

목차

- 00. 개요
- 01. 파일의 속성
- 02. 파일의 접근 권한
- 03. 기호를 이용한 파일 접근 권한 변경
- 04. 숫자를 이용한 파일 접근 권한 변경
- 05. 기본 접근 권한 설정
- 06. 특수 접근 권한

■ 파일 접근 권한 보호

- 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

■ 파일의 속성

```
user1@myubuntu:~$ ls -l /etc/hosts
-rw-r--r-- 1 root root 223 11월  8 23:13 /etc/hosts
user1@myubuntu:~$
```

표 5-1 파일의 속성

| 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 | 223 | 파일의 크기(바이트 단위) |
| ⑦ | 11월 8 23:13 | 파일이 마지막으로 수정된 날짜와 시간 |
| ⑧ | /etc/hosts | 파일명 |

01 파일의 속성

■ 파일의 종류

- 파일 속성의 첫 번째 항목은 파일의 종류를 표시
- -는 일반 파일을, d는 디렉토리를 의미
- 파일의 종류를 알려주는 명령

file

- **기능** 지정한 파일의 종류를 알려준다.
- **형식** file 파일
- **사용 예** file /etc/services

```
user1@myubuntu:~$ file /etc/hosts temp
/etc/hosts: ASCII text
temp:      directory
user1@myubuntu:~$
```

■ 파일의 접근 권한 표시

- 파일의 소유자와 그룹이나 기타 사용자들이 파일에 대해 가지고 있는 접근 권한을 표시

■ 하드 링크의 개수

- 하드 링크는 한 파일에 대해 여러 개의 파일명을 가질 수 있도록 하는 기능

01 파일의 속성

■ 파일 소유자의 로그인 ID

- 리눅스에서 모든 파일은 소유자가 있음

■ 파일 소유자의 그룹 이름

- `ls -l` 명령에서 출력되는 그룹명은 파일이 속한 그룹
- 사용자가 속한 기본 그룹은 시스템 관리자가 사용자를 등록할 때 결정
- 사용자가 속한 그룹을 알려주는 명령은 `groups`

groups

- **기능** 사용자가 속한 그룹을 알려준다.
- **형식** `groups [사용자명]`

```
user1@myubuntu:~$ groups
user1 adm cdrom sudo dip plugdev lpadmin sambashare
user1@myubuntu:~$ groups root
root : root
user1@myubuntu:~$
```

■ 파일의 크기: 바이트 단위

■ 파일이 마지막으로 수정된 날짜

02 파일의 접근 권한

■ 접근 권한의 종류

- 읽기 권한, 쓰기 권한, 실행 권한 등 세 가지로 구성

표 5-2 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. |

■ 접근 권한의 표기 방법

- 사용자 카테고리별로 누가 파일을 읽고 쓰고 실행할 수 있는지를 문자로 표현한 것
- 읽기 권한은 r, 쓰기 권한은 w, 실행 권한은 x로 나타내며, 해당 권한이 없는 경우에는 -로 표기
- 사용자 카테고리별로 세 가지 권한의 부여 여부를 rwx 세 문자를 묶어서 표기

```
user1@myubuntu:~$ ls -l /etc/hosts
-rw-r--r-- 1 root root 223 11월  8 23:13 /etc/hosts
user1@myubuntu:~$
```

