dev/sda)

**c** : 입출력에 사용되는 특수파일(Character Special File)을 의미함.(예, /dev/console)

**d** : 디렉토리(Directory)를 의미하며, 디렉토리도 하나의 특수파일로 취급됨.

**l** : 심볼릭링크파일(Link File) 임을 의미함

**p** : 파이프파일(Pipe)임을 의미함

**s** : 소켓파일(Socket)임을 의미함

B. C. D. 소유자권한

--- : 아무 권한 없음

--x : 실행(execution) 권한만 있음

-w- : 쓰기(write) 권한만 있음

-wx : 쓰기와 실행권한만 있음

r-- : 읽기(Read) 권한만 있음

r-x : 읽기와 실행권한만 있음

rw- : 읽기와 쓰기 권한만 있음

rx : 읽기, 쓰기, 실행권한이 있음

## chmod

---

### 퍼미션 설정 명령어

#### 퍼미션 설정 방법 1 : 8 진수 숫자 이용 (권장)

---

형식 : **chmod [8 진수] 파일명**

# ls -l

**-rw-r--r--** 1 root root 0 9 월 11 17:18 a

# **chmod 755 a.txt**

# ls -l

**-rwxr-xr-x** 1 root root 0 9 월 11 17:18 a.txt

- 파일소유자에게 7 의 권한(모든권한) 즉 rwx 의 권한을 부여함
- 파일 그룹 소속자에게 5 의 권한 즉 r-x 의 권한을 부여함
- 다른 사용자들에게 5 의 권한 즉 r-x 의 권한을 부여함

✧ 실습 1

1. /perm 디렉토리를 755 permission 으로 생성

2. perm 디렉토리로 이동
3. /perm 디렉토리 밑에 a.txt 파일 생성
4. 생성한 a.txt 파일을 퍼미션과 함께 확인
5. a.txt 파일에 퍼미션 설정 : 소유자에게는 읽기,쓰기 권한부여, 그룹소속자와 다른 사용자에게는 아무 권한도 부여하지 않음
6. a.txt 파일 퍼미션 확인
 

```
# ls -l a.txt
```

```
-rw-----. 1 root root 0  3 월  6 16:48 a.txt
```
7. a.txt 파일에 퍼미션 설정 : 소유자에게 모든권한 부여, 그룹소속자와 다른 사용자에게 읽기 권한만 부여
8. a.txt 파일 퍼미션 확인
 

```
# ls -l a.txt
```

```
-rwxr--r--. 1 root root 0  3 월  6 16:48 a.txt
```

## 퍼미션 설정 방법 2 : 퍼미션 심볼문자 이용

---

### 형식 : **chmod** [심볼문자] 파일명

#### ❖ 심볼문자

- u : 소유자(user)를 의미함
- g : 그룹(group)을 의미함
- o : 다른사용자(others)들을 의미함
- a : 모두(all)를 의미함(default)

#### ❖ 설정모드

- + : 권한을 부여
- : 권한을 제거

#### ◇ 예

```
# cd /test07
```

```
# ls -l
```

```
-rwxr-xr-x 1 root root 0  9 월 11 16:45 a.txt
```

```
# chmod a+w a.txt
```

```
# ls -l
```

```
-rwxrwxrwx 1 root root 0  9 월 11 16:45 a.txt
```

➤ /test07/a.txt 파일에 모든 사용자(a)에게 쓰기(w) 권한 부여(+) 하기

✧ 실습 2 (심볼문자 이용하여 실습)

1. /perm/a.txt 파일의 현재 퍼미션 확인

```
# ls -l a.txt
```

```
-rwxr--r--. 1 root root 0  3 월  6 16:48 a.txt
```

2. /perm/a.txt 파일의 그룹소속자에 쓰기 권한 부여

3. /perm/a.txt 파일의 퍼미션 확인

```
# ls -l a.txt
```

```
-rwxrw-r--. 1 root root 0  3 월  6 16:48 a.txt
```

4. /perm/a.txt 파일의 그룹소속자와 다른사용자의 모든 권한을 제거

5. /perm/a.txt 파일의 퍼미션 확인

```
# ls -l a.txt
```

```
-rwx-----. 1 root root 0  3 월  6 16:48 a.txt
```

6. /perm/a.txt 파일에 그룹소속자와 다른 사용자에게 읽기 권한만 부여

7. /perm/a.txt 파일의 퍼미션 확인

```
# ls -l a.txt
```

```
-rwxr--r--. 1 root root 0  3 월  6 16:48 a.txt
```

