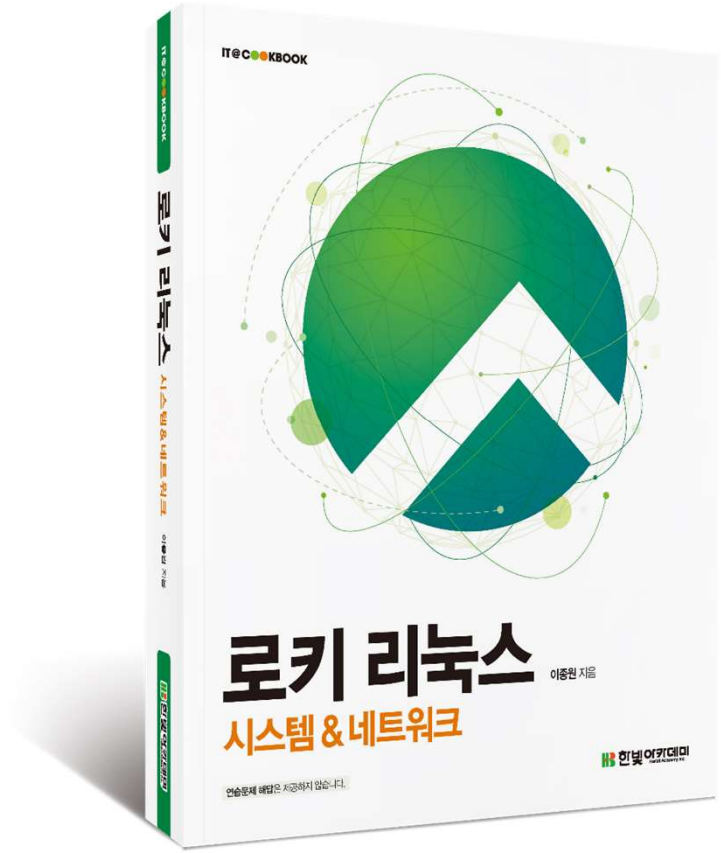

Chapter 03 파일 접근 권한 관리

목차

학습목표

- 파일 접근 권한의 개념을 이해할 수 있다.
- 접근 권한의 종류와 표기 방법을 이해하고 설명할 수 있다.
- 기호와 숫자로 표기된 접근 권한을 변경할 수 있다.
- 기본 접근 권한을 확인하고 원하는 값으로 바꿀 수 있다.

00 Preview

00 Preview

3장의 내용 구성

- 리눅스는 기본적으로 여러 사용자가 함께 사용하는 시스템이므로 각 사용자의 파일과 정보를 보호하기 위해 접근 권한을 제어하는 보안 관리 기능이 필수
- 리눅스에서 접근 권한을 설정하고 변경하는 방법을 알아봄
- 리눅스는 명확하게 접근 권한을 제어하며, 이는 유닉스에서부터 사용된 방법

01 파일 접근 권한

01 파일 접근 권한

다중 사용자 시스템

- 리눅스와 같은 다중 사용자 시스템은 사용자의 파일에 마음대로 접근할 수 없도록 보안 기능을 제공
- 사용자는 자신의 파일과 디렉터리 중에서 다른 사용자가 접근해도 되는 것과 그렇지 않은 것을 구분하여 접근 권한을 제한할 수 있음

접근 권한

- 해당 파일을 읽고 쓰고 실행할 수 있는 권한으로 사용자의 파일을 보호하는 가장 기본적인 보안 기능
- 리눅스에서는 사용자 카테고리별로 접근 권한을 다르게 부여하여 파일을 보호할 수 있음
- 카테고리별로 파일에 접 근하는 권한을 소유자나 시스템 관리자가 조정할 수 있음

01 파일 접근 권한

■ 접근 권한 종류

- 읽기 권한: 파일을 변경할 수는 없지만 내용은 볼 수 있는 권한
- 쓰기 권한: 파일의 내용을 보는 것뿐 아니라 수정하거나 삭제하는 것도 가능한 권한

표 3-1 파일과 디렉터리의 접근 권한

권한	파일	디렉터리
읽기	파일을 읽거나 복사할 수 있다.	ls 명령으로 디렉터리 목록을 볼 수 있다(ls 명령의 옵션은 실행 권한이 있어야 사용할 수 있다).
쓰기	파일을 수정 · 이동 · 삭제할 수 있다(디렉터리에 쓰기 권한이 있어야 한다).	파일을 생성하거나 삭제할 수 있다.
실행	파일을 실행할 수 있다(셸 스크립트나 실행 파일의 경우).	cd 명령을 사용할 수 있다. 파일을 디렉터리로 이동하거나 복사할 수 있다.

