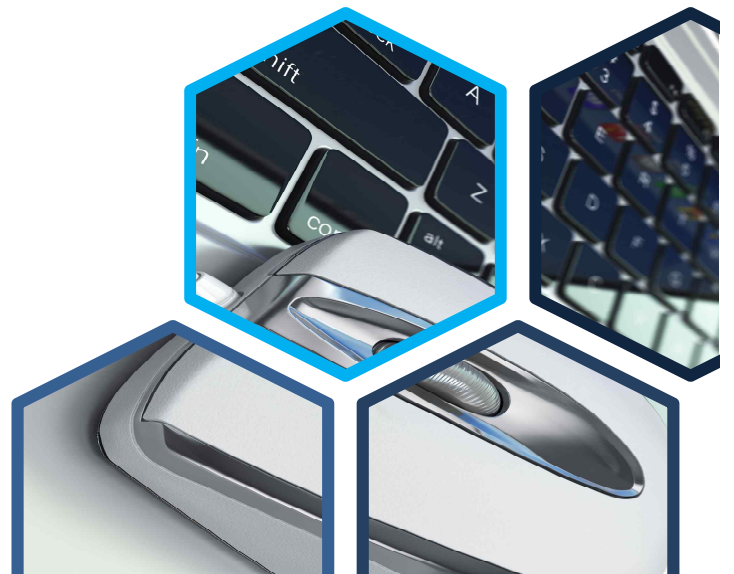




파일의 접근 권한 관리





학습목표

- 파일의 속성 및 접근 권한 방법을 설명할 수 있다.
- 기호와 숫자를 이용한 파일 접근 권한 변경을 할 수 있다.
- 기본 접근 권한 설정 방법을 설명할 수 있고, 특수 접근 권한 설정을 할 수 있다.



학습내용

- 파일의 속성 및 접근 권한
- 접근 권한 변경
- 접근 권한 설정



파일의 속성 및 접근 권한

1 파일의 속성



파일 접근 권한 보호

- 리눅스는 파일에 무단으로 접근하는 것을 방지하고 보호하는 기능 제공
- 사용자는 자신의 파일과 디렉터리 중에서 다른 사용자가 접근해도 되는 것과 그렇지 않은 것을 구분하여 접근 권한 제한



파일의 속성

```
[user1@]localhost ~] ls -l /etc/hosts  
-rw-r--r--. 1 root root 158 12월 7 /etc/hosts
```

번호	속성 값	의미
①	-	파일의 종류(-: 일반 파일, d: 디렉터리)
②	rw-r--r--	파일을 읽고 쓰고 실행할 수 있는 접근 권한 표시
③	1	하드 링크의 개수
④	root	파일 소유자의 로그인 ID
⑤	root	파일 소유자의 그룹 이름
⑥	158	파일의 크기(바이트 단위)
⑦	12월 7 2016	파일이 마지막으로 수정된 날짜
⑧	/etc/hosts	파일명



파일의 속성 및 접근 권한

1 파일의 속성



파일의 종류: file

- 파일 속성의 첫 번째 항목은 **파일의 종류**를 표시

-

일반 파일

d

디렉터리

- 기능: 파일의 종류를 알려주는 명령
- 형식: file [파일]
- 사용 예

```
[user1@]localhost ~]$ file /etc/hosts temp
/etc/hosts: ASCII text
temp:      directory
[user1@]localhost ~]$
```



파일의 속성 및 접근 권한

1 파일의 속성



파일의 접근 권한 표시

- ☁ 파일의 **소유자**와 **그룹**이나 **기타 사용자**들이 파일에 대해 가지고 있는 접근 권한을 표시



하드 링크의 개수

- ☁ **한 파일에 대해 여러 개의 파일명**을 가질 수 있도록 하는 기능



파일 소유자의 로그인 ID

- ☁ 리눅스에서 모든 파일은 소유자가 있음



파일 소유자의 그룹 이름: groups

- ☁ **ls -l 명령**에서 출력되는 그룹명은 파일이 속한 그룹
- ☁ 사용자가 속한 기본 그룹은 **시스템 관리자가 사용자를 등록할 때 결정**
- ☁ 기능: 사용자가 속한 그룹을 알려주는 명령
- ☁ 형식: groups [사용자명]