## 특정 디렉토리내의 모든 파일에 대하여 동시에 퍼미션 설정하기

---

**chmod의 R 옵션** : 현재 디렉토리에 존재하는 모든 파일과 디렉토리뿐만아니라 그 디렉토리에 존재하는 서브디렉토리들과 그 안의 파일들 모두를 동일한 퍼미션으로 설정

✧ 예

1. test07 디렉토리의 상위 디렉토리로 이동

```
# cd ..
```

2. test07 디렉토리의 퍼미션 확인

```
# ls -l / | grep test07
```

```
drwxr-xr-x.  2 root root  19  9 월  11 16:43 test07
```

```
# ls -l /test07
```

```
-rwxrwxrwx. 1 root root 0  9 월 11 17:18 a.txt
```

```
# chmod 660 test07
```

```
# ls -l / | grep test07
```

```
drw-rw----.  2 root root  19  3 월  6 16:43 test07
```

```
# ls -l /test07
```

```
-rwxrwxrwx. 1 root root 0  3 월  6 16:43 a.txt
```

- test07 디렉토리의 권한만 660 으로 변경되고 하위 a.txt 파일은 권한이 변경되지 않음

```
# chmod -R 660 test07
```

```
# ls -l / | grep test07
```

```
drw-rw----.  2 root root  19  3 월  6 16:43 test07
```

```
# ls -l /test07
```

```
-rw-rw----. 1 root root 0  3 월  6 16:43 a.txt
```

- chmod 명령어에 -R 옵션을 사용하여 /test07 디렉토리의 권한을 660 으로 변경하자 그 하위의 디렉토리 및 파일도 같은 권한으로 변경됨

#### ◇ 실습 3

1. /perm 디렉토리와 그 하위 디렉토리 및 파일의 퍼미션 확인
2. /perm 디렉토리와 그 하위 디렉토리 및 파일의 퍼미션을 660 으로 변경
3. /perm 디렉토리 및 그 하위 디렉토리의 퍼미션 확인

```
# ls -l / | grep perm
```

```
drw-rw----.  2 root root  19  3 월  6 16:48 perm
```

```
# ls -l perm
```

```
-rw-rw----. 1 root root 0  3 월  6 16:48 a.txt
```

## SetUID, SetGID 그리고 Stickybit

---

### SetUID bit(4)

---

- SetUID가 설정된 파일을 실행하였을 경우 실행되는 동안에는 실행시킨 사용자의 권한이 아닌 파일의 소유자권한으로 실행된다. (단, 실행파일에 대한 일반 실행권한은 가지고 있어야 함)

### SetUID 설정 방법 : 일반 퍼미션 앞에 4를 붙여 설정

---

✧ 예

1. /test07/a.txt 파일에 SetUID 설정

```
# ls -l /test07
```

```
-rw-rw---- 1 root root 0  9 월 11 16:45 a.txt
```

```
# chmod 4755 /test07/a.txt
```

```
# ls -l /test07
```

```
-rwsr-xr-x 1 root root 0  9 월 11 16:45 a.txt
```

- 소유자 실행퍼미션 자리에 **s**라는 문자 설정

### SetGID bit(2)

---

- SetGID가 설정된 파일을 실행권한이 있는 어떤 누가 설정하더라도 그 파일의 그룹권한으로 실행된다.

### SetGID 설정 방법 : 일반 퍼미션 앞에 2를 붙여 설정

---

✧ 예

1. /test07/a.txt 파일에 SetGID 설정

```
# ls -l /test07
```

```
-rwsr-xr-x 1 root root 0  9 월 11 16:45 a.txt
```

```
# chmod 2754 /test07/a.txt
```

```
# ls -l /test07
```

```
-rwxr-sr-- 1 root root 0  9 월 11 16:45 a.txt
```

- 그룹소유자의 실행퍼미션 자리에 **s**라는 문자 설정

## Sticky bit(1)

---

- 파일생성을 아무나 할 수 있으며 생성된 파일은 각각 생성한 사용자의 소유로 되며 수정, 삭제 또한 생성한 사용자만이 그 파일을 수정하고, 지울 수 있도록 하는 "공용장소" 즉 /tmp 와 같은 디렉토리 설정을 하기 위하여 Sticky Bit 를 설정한다.

```
# ls -l / | grep tmp
```

```
drwxrwxrwt. 19 root root 4096  3 월  6 17:11 tmp
```