01 파일 접근 권한

■ 접근 권한 표기 방법

- 접근 권한 표기: 사용자 카테고리별로 누가 파일을 읽고 쓰고 실행할 수 있는지를 문자로 표현한 것
- 읽기 권한은 r, 쓰기 권한은 w, 실행 권한은 x
- 해당 권한이 없는 경우에는 -로 표기
- 사용자 카테고리별로 세 가지 권한의 부여 여부를 rwx 세 문자를 묶어서 표기
- 사용자 카테고리가 세 개이고 권한도 세 개이므로 총 아홉 개의 문자로 나타냄

```
[user1@localhost ~]$ ls -l /etc/hosts
-rw-r--r--. 1 root root 158  6월 23  2020 /etc/hosts
```

- 실제 접근 권한은 맨 앞의 -를 제외한 rw-r--r--
- 사용자 카테고리별로 세 문자 씩 한 묶음으로 표기한 것

01 파일 접근 권한

■ 접근 권한 표기 방법

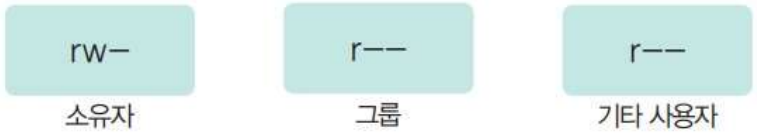

그림 3-1 파일의 접근 권한 표기

- 소유자가 읽기와 쓰기 권한을 가지고 있고, 그룹과 기타 사용자는 읽기 권한만 가지고 있음을 나타냄

표 3-2 다양한 접근 권한 조합의 예

접근 권한	의미
rwxr-xr-x	소유자는 읽기 · 쓰기 · 실행 권한을 모두 가지고, 그룹과 기타 사용자는 읽기 · 실행 권한을 가지고 있다.
r-xr-xr-x	소유자, 그룹, 기타 사용자 모두 읽기 · 실행 권한을 가지고 있다.
rw-------	소유자만 읽기 · 쓰기 권한을 가지고, 그룹과 기타 사용자에게는 아무 권한이 없다.
rw-rw-rw-	소유자, 그룹, 기타 사용자 모두 읽기 · 쓰기 권한을 가지고 있다.
rwxrwxrwx	소유자, 그룹, 기타 사용자 모두 읽기 · 쓰기 · 실행 권한을 가지고 있다.
rwx------	소유자만 읽기 · 쓰기 · 실행 권한을 가지고, 그룹과 기타 사용자에게는 아무 권한이 없다.
r-------	소유자만 읽기 권한을 가지고 있다.

01 파일 접근 권한

■ 접근 권한 변경 명령

- 파일의 소유자와 시스템 관리자는 접근 권한을 바꿀 수 있음
- 일반 사용자는 자신이 소유한 파일에서 자신의 접근 권한뿐 아니라 그룹과 기타 사용자의 권한도 변경할 수 있지만 다른 사용자가 소유한 파일의 접근 권한은 손댈 수 없음

chmod

- 기능 파일이나 디렉터리의 접근 권한을 변경한다.
- 형식 chmod [옵션] [권한 파일(디렉터리)]
- 옵션 -R: 하위 디렉터리까지 모두 변경할 수 있다.

- 기호 모드: 접근 권한을 변경하기 위해 문자와 기호를 사용하여 권한을 표시
- 숫자 모드: 접근 권한을 변경하기 위해 숫자를 사용

02 기호를 이용한 파일 접근 권한 변경

02 기호를 이용한 파일 접근 권한 변경

■ 기호 모드

- 사용자 카테고리: 소유자, 그룹, 기타 사용자를 나타내는 문자로 표기
- 연산자: 권한 부여나 제거를 나타내는 기호로 표기
- 접근 권한 기호: 읽기, 쓰기, 실행을 나타내는 문자를 사용

02 기호를 이용한 파일 접근 권한 변경

■ 기호 모드

표 3-3 기호 모드에서 사용하는 문자와 기호

구분	문자/기호	의미
사용자 카테고리 문자	u	파일 소유자
	g	파일 소유 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