파일의 크기: **바이트** 단위



파일이 마지막으로 수정된 날짜



파일의 속성 및 접근 권한

2 파일의 접근 권한



접근 권한의 종류

- 읽기 권한, 쓰기 권한, 실행 권한 등 세 가지로 구성
- 파일과 디렉터리의 접근 권한

권한	파일	디렉터리
읽기	파일을 읽거나 복사 가능	ls 명령으로 디렉터리 목록을 볼 수 있음 (ls 명령의 옵션은 실행 권한이 있어야 사용 가능)
쓰기	파일을 수정·이동·삭제 가능 (디렉터리에 쓰기 권한이 있어야 함)	파일을 생성하거나 삭제 가능
실행	파일을 실행 가능 (셸 스크립트나 실행 파일의 경우)	cd 명령을 사용 가능 → 파일을 디렉터리로 이동하거나 복사 가능



파일의 속성 및 접근 권한

2 파일의 접근 권한



접근 권한의 표기 방법

- 사용자 카테고리별로 누가 파일을 읽고 쓰고 실행할 수 있는지를 문자로 표현한 것

r
읽기 권한

w
쓰기 권한

x
실행 권한

-
해당 권한이 없는 경우

- 사용자 카테고리별로 세 가지 권한의 부여 여부를 **rwX** 세 문자를 묶어서 표기

```
[user1@]localhost ~]$ ls -l /etc/hosts
-rw-r--r--. 1 root root 158 12월 2 2016 /etc/hosts
[user1@]localhost ~]$
```

rw-
소유자

r--
그룹

r--
기타 사용자



파일의 속성 및 접근 권한

2 파일의 접근 권한



접근 권한의 표기 방법



다양한 접근 권한 조합의 예

<code>rwXr-Xr-X</code>	<ul style="list-style-type: none"> ▶ 소유자는 읽기·쓰기·실행 권한을 모두 가짐 ▶ 그룹과 기타 사용자는 읽기·실행 권한을 가짐
<code>r-Xr-Xr-X</code>	▶ 소유자, 그룹, 기타 사용자 모두 읽기·실행 권한을 가짐
<code>rw-----</code>	<ul style="list-style-type: none"> ▶ 소유자만 읽기·쓰기 권한을 가짐 ▶ 그룹과 기타 사용자는 아무 권한 없음
<code>rw-rw-rw-</code>	▶ 소유자, 그룹, 기타 사용자 모두 읽기·쓰기 권한을 가짐
<code>rwXrwXrwx</code>	▶ 소유자, 그룹, 기타 사용자 모두 읽기·쓰기·실행 권한을 가짐
<code>rwX-----</code>	<ul style="list-style-type: none"> ▶ 소유자만 읽기·쓰기·실행 권한을 가짐 ▶ 그룹과 기타 사용자는 아무 권한 없음
<code>r-----</code>	▶ 소유자만 읽기 권한을 가짐