02 파일의 접근 권한

■ 접근 권한의 표기 방법

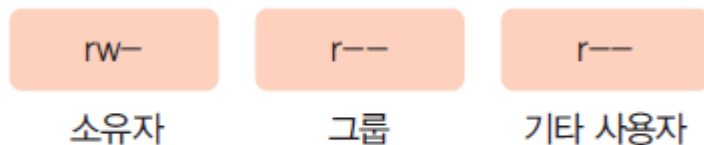


그림 5-2 파일의 접근 권한 표기

표 5-3 다양한 접근 권한 조합의 예

| 접근 권한 | 의미 |
|------------------------|--|
| <code>rwXr-Xr-X</code> | 소유자는 읽기·쓰기·실행 권한을 모두 가지고 그룹과 기타 사용자는 읽기·실행 권한을 가지고 있다. |
| <code>r-Xr-Xr-X</code> | 소유자, 그룹, 기타 사용자 모두 읽기·실행 권한을 가지고 있다. |
| <code>rw-----</code> | 소유자만 읽기·쓰기 권한을 가지고 그룹과 기타 사용자는 아무 권한이 없다. |
| <code>rw-rw-rw-</code> | 소유자, 그룹, 기타 사용자 모두 읽기·쓰기 권한을 가지고 있다. |
| <code>rwXrwxrwx</code> | 소유자, 그룹, 기타 사용자 모두 읽기·쓰기·실행 권한을 가지고 있다. |
| <code>rwX-----</code> | 소유자만 읽기·쓰기·실행 권한을 가지고 그룹과 기타 사용자는 아무 권한이 없다. |
| <code>r-----</code> | 소유자만 읽기 권한을 가지고 있다. |

02 파일의 접근 권한

■ 접근 권한의 변경 명령

chmod

- **기능** 파일이나 디렉터리의 접근 권한을 변경한다.
 - **형식** chmod [옵션] 권한 모드 파일 또는 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

03 기호를 이용한 파일 접근 권한 변경

■ 기호 모드

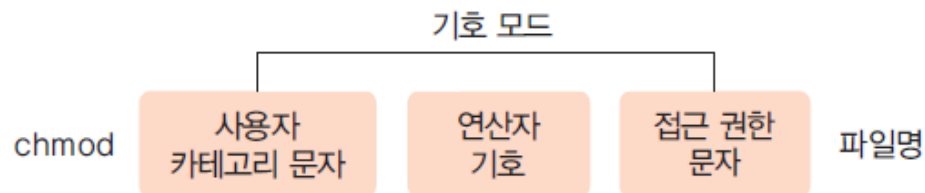


그림 5-3 기호 모드의 구성 요소

표 5-4 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 |

03 기호를 이용한 파일 접근 권한 변경

■ 기호 모드를 사용한 접근 권한 설정의 예

표 5-5 기호 모드를 이용한 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 umask에 따라 권한 부여(+) |
| a+rw | 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) 권한을 부여하고(+) 그룹(g)과 기타 사용자(o)에게 쓰기(w) 권한 부여(+) |

03 기호를 이용한 파일 접근 권한 변경

■ 기호를 이용한 접근 권한 변경 예

- ① 현재 접근 권한 확인: `rw-r--r--`

```
user1@myubuntu:~/linux_ex/ch5$ ls -l
합계 4
-rw-r--r-- 1 user1 user1 223 11월 17 13:25 test.txt
user1@myubuntu:~/linux_ex/ch5$
```

- ② 소유자의 쓰기 권한을 제거: `u-w`

```
user1@myubuntu:~/linux_ex/ch5$ chmod u-w test.txt
user1@myubuntu:~/linux_ex/ch5$ ls -l
합계 4
-r--r--r-- 1 user1 user1 223 11월 17 13:25 test.txt
user1@myubuntu:~/linux_ex/ch5$
```

03 기호를 이용한 파일 접근 권한 변경

■ 실습

- 그룹에 쓰기와 실행 권한을 부여한다 : g+wx
- 기타 사용자에게 실행 권한을 부여한다 : o+x
- 그룹과 기타 사용자의 실행 권한을 제거한다 : go-x
- 모두에게 실행 권한을 부여한다 : a+x
- 소유자에게 쓰기 권한을 부여하고 그룹의 쓰기 권한은 제거한다 : u+w ,g-w