02 기호를 이용한 파일 접근 권한 변경

■ 기호 모드

표 3-4 기호 모드를 사용한 접근 권한 설정의 예

권한 표기	의미
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+wx	그룹(g)에 쓰기(w)와 실행(x) 권한 부여(+)
+wx	모든 사용자에게 쓰기(w)와 실행(x) 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 설정(=)
go+w	그룹(g)과 기타 사용자(o)에게 쓰기(w) 권한 부여(+)
u+x,go+w	소유자(u)에게 실행(x) 권한을 부여하고(+) 그룹(g)과 기타 사용자(o)에게 쓰기(w) 권한 부여(+)

02 기호를 이용한 파일 접근 권한 변경

■ 기호 모드

- ch3 디렉터리와 파일을 준비

```
[user1@localhost ~]$ mkdir Test/ch3
[user1@localhost ~]$ cd Test/ch3
[user1@localhost ch3]$ cp /etc/hosts test.txt
```

- 현재 접근 권한을 확인. test.txt 파일의 현재 접근 권한은 rw-r--r—

```
[user1@localhost ch3]$ ls -l test.txt
-rw-r--r--. 1 user1 user1 158  8월 13 20:57 test.txt
```

- 소유자의 쓰기 권한을 제거하는 옵션은 u-w

```
[user1@localhost ch3]$ chmod u-w test.txt
[user1@localhost ch3]$ ls -l test.txt
-r--r--r--. 1 user1 user1 158  8월 13 20:57 test.txt
```

02 디렉터리 관련 명령

■ 기호 모드로 접근 권한 변경하기

① 그룹에 쓰기와 실행 권한을 부여(g+wx)

```
[user1@localhost ch3]$ chmod g+wx test.txt
[user1@localhost ch3]$ ls -l test.txt
-r--rwxr--. 1 user1 user1 158  8월 13 20:57 test.txt
```

② 기타 사용자에게 실행 권한을 부여(o+x)

```
[user1@localhost ~]$ mkdir Test
[user1@localhost ~]$ cd Test
[user1@localhost Test]$
```

③ 그룹과 기타 사용자의 실행 권한을 제거(go-x)

```
[user1@localhost ch3]$ chmod go-x test.txt
[user1@localhost ch3]$ ls -l test.txt
-r--rw-r--. 1 user1 user1 158  8월 13 20:57 test.txt
```

02 디렉터리 관련 명령

■ 기호 모드로 접근 권한 변경하기

④ 모두에게 실행 권한을 부여(a+x)

```
[user1@localhost ch3]$ ls -l test.txt
-r-xrwxr-x. 1 user1 user1 158  8월 13 20:57 test.txt
[user1@localhost ch3]$ chmod a+x test.txt
```

⑤ 소유자에게 쓰기 권한을 부여하고 그룹의 쓰기 권한은 제거(u+w,g-w)

```
[user1@localhost ch3]$ chmod u+w,g-w test.txt
[user1@localhost ch3]$ ls -l test.txt
-rwxr-xr-x. 1 user1 user1 158  8월 13 20:57 test.txt
```

- u+w,g-w와 같이 사용할 때는 u+w와 g-w 사이에 쉼표 외에 공백문자가 있으면 안 됨

02 디렉터리 관련 명령

■ 기호 모드로 접근 권한 변경하기

혼자해보기 기호 모드로 접근 권한 변경하기

❶ 앞의 [따라해보기]에서 사용한 test.txt 파일을 현재 접근 권한 상태에서 그대로 사용한다.
❷ 모든 사용자의 실행 권한을 한 번에 제거한다.
❸ 그룹과 기타 사용자에게 쓰기 권한을 한 번에 부여한다.
❹ 그룹과 기타 사용자의 읽기 · 쓰기 권한을 한 번에 모두 제거한다.
❺ 소유자의 쓰기 권한을 제거한다.
❻ 소유자의 읽기 권한을 제거한다.
❼ 소유자에게는 읽기 · 쓰기 권한을, 그룹과 기타 사용자에게는 읽기 권한을 한 번에 부여한다.