파일의 속성 및 접근 권한

2 파일의 접근 권한



접근 권한의 변경 명령: chmod

- 기능: 파일이나 디렉터리의 접근 권한을 변경함
- 형식: chmod [옵션] 권한 모드 파일 또는 디렉터리
- 옵션

-R	하위 디렉터리까지 모두 변경 가능
----	--------------------

- 권한 모드

기호 모드	숫자 모드
접근 권한을 변경하기 위해 문자와 기호를 사용하여 권한 표시	접근 권한을 변경하기 위해 숫자 사용

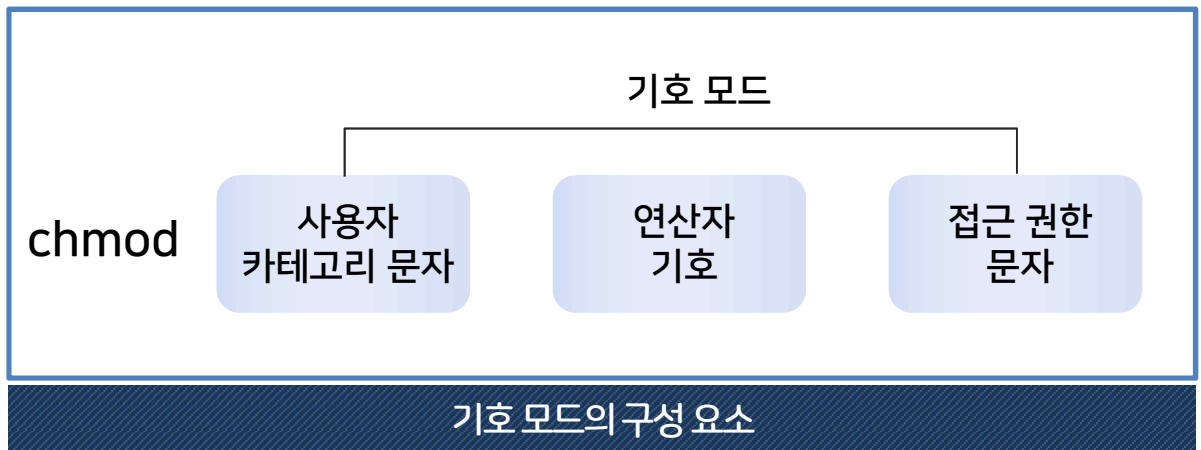


접근 권한 변경

1 기호를 이용한 파일 접근 권한 변경



기호 모드



기호 모드에서 사용하는 문자와 기호

구분	문자/기호	의미
사용자 카테고리 문자	u	파일 소유자
	g	소유자가 속한 그룹
	o	소유자와 그룹 이외의 기타 사용자
	a	전체 사용자
연산자 기호	+	권한 부여
	-	권한 제거
	=	접근 권한 설정
접근 권한 문자	r	읽기 권한
	w	쓰기 권한
	x	실행 권한



접근 권한 변경

1 기호를 이용한 파일 접근 권한 변경



기호 모드를 사용한 접근 권한 설정의 예

u+w	▶ 소유자(u)에게 쓰기(w) 권한 부여(+)
u-x	▶ 소유자(u)의 실행(x) 권한 제거(-)
g+w	▶ 그룹(g)에 쓰기(w) 권한 부여(+)
o-r	▶ 기타 사용자(o)의 읽기(r) 권한 제거(-)
g+wr	▶ 그룹(g)에 쓰기(w)와 실행(x) 권한 부여(+)
+wr	▶ 모든 사용자에게 umask에 따라 권한 부여(+)
a+rwx	▶ 모든 사용자에게 읽기(r), 쓰기(w), 실행(x) 권한 부여(+)
u=rwx	▶ 소유자(u)에게 읽기(r), 쓰기(w), 실행(x) 권한 부여(=)
go+w	▶ 그룹(g)과 기타 사용자(o)에게 쓰기(w) 권한 부여(+)
u+x,go+w	▶ 소유자(u)에게 실행(x) 권한 부여(+) ▶ 그룹(g)과 기타 사용자(o)에게 쓰기(w) 권한 부여(+)



접근 권한 변경

1 기호를 이용한 파일 접근 권한 변경



기호를 이용한 접근 권한 변경 예

- ☁ 현재 접근 권한 확인: `rw-r--r--`

```
[user1@]localhost ch5]$ ls -l
합계 4
-rw-r--r--. 1 user1 user1 158  7월 30  14:21 test.txt
[user1@]localhost ch5]$
```

- ☁ 소유자의 쓰기 권한을 제거: `u-w`

```
[user1@]localhost ch5]$ chmod u-w test.txt
[user1@]localhost ch5]$ ls -l
합계 4
-r--r--r--. 1 user1 user1 158  7월 30  14:21 test.txt
[user1@]localhost ch5]$
```