04 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

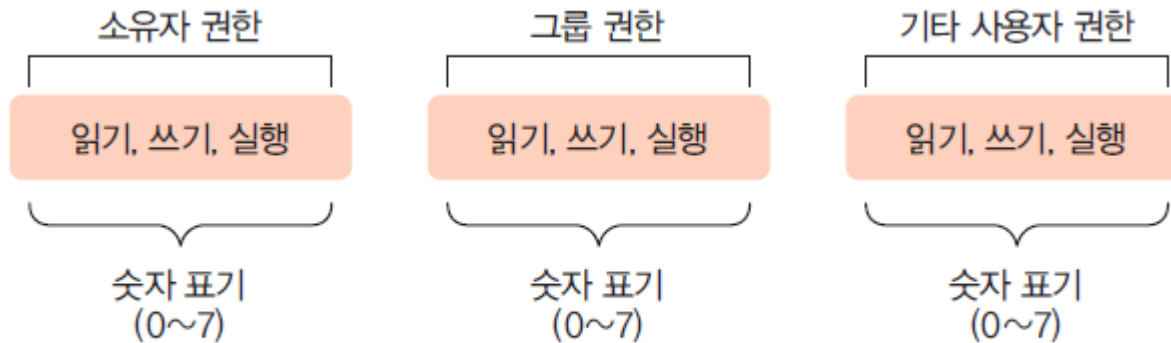


그림 5-4 숫자 모드의 구성 요소

04 숫자를 이용한 파일 접근 권한 변경

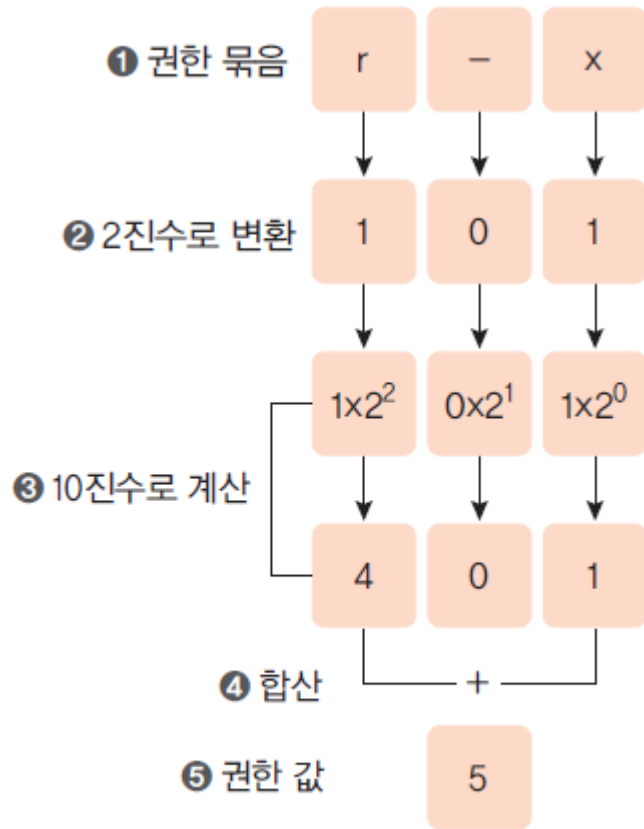


그림 5-5 권한을 숫자로 환산하는 과정

■ 권한을 숫자로 환산하는 과정

- ① r-x라고 할 때 권한이 있는 것은 1로, 없는 것은 0으로 변환
- ② 2진수 1, 0, 1로 변환
- ③ 2진수를 각 자릿수별로 10진수로 환산하면 4, 0, 1이 된다.
- ④ 세 숫자를 더하면
- ⑤ 최종 권한 값은 5가 됨