## Sticky bit 설정 방법 : 일반 퍼미션 앞에 1 을 붙여 설정

---

◇ 예

1. /test07/a.txt 파일에 Sticky bit 설정

```
# ls -alrt
```

```
drw-rw----  2 root root  4096  9 월 11 17:18 test07
```

```
# chmod 1777 /test07
```

```
# ls -l / | grep test07
```

```
drwxrwxrwt  2 root root  4096  9 월 12 16:45 test07
```

- 다른사용자 실행 퍼미션자리에 **t** 라는 문자 설정  
변경한 설정을 복원시킨다.

```
# chmod -R 755 test07
```

## 잘못된 설정

---

◇ 예

1. /test07/a.txt file 에 SetUID 설정

```
# cd /test07
```

```
# chmod 4655 a.txt
```

```
# ls -l
```

```
-rwSr-xr-x 1 root root 0  9 월 11 16:45 a.txt
```

- SetUID 를 설정하고 있지만 소유자에게 실행권한을 부여 하지 않았음
- 이럴 경우 소문자 s 가 아닌 대문자 S 로 표시됨

2. /test07/a.txt file 에 SetGID 설정

```
# chmod 2760 a.txt
```

```
# ls -l
```



```
-rwxrwS--- 1 root root 0  9 월 11 16:45 a.txt
```

- SetGID 를 설정하고 있지만 그룹소유자에게 실행권한을 부여 하지 않았음
- 이럴 경우 소문자 s 가 아닌 대문자 S 로 표시됨

3. /test07/a.txt file 에 Sticky bit 설정

```
# chmod 1760 a.txt
```

```
# ls -l
```

```
-rwxrw---T 1 root root 0  9 월 11 16:45 a.txt
```

- Sticky bit 를 설정하고 있지만 다른 사용자에게 실행권한을 부여 하지 않았음
- 이럴 경우 소문자 t 가 아닌 대문자 T 로 표시됨

✓ 주의

SetUID, SetGID, Sticky Bit 를 설정하면 모두 소문자 s,t 로 설정된다.

하지만 SetUID 등의 설정을 잘못할 경우 대문자 S,T 로 마킹된다.

이는 SetUID 는 설정하였지만, 일반퍼미션에 소유자퍼미션자리에 실행권한이 설정되지 않았기 때문에 소문자 s 가 아닌 대문자 S 로 마킹된 것이다.

**이것은 SetUID 는 설정되어 있으나 유효하지 않은 설정이다.**

**SetGID, Stickit bit 또한 마찬가지이다.**

✧ 복원

```
# chmod 755 a.txt
```

```
# ls -l
```

```
-rwxr-xr-x 1 root root 0  9 월 12 16:45 a.txt
```

✧ 실습 4

1. /perm/a.txt 파일 퍼미션 확인

2. /perm/a.txt 파일에 SetUID 을 설정, 단 파일에 대한 권한은 현재 설정된 권한대로 설정

3. /perm/a.txt 파일 퍼미션 확인

```
# ls -l /perm/a.txt
```

```
-rwSrW----. 1 root root 0  3 월  6 16:48 /perm/a.txt
```

4. 위의 설정에서 잘못된 부분 수정

5. /perm/a.txt 파일 퍼미션 확인

```
# ls -l /perm/a.txt
```

```
-rwsrw----. 1 root root 0  3 월  6 16:48 /perm/a.txt
```

6. /perm/a.txt 파일에 SetGID 을 권한이 유효하도록 설정, 단 다른 사용자권한은 아무것도 주지 않는다.

7. /perm/a.txt 파일 퍼미션 확인

```
# ls -l /perm/a.txt
```

```
-rwxrws---. 1 root root 0  3 월  6 16:48 /perm/a.txt
```

8. 다음은 setUID 설정이 어떻게 실행되는지 확인하는 실습이다.