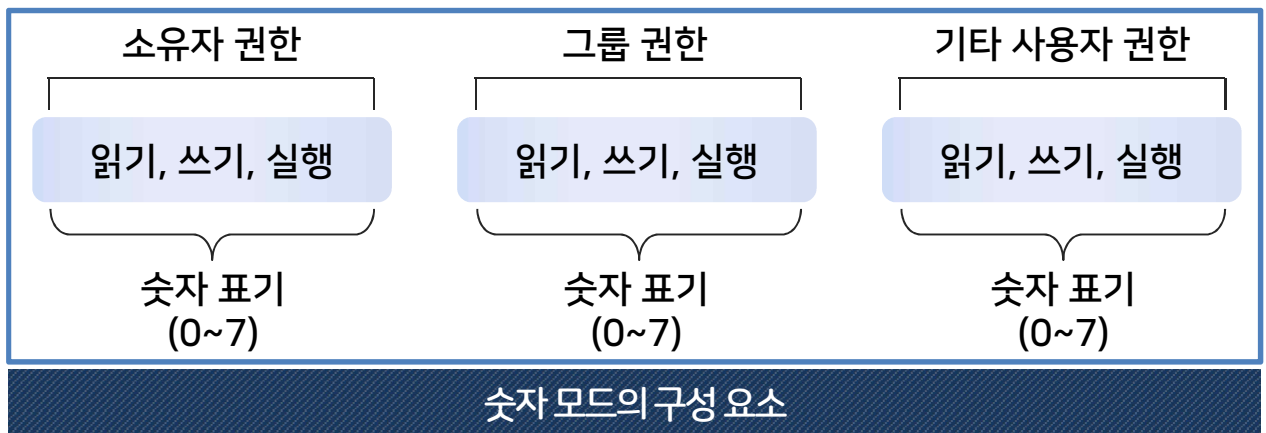
접근 권한 변경

2 숫자를 이용한 파일 접근 권한 변경

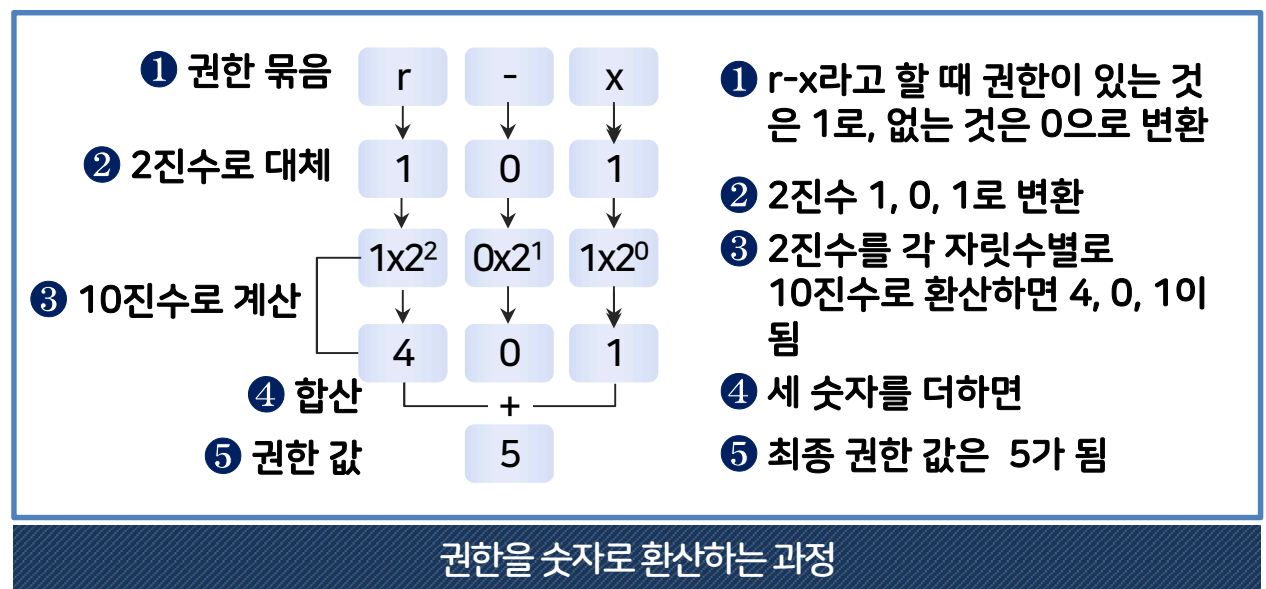


숫자로 환산하기

- 숫자 모드에서는 각 권한이 있고 없고를 0과 1로 표기하고 이를 다시 환산하여 숫자로 표현
- 카테고리별로 권한의 조합에 따라 0부터 7로 나타내는 것