03 숫자를 이용한 파일 접근 권한 변경

03 숫자를 이용한 파일 접근 권한 변경

■ 숫자로 환산하는 방법

- 기호 모드에서는 카테고리와 권한을 모두 문자로 표현했으나 숫자 모드에서는 각 권한의 유무를 0과 1로 표기하고 이를 다시 환산하여 숫자로 표현함

그림 3-3 숫자 모드의 구성 요소

03 숫자를 이용한 파일 접근 권한 변경

■ 권한 묶음(rwx)을 숫자로 환산하는 방법

- 읽기, 쓰기, 실행의 각 권한이 있으면 1, 없으면 0으로 바꾼 후 이를 2진수 세 자리로 간주하여 환산하면 권한 묶음을 0~7까지의 숫자 중 하나로 표기할 수 있음
- 접근 권한이 ❶과 같이 r-x일 때, 권한이 있는 것은 1로, 없는 것은 0으로 바꾸면 ❷처럼 2진수 1, 0, 1이 됨
- 이 2진수를 자릿수별로 10진수로 환산하면 ❸처럼 4, 0, 1이 되고, 이 세 숫자를 더하면 최종 권한 값은 ❺와 같이 5가 됨
- 즉, 접근 권한 r-x는 숫자로 5가 되는 것

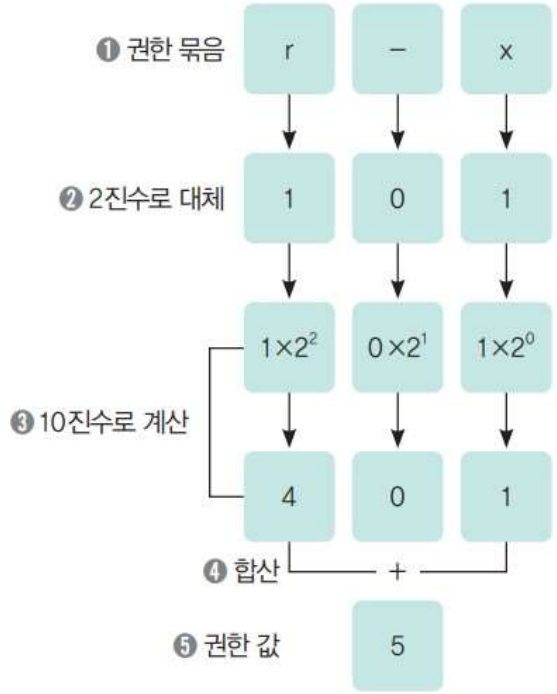

그림 3-4 권한을 숫자로 환산하는 과정

03 숫자를 이용한 파일 접근 권한 변경

■ 접근 권한의 표기와 숫자의 대응 관계

표 3-5 접근 권한과 숫자의 대응 관계

접근 권한	환산	숫자	의미
rwx	111 → 4+2+1	7	읽기, 쓰기, 실행
rw-	110 → 4+2+0	6	읽기, 쓰기
r-x	101 → 4+0+1	5	읽기, 실행
r--	100 → 4+0+0	4	읽기
-wx	011 → 0+2+1	3	쓰기, 실행
-w-	010 → 0+2+0	2	쓰기
--x	001 → 0+0+1	1	실행
---	000 → 0+0+0	0	권한이 없음

- 숫자 7, 6, 5, 4, 0의 형태가 자주 사용되며 1, 2, 3은 잘 사용하지 않음

03 숫자를 이용한 파일 접근 권한 변경

■ 접근 권한의 표기와 숫자의 대응 관계

- 소유자 권한, 그룹 권한, 기타 사용자 권한을 각각 숫자로 환산하면 전체 접근 권한을 나타내는 숫자 세 개가 나옴. 이 세 숫자를 연결하면 전체 접근 권한을 나타낼 수 있음

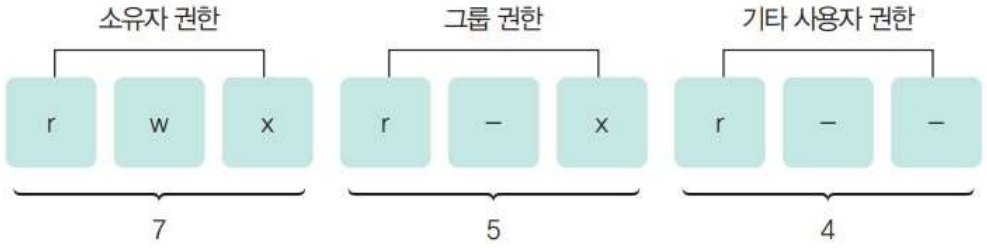

그림 3-5 전체 접근 권한을 숫자로 표기한 예

03 숫자를 이용한 파일 접근 권한 변경

■ 전체 접근 권한을 숫자로 표시한 형태

표 3-6 숫자로 표현한 접근 권한의 예

접근 권한	숫자 모드	접근 권한	숫자 모드
rwxrwxrwx	777	rw-r--r--	644
rwxr-xr-x	755	rwx------	700
rw-rw-rw	666	rw-r-----	640
r-xr-xr-x	555	r--------	400