	터미널 1(root 계정)	터미널 2(perm 계정)
1	/bin/tar 에 대한 퍼미션 확인	
2	perm 계정 생성 및 패스워드 설정	
3		perm 계정으로 login (명령어 : #su - perm)
4		/etc directory 를 tar 명령어로 묶는다. (명령어 : \$tar cvf etc.tar /etc) 실행되는 동안 다음 5 번 수행
5	tar 명령이 실행되는 프로세스 확인 (명령어 : #ps -ef   grep tar) ps 명령어는 현재 실행중인 프로세스를 확인하는 명령어로 맨앞의 계정명이 현재 실행하는 계정을 나타낸다. 위의 명령어를 통해 현재 tar 명령이 실행되는 유저를 확인한다.	
6		실행한 tar 명령이 끝났다면 묶은 파일을 확인해본다.(명령어 : \$ls -l)
7	/bin/tar 에 SetUID 설정을 한다. 단 일반 퍼미션에 대해선 기존 설정 그대로 한다.	
8	/bin/tar 에 설정한 퍼미션 확인	
9		\$tar cvf sbin.tar /sbin tar 명령이 실행되는 동안 다음 10 번 수행
10	#ps -ef   grep tar	

11		실행한 tar 명령이 끝났다면 묶은 파일을 확인해 본다. ( <code>\$ls -l</code> )
12	원복 :  <code>/bin/tar</code> 의 퍼미션을 일반퍼미션 755 로 원복시킨다.	

해석

	터미널 1(root 계정)	터미널 2(perm 계정)
1	<code>/bin/tar</code> 에 대한 퍼미션 확인 # <code>ls -l /bin/tar</code>  <code>-rwxr-xr-x 1 root root 223964 8??23 2007 /bin/tar</code>	
2	perm 계정 생성 및 패스워드 설정 # <code>useradd perm</code> # <code>passwd perm</code>	
3		perm 계정으로 login (명령어 : <code>#su - perm</code> )
4		<code>/etc</code> directory 를 tar 명령어로 묶는다. (perm 계정의 홈디렉토리에서 실행) (명령어 : <code>\$tar cvf etc.tar /etc</code> )  실행되는 동안 다음 5 번 수행
5	tar 명령이 실행되는 프로세스 확인 (명령어 : <code>#ps -ef   grep tar</code> )  ps 명령어는 현재 실행중인 프로세스를 확인하는 명령어로 맨앞의 계정명이 현재 실행하는 계정을 나타낸다. 위의 명령어를 통해 현재 tar 명령이 실행되는 유저를 확인한다.  # <code>ps -ef   grep tar</code>  <code>perm 7773 7738 4 19:30 pts/3 00:00:00 tar cvf etc.tar /etc</code>	

	→perm 유저로 tar 명령이 실행되고 있음을 확인할 수 있다.	
6		<p>실행한 tar 명령이 끝났다면 묶은 파일을 확인해본다. (명령어 : \$ls -l)</p> <pre>\$ ls -l</pre> <pre>-rw-rw-r-- 1 perm perm 78295040  9??12 19:31 etc.tar</pre> <p>→소유자가 perm 임을 확인</p>
7	<p>/bin/tar 에 SetUID 설정을 한다. 단 일반 퍼미션에 대해선 기존 설정 그대로 한다.</p> <pre># chmod 4755 /bin/tar</pre>	
8	<p>/bin/tar 에 설정한 퍼미션 확인</p> <pre># ls -l /bin/tar</pre> <pre>-rwsr-xr-x 1 root root 223964  8 월 23 2007 /bin/tar</pre>	
9		<pre>\$tar cvf sbin.tar /sbin</pre> <p>tar 명령이 실행되는 동안 다음 10 번 수행</p>
10	<pre>#ps -ef   grep tar</pre> <pre># ps -ef   grep tar</pre> <pre>root      8024  7738  4 19:39 pts/3 00:00:00 tar cvf sbin.tar /sbin</pre> <p>→SetUID 를 설정하기 전과는 다르게 tar 명령을 perm 계정에서 실행했음에도 불구하고 root 계정으로 tar 명령이 수행되고 있음을 확인</p>	
11		<p>실행한 tar 명령이 끝났다면 묶은 파일을 확인해 본다.(명령어 : \$ls -l)</p> <pre>\$ ls -l</pre> <pre>-rw-rw-r-- 1 perm perm 78295040  9 월 12 19:31 etc.tar</pre>

		-rw-rw-r-- 1 <b>root perm</b> 34560000 9 월 12 19:39 sbin.tar → sbin.tar 는 perm 계정으로 tar 를 실행했음에도 불구하고 root 소유자로 되었음을 확인
12	원복 : /bin/tar 의 퍼미션을 일반퍼미션 755 로 원복시킨다 # ls -l /bin/tar -rwsr-xr-x 1 root root 223964 8 월 23 2007 /bin/tar # chmod 755 /bin/tar # ls -l /bin/tar -rwxr-xr-x 1 root root 223964 8 월 23 2007 /bin/tar	