권한을 숫자로 환산하는 과정





접근 권한 변경

2 숫자를 이용한 파일 접근 권한 변경

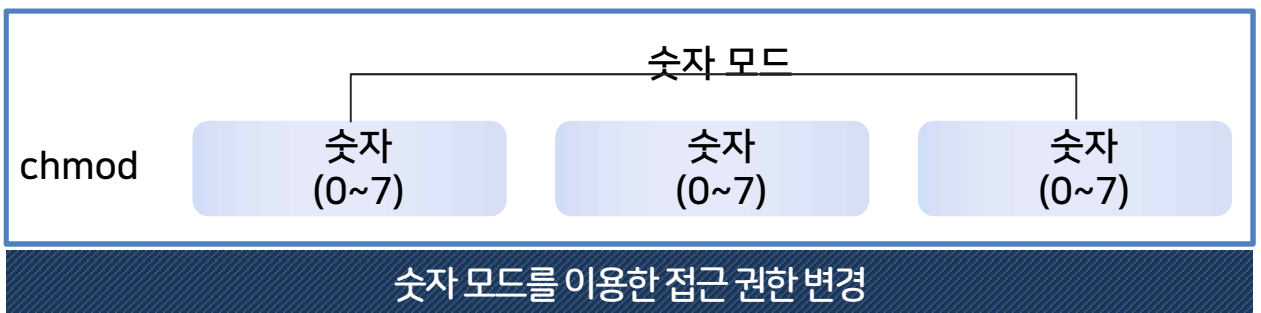


접근 권한과 숫자의 대응 관계

접근 권한	환산	숫자	의미
rwX	111	7	읽기, 쓰기, 실행
rw-	110	6	읽기, 쓰기
r-X	101	5	읽기, 실행
r--	100	4	읽기
-wX	011	3	쓰기, 실행
-w-	010	2	쓰기
--X	001	1	실행
---	000	0	권한이 없음



숫자 모드로 접근 권한 변경하기



- 숫자의 각 위치가 사용자 카테고리를 나타내기 때문에 **사용자 카테고리를 따로 지정할 필요가 없음**
- 항상 세 자리 수를 사용해야 하므로 변경하려는 사용자 카테고리의 권한뿐만 아니라 **그룹과 기타 사용자의 권한도 반드시 같이 명시**



접근 권한 변경

2 숫자를 이용한 파일 접근 권한 변경



숫자 모드로 접근 권한 변경하기 예

- ☁ 현재 접근 권한: 644(rw-r--r--)

```
[user1@]localhost ch5]$ ls -l
합계 4
-rw-r--r--. 1 user1 user1 158  7월 30  14:21 test.txt
[user1@]localhost ch5]$
```

- ☁ 소유자의 쓰기 권한을 제거: r--r--r--이므로 444

```
[user1@]localhost ch5]$ chmod 444 test.txt
[user1@]localhost ch5]$ ls -l
합계 4
-r--r--r--. 1 user1 user1 158  7월 30  14:21 test.txt
[user1@]localhost ch5]$
```

- ☁ 그룹에 쓰기와 실행 권한을 부여: r--rwxr--이므로 474

```
[user1@]localhost ch5]$ chmod 474 test.txt
[user1@]localhost ch5]$ ls -l
합계 4
-r--rwxr--. 1 user1 user1 158  7월 30  14:21 test.txt
[user1@]localhost ch5]$
```



기호 및 숫자를 이용한 파일 접근 권한 변경 실습 영상은
학습 콘텐츠에서 확인하실 수 있습니다.