- chmod 명령의 숫자 모드에서 사용하는 것이 이와 같은 형태

03 숫자를 이용한 파일 접근 권한 변경

■ 숫자 모드를 이용한 접근 권한 변경

그림 3-6 숫자 모드를 이용한 접근 권한 변경

- 기호 모드와 다른 점은 숫자의 각 위치가 사용자 카테고리를 나타내기 때문에 사용자 카테고리를 따로 지정할 필요가 없다는 것
- 항상 세 자리 수를 사용해야 하므로 변경하려는 한 카테고리의 권한뿐 아니라 다른 카테고리의 권한도 반드시 같이 명시해야 함

03 숫자를 이용한 파일 접근 권한 변경

■ 숫자 모드로 chmod 명령을 사용하는 방법

- test.txt 파일의 현재 접근 권한은 다음과 같이 644(rw-r--r--)

```
[user1@localhost ch3]$ ls -l test.txt
-rw-r--r--. 1 user1 user1 158  8월 13 20:57 test.txt
```

- 소유자의 쓰기 권한을 제거하면 기호로는 u-w이지만 숫자로 표기하면 최종 권한의 형태가 r--r--r--이므로 444가 됨

```
[user1@localhost ch3]$ chmod 444 test.txt
[user1@localhost ch3]$ ls -l test.txt
-r--r--r--. 1 user1 user1 158  8월 13 20:57 test.txt
```

- 그룹에 쓰기와 실행 권한을 부여. 최종 권한 형태가 r--rwxr--이므로 숫자로 표시하면 474가 됨

```
[user1@localhost ch3]$ chmod 474 test.txt
[user1@localhost ch3]$ ls -l test.txt
-r--rwxr--. 1 user1 user1 158  8월 13 20:57 test.txt
```

03 숫자를 이용한 파일 접근 권한 변경

■ 숫자 모드로 접근 권한 변경하기

① 기타 사용자에게 실행 권한을 부여(o+x)

- r--rwxr-x → 475

```
[user1@localhost ch3]$ chmod 475 test.txt
[user1@localhost ch3]$ ls -l test.txt
-r--rwxr-x. 1 user1 user1 158  8월 13 20:57 test.txt
```

② 그룹과 기타 사용자의 실행 권한을 제거(go-x)

- r--rw-r-- → 464

```
[user1@localhost ch3]$ chmod 464 test.txt
[user1@localhost ch3]$ ls -l test.txt
-r--rw-r--. 1 user1 user1 158  8월 13 20:57 test.txt
```

03 숫자를 이용한 파일 접근 권한 변경

■ 숫자 모드로 접근 권한 변경하기

③ 모두에게 실행 권한을 부여(a+x)

- r-xrwxr-x → 575

```
[user1@localhost ch3]$ chmod 575 test.txt
[user1@localhost ch3]$ ls -l test.txt
-r-xrwxr-x. 1 user1 user1 158  8월 13 20:57 test.txt
```

④ 소유자에게 쓰기 권한을 부여하고 그룹의 쓰기 권한은 제거(u+w,g-w)

- rwxr-xr-x → 755

```
[user1@localhost ch3]$ chmod 755 test.txt
[user1@localhost ch3]$ ls -l test.txt
-rwxr-xr-x. 1 user1 user1 158  8월 13 20:57 test.txt
```

03 숫자를 이용한 파일 접근 권한 변경

■ 숫자 모드로 접근 권한 변경하기

⑤ 소유자의 권한만 남기고 나머지 사용자의 권한은 모두 제거

- rwx------ → 700

```
[user1@localhost ch3]$ chmod 700 test.txt
[user1@localhost ch3]$ ls -l test.txt
-rwx------. 1 user1 user1 158  8월 13 20:57 test.txt
```

혼자해보기 숫자 모드로 접근 권한 변경하기

❶ test.txt 파일을 현재 접근 권한 상태에서 그대로 사용한다.
❷ 소유자의 권한을 읽기와 쓰기로 변경한다.
❸ 그룹과 기타 사용자도 읽고 쓰기가 가능하도록 변경한다.
❹ 기타 사용자의 쓰기 권한을 제거한다.
❺ 그룹의 쓰기 권한을 제거한다.
❻ 소유자의 읽기 권한만 남겨놓고 모든 권한을 제거한다.
❼ 현재 파일에 설정된 모든 권한을 제거한다.

04 기본 접근 권한 설정

04 기본 접근 권한 설정

■ 기본 접근 권한

- 리눅스 - 파일이나 디렉터리를 생성할 때 기본 접근 권한이 자동으로 설정
- 일반 파일: 소유자-읽기와 쓰기 권한이 설정/ 그룹과 기타 사용자 - 읽기 권한만 설정
- 디렉터리: 소유자 - 읽기·쓰기·실행 권한이 설정/ 그룹과 기타 사용자 - 읽기·실행 권한만 설정

```
[user1@localhost ch3]$ touch linux.txt
[user1@localhost ch3]$ mkdir temp
[user1@localhost ch3]$ ls -l
합계 4
-rw-r--r--. 1 user1 user1   0  8월 13 21:22 linux.txt
drwxr-xr-x. 2 user1 user1   6  8월 13 21:22 temp
-rwx------. 1 user1 user1 158  8월 13 20:57 test.txt
```

- 기본 접근 권한은 리눅스 시스템에 설정된 기본값에 따른 것으로, 이 기본값을 바꾸면 기본 접근 권한도 바꿀 수 있음

04 기본 접근 권한 설정

■ 기본 접근 권한 확인 및 변경

umask

- 기능 기본 접근 권한을 출력하거나 변경한다.
- 형식 umask [옵션] [마스크 값]
- 옵션 -S: 마스크 값을 문자로 출력한다.
- 사용 예 umask 002
  umask

- 아무 인자 없이 umask 명령만 사용하면 현재 설정된 기본 마스크 값을 보여줌

```
[user1@localhost ch3]$ umask
0022
```

04 기본 접근 권한 설정

■ 마스크 값의 의미

- 마스크 값: 파일이나 디렉터리 생성 시 부여하지 않을 권한을 지정해 놓는 것

```
[user1@localhost ch3]$ umask -S
u=rwx,g=rx,o=rx
```

- umask 명령에 -S 옵션을 적용하여 마스크 값을 문자로 출력하면 소유자에게 읽기·쓰기·실행 권한을 설정하고 그룹과 기타 사용자에게는 읽기와 실행 권한만 설정한다는 것을 알 수 있음

04 기본 접근 권한 설정

■ 마스크 값 변경하기

```
[user1@localhost ch3]$ umask 077
[user1@localhost ch3]$ umask
0077
```

- umask로 마스크 값을 바꾸면 파일이나 디렉터리를 생성할 때 적용되는 기본 접근 권한도 바뀜

```
[user1@localhost ch3]$ touch han.txt
[user1@localhost ch3]$ ls -l han.txt
-rw-------. 1 user1 user1 0  8월 13 21:28 han.txt
```

- han.txt 파일의 기본 접근 권한이 600(rw-------)임을 알 수 있음

04 기본 접근 권한 설정

■ 마스크 값의 적용 과정

- 마스크 값은 파일이 생성될 때마다 적용
- 마스크 값을 비트로 표시했을 때 그 값이 1인 경우 대응하는 권한은 제외

표 3-7 umask 진리표

요청 권한	1	0	1	0
마스크	1		0	
부여된 권한	0		1	0

- 마스크 값이 1이면 요청 권한이 1이든 0이든 상관없이 부여되는 권한은 0이 되고, 마스크 값이 0이면 해당 요청 권한이 그대로 부여된다는 것을 의미

04 기본 접근 권한 설정

■ 마스크 값 계산 방법

- 일반적인 사용자 환경에서 마스크 값을 쉽게 이해하는 방법은 최대 접근 권한에서 마스크 값을 빼는 것
- 실행 파일도 기본적으로 실행 권한을 가져야 하므로 디렉터리와 같은 형태로 권한이 부여됨