접근 권한 설정

1 기본 접근 권한 설정



기본 접근 권한

- 리눅스에서는 파일이나 디렉터리를 생성할 때 기본 접근 권한이 **자동적으로 설정**
- 일반 파일의 경우

소유자와 그룹	기타 사용자
읽기와 쓰기 권한 설정	읽기 권한만 설정

- 디렉터리의 경우

소유자와 그룹	기타 사용자
읽기, 쓰기, 실행 권한 설정	읽기, 실행 권한만 설정



기본 접근 권한 확인하고 변경하기: umask

- 기능: 기본 접근 권한을 출력하거나 변경함
- 형식: `umask [옵션] [마스크 값]`
- 옵션

-S	마스크 값을 문자로 출력함
-----------	----------------

- 사용 예

```
umask 022    umask
```

- 아무 인자 없이 `umask` 명령만 사용할 경우 **기본 마스크 값 출력**

```
[user1@]localhost ch5]$ umask
0002
[user1@]localhost ch5]$
```




접근 권한 설정

1 기본 접근 권한 설정



마스크 값의 의미

- ☁ 파일이나 디렉터리 생성 시 **부여하지 않을 권한을 지정해놓는 것**
- ☁ 마스크 값이 002일 경우
 - 👤 -----w-이고, 기타 사용자에게 쓰기 권한은 부여하지 않겠다는 의미임
- ☁ 마스크 값을 바꾸면 파일이나 디렉터리를 생성할 때 적용되는 **기본 접근 권한도 변경**

```
[user1@]localhost ch5]$ umask 077
[user1@]localhost ch5]$ umask
0077
[user1@]localhost ch5]$
```



접근 권한 설정

1 기본 접근 권한 설정

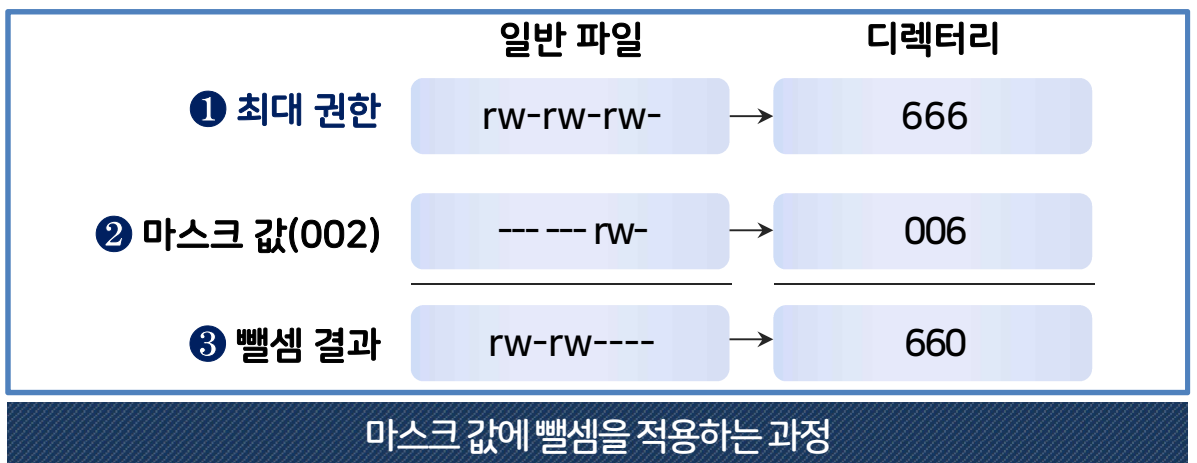


마스크 값의 적용 과정



umask 진리표

요청 권한	1	1	0	0
마스크	1	0	1	0
부여된 권한	0	1	0	0





접근 권한 설정

1 기본 접근 권한 설정



여러 가지 마스크 값



마스크 값의 의미

마스크 값	일반 파일	디렉터리	의미
022	644	755	그룹과 기타 사용자는 읽기만 가능함
077	600	700	그룹과 기타 사용자의 접근 권한을 모두 제거함
027	640	750	그룹은 읽기와 실행만 가능하고, 기타 사용자의 접근 권한을 모두 제거함



umask로 마스크 값을 바꿀 때 파일과 디렉터리에 **모두 적용**해봐야 함



마스크 값이 파일에는 적합하지만 **디렉터리에는 적합하지 않을 수도 있음**



접근 권한 설정

2 특수 접근 권한



특수 접근 권한

- 접근 권한은 원래 4자리
- 생략된 맨 앞자리는 특수 접근 권한 의미
- 맨 앞자리 숫자가 0이면 일반적인 접근 권한이지만 이 숫자가 1, 2, 4이면 특수 접근 권한이 설정됨

SetUID	맨 앞자리가 4
SetGID	맨 앞자리가 2
스티키 비트(Sticky bit)	맨 앞자리가 1



SetUID

- 해당 파일이 실행되는 동안에는 파일을 실행한 사용자의 권한이 아니라 **파일 소유자의 권한으로 실행**
- 파일에 SetUID 설정: SetUID는 접근 권한에서 맨 앞자리에 4를 설정
- SetUID가 설정되면 소유자의 실행 권한에 's'가 표시
- set.exe를 실행하면 **항상 user1의 권한을 가지고 실행된다는 의미**

```
[user1@]localhost ch5]$ ls -l /usr/bin/passwd
-rwsr-xr-x. 1 root root 27768 2월 11 20:02 /usr/bin/passwd
[user1@]localhost ch5]$
```

- /etc/shadow 파일은 **root 계정으로만 수정 가능**
- passwd 명령은 SetUID가 설정되어 있기 때문에 **소유자인 root 권한으로 실행이 되어 암호 변경 가능**
- SetUID를 이용한 해킹도 등장하여 보안에 신경을 써야 함



접근 권한 설정

2 특수 접근 권한



SetGID

- SetGID가 설정된 파일을 실행하면 해당 파일이 실행되는 동안에는 **파일 소유 그룹의 권한으로 실행**
- SetGID는 2755와 같이 접근 권한에서 **맨 앞자리에 2를 설정**



스티키 비트

- 스티키 비트는 디렉터리에 설정
- 디렉터리에 스티키 비트가 설정되어 있으면 이 디렉터리에는 **누구나 파일을 생성 가능**
- 파일은 파일을 생성한 계정으로 소유자가 설정되며, **다른 사용자가 생성한 파일은 삭제 불가**
- **/tmp 디렉터리가 대표적**
- 스티키 비트는 접근 권한에서 **맨 앞자리에 1을 설정**





특수 접근 권한 설정 실습 영상은
학습 콘텐츠에서 확인하실 수 있습니다.






핵심요약

1 파일의 속성 및 접근 권한

-  리눅스는 파일에 무단으로 접근하는 것을 방지하고 보호하는 기능을 제공
-  사용자는 자신의 파일과 디렉터리 중에서 다른 사용자가 접근해도 되는 것과 그렇지 않은 것을 구분하여 접근 권한을 제한

2 접근 권한 변경

-  기호 모드에서는 각 항목별로 사용할 수 있는 문자와 기호가 정해져 있고, 사용자 카테고리는 소유자, 그룹, 기타 사용자를 나타내는 문자로 표기되고, 연산자는 권한 부여나 제거를 나타내는 기호로 표기
-  숫자 모드에서는 각 권한이 있고 없음을 0과 1로 표기하고 이를 다시 10진수로 변환하여 숫자로 나타내며 카테고리별로 권한의 조합에 따라 0부터 7로 나타냄
-  접근 권한은 원래 4자리로 생략된 맨 앞자리는 특수 접근 권한 의미로 이 숫자가 0이 아니면 특수 접근 권한이 설정됨



핵심요약

3 접근 권한 설정

- 📁 일반 파일의 경우 소유자와 그룹은 읽기와 쓰기 권한이 설정되고 기타 사용자는 읽기 권한만 설정
- 📁 디렉터리의 경우 소유자와 그룹은 읽기, 쓰기, 실행 권한이 설정되고 기타 사용자는 읽기, 실행 권한만 설정