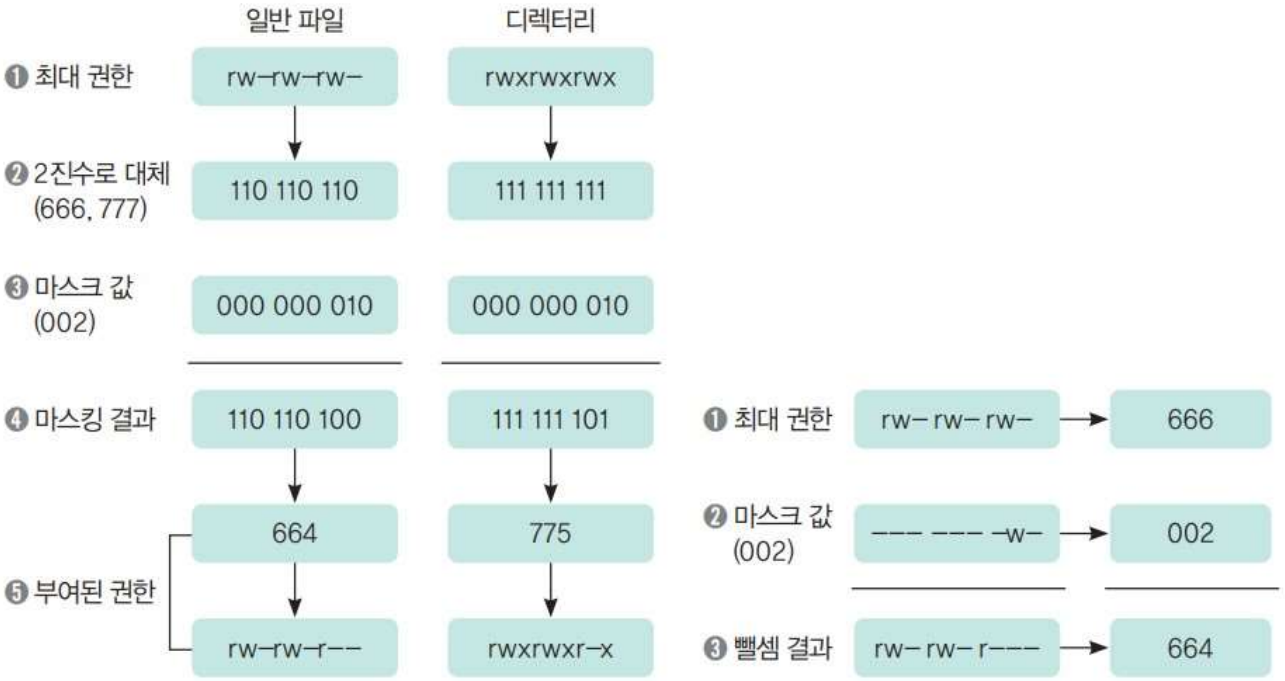

그림 3-7 마스크 값 계산 방법

04 기본 접근 권한 설정

■ 여러 가지 마스크 값

- 리눅스의 기본 마스크 값은 배포판과 버전에 따라 다른데 이를 필요에 따라 적절히 변경하여 자신의 파일과 디렉터리를 보호할 수 있음

표 3-8 마스크 값의 의미

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

- umask로 마스크 값을 바꿀 때 파일과 디렉터리에 모두 적용해 봐야 함. 마스크 값이 파일에는 적합하지만 디렉터리에는 적합하지 않을 수도 있기 때문

04 기본 접근 권한 설정

■ 기본 접근 권한 변경하기

① 현재의 기본 접근 권한을 확인

```
[user1@localhost ch3]$ umask
0077
```

② 소유자와 그룹은 읽기와 쓰기가 가능하고 기타 사용자는 읽기만 할 수 있도록 기본 접근 권한을 변경

```
[user1@localhost ch3]$ umask 002
[user1@localhost ch3]$ umask
0002
```

04 기본 접근 권한 설정

■ 기본 접근 권한 변경하기

③ 파일을 생성하여 기본 접근 권한이 적용되었는지 확인

```
[user1@localhost ch3]$ touch mask.txt
[user1@localhost ch3]$ ls -l mask.txt
-rw-rw-r--. 1 user1 user1 0  8월 13 21:40 mask.txt
```

혼자해보기 기본 접근 권한 변경하기

❶ 소유자와 그룹은 읽기와 쓰기가 가능하고 기타 사용자에게는 아무 권한이 없도록 마스크 값을 변경한다.
❷ mask2.txt 파일과 tmp 디렉터리를 생성하여 기본 접근 권한이 변경되었는지 확인한다.
❸ 기본 접근 권한을 원래의 기본 마스크 값(022)으로 변경한다.

05 특수 접근 권한 설정

05 특수 접근 권한 설정

기본 접근 권한

- 특수 접근 권한 : 일반적인 접근 권한 외에 리눅스에서 제공하는 특별한 접근 권한

```
[user1@localhost ch3]$ umask
0022
```

- umask가 출력하는 네 자리 중 맨 앞자리는 특수 접근 권한을 나타냄
- 맨 앞자리의 숫자가 0이면 일반적인 접근 권한이지만 1, 2, 4이면 특수 접근 권한이 설정됨

특수 접근 권한

- SetUID: 맨 앞자리가 4
- SetGID: 맨 앞자리가 2
- 스티키 비트sticky bit: 맨 앞자리가 1

05 특수 접근 권한 설정

SetUID

- SetUID가 설정된 파일을 실행하면 해당 파일이 실행되는 동안에는 파일을 실행한 사용자의 권한이 아니라 파일 소유자의 권한이 적용됨

```
[user1@localhost ch3]$ touch set.exe
[user1@localhost ch3]$ chmod 755 set.exe    ➞ 실행 권한을 부여한다.
[user1@localhost ch3]$ ls -l set.exe
-rwxr-xr-x. 1 user1 user1 0  8월 13 21:45 set.exe
```

- SetUID는 접근 권한에서 맨 앞자리에 4를 설정해야 함

```
[user1@localhost ch3]$ chmod 4755 set.exe
[user1@localhost ch3]$ ls -l set.exe
-rwsr-xr-x. 1 user1 user1 0  8월 13 21:45 set.exe
```

- SetUID가 설정되면 소유자의 실행 권한에 's'가 표시됨. set.exe 파일을 실행하면 항상 user1의 권한으로 실행한다는 의미

05 특수 접근 권한 설정

SetUID

```
[user1@localhost ch3]$ ls -l /usr/bin/passwd
-rwsr-xr-x. 1 root root 32656  5월 15  2022 /usr/bin/passwd
```

- 계정의 암호가 저장된 /etc/shadow파일은 root 계정으로만 수정할 수 있음
- 일반 사용자가 passwd명령으로 암호를 바꾸려고 할 때 본인의 권한으로 실행하면 암호를 바꿀 수 없을 것. /etc/shadow 파일을 수정할 수 없기 때문
- passwd 명령에는 SetUID가 설정되어 있기 때문에 소유자인 root 권한으로 실행되어 /etc/shadow 파일을 수정해 암호를 바꿀 수 있음

05 특수 접근 권한 설정

SetGID

- SetGID가 설정된 파일을 실행하면 해당 파일이 실행되는 동안에는 파일 소유 그룹의 권한으로 실행
- SetGID는 접근 권한의 맨 앞자리에 2를 설정해야 함

```
[user1@localhost ch3]$ chmod 2755 set.exe
[user1@localhost ch3]$ ls -l set.exe
-rwxr-sr-x. 1 user1 user1 0  8월 13 21:45 set.exe
```

05 특수 접근 권한 설정

■ 스티키 비트

- 스티키 비트는 디렉터리에 설정하며, 디렉터리에 스티키 비트가 설정되어 있으면 이 디렉터리에는 누구나 파일을 생성할 수 있음
- 파일은 파일을 생성한 계정으로 소유자가 설정되고, 다른 사용자가 생성한 파일은 삭제할 수 없음
- 스티키 비트가 설정되면 기타 사용자의 실행 권한에 't'가 표시 됨

```
[user1@localhost ch3]$ ls -ld /tmp
drwxrwxrwt. 15 root root 4096  8월 13 20:59 /tmp
```

- 스티키 비트는 접근 권한의 맨 앞자리에 1을 설정해야 함

```
[user1@localhost ch3]$ chmod 1755 temp
[user1@localhost ch3]$ ls -ld temp
drwxr-xr-t. 2 user1 user1 6  8월 13 21:22 temp
```

05 특수 접근 권한 설정

■ 특수 접근 권한 설정 오류

- 특수 권한을 설정하는 파일이나 디렉터리 모두 실행 권한을 가지고 있어야 함
- 실행 권한이 없는 파일에 SetUID나 SetGID를 설정하면 's'가 아니라 'S'가 표시됨
- 디렉터리에 실행 권한이 없는데 스티키 비트를 설정하면 'T'가 표시됨
- 대문자 S나 T가 나타나면 특수 접근 권한을 잘못 설정한 것

```
[user1@localhost ch3]$ chmod 4644 test.txt
[user1@localhost ch3]$ ls -l test.txt
-rwSr--r--. 1 user1 user1 158  8월 13 20:57 test.txt
[user1@localhost ch3]$ chmod 2644 test.txt
[user1@localhost ch3]$ ls -l test.txt
-rw-r-Sr--. 1 user1 user1 158  8월 13 20:57 test.txt
[user1@localhost ch3]$ chmod 1644 temp
[user1@localhost ch3]$ ls -ld temp
drw-r--r-T. 2 user1 user1 6  8월 13 21:22 temp
```

Thank you!

한빛아카데미 Hanbit Academy, Inc.