표 5-6 접근 권한과 숫자의 대응 관계

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

04 숫자를 이용한 파일 접근 권한 변경

■ 전체 권한을 숫자로 표기

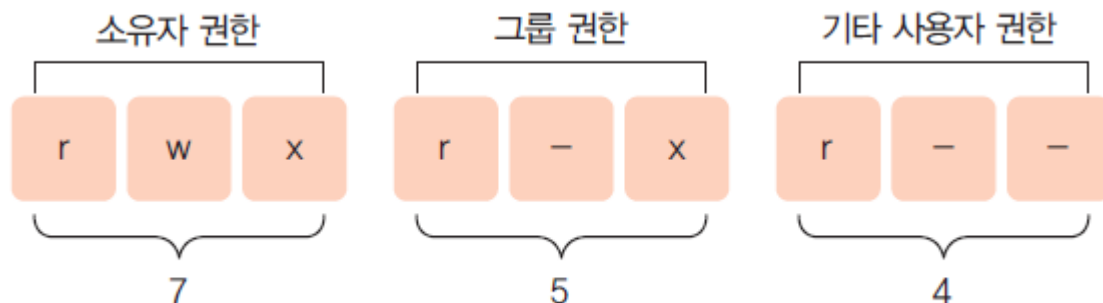


그림 5-6 전체 접근 권한을 숫자로 표기한 예

표 5-7 숫자로 표현한 접근 권한의 예

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

04 숫자를 이용한 파일 접근 권한 변경

■ 숫자 모드로 접근 권한 변경하기

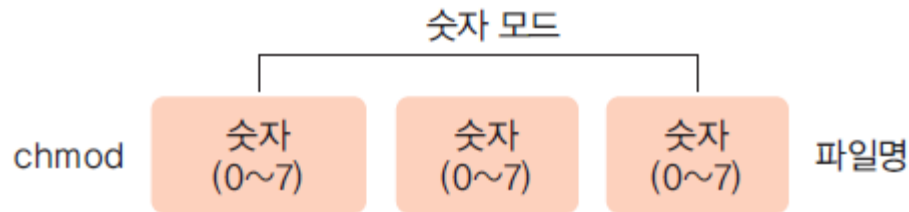


그림 5-7 숫자 모드를 이용한 접근 권한 변경

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

04 숫자를 이용한 파일 접근 권한 변경

■ 숫자 모드로 접근 권한 변경하기 예

① 현재 접근 권한: 644(rw-r--r--)

```
user1@myubuntu:~/linux_ex/ch5$ ls -l
합계 4
-rw-r--r-- 1 user1 user1 223 11월 17 13:25 test.txt
user1@myubuntu:~/linux_ex/ch5$
```

② 소유자의 쓰기 권한을 제거: r--r--r--이므로 444

```
user1@myubuntu:~/linux_ex/ch5$ chmod 444 test.txt
user1@myubuntu:~/linux_ex/ch5$ ls -l
합계 4
-r--r--r-- 1 user1 user1 223 11월 17 13:25 test.txt
user1@myubuntu:~/linux_ex/ch5$
```

③ 그룹에 쓰기와 실행 권한을 부여: r--rwxr--이므로 474

```
user1@myubuntu:~/linux_ex/ch5$ chmod 474 test.txt
user1@myubuntu:~/linux_ex/ch5$ ls -l
합계 4
-r--rwxr-- 1 user1 user1 223 11월 17 13:25 test.txt
user1@myubuntu:~/linux_ex/ch5$
```


04 숫자를 이용한 파일 접근 권한 변경

■ 실습

- 기타 사용자에게 실행 권한을 부여한다 : o+x , r--rwxr-x → 475
- 그룹과 기타 사용자의 실행 권한을 제거한다 : go-x, r--rw-r-- → 464
- 모두에게 실행 권한을 부여한다 : a+x, r-xrwxr-x → 575
- 소유자에게 쓰기 권한을 부여하고 그룹의 쓰기 권한은 제거한다 : u+w,g-w, rwxr-xr-x → 755
- 소유자의 권한만 남기고 나머지 사용자의 권한은 모두 제거: rwx----- → 700

05 기본 접근 권한 설정

■ 기본 접근 권한

- 리눅스에서는 파일이나 디렉토리를 생성할 때 기본 접근 권한이 자동적으로 설정
- 일반 파일의 경우 소유자는 읽기와 쓰기 권한이 설정되고 그룹과 기타 사용자는 읽기 권한만 설정
- 디렉토리의 경우 소유자는 읽기, 쓰기, 실행 권한이 설정되고 그룹과 기타 사용자는 읽기, 실행 권한만 설정

```
user1@myubuntu:~/linux_ex/ch5$ touch ubuntu.txt
user1@myubuntu:~/linux_ex/ch5$ mkdir temp
user1@myubuntu:~/linux_ex/ch5$ ls -l
합계 8
drwxrwxr-x 2 user1 user1 4096 11월 17 13:50 temp
-rwx----- 1 user1 user1 223 11월 17 13:25 test.txt
-rw-rw-r-- 1 user1 user1 0 11월 17 13:50 ubuntu.txt
user1@myubuntu:~/linux_ex/ch5$
```

05 기본 접근 권한 설정

■ 기본 접근 권한 확인하고 변경하기

umask

- **기능** 기본 접근 권한을 출력하거나 변경한다.
- **형식** umask [옵션] [마스크 값]
- **옵션** -S: 마스크 값을 문자로 출력한다.
- **사용 예** umask 022 umask

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

```
user1@myubuntu:~/linux_ex/ch5$ umask
0002
user1@myubuntu:~/linux_ex/ch5$
```

■ 마스크 값의 의미

- 마스크 값은 파일이나 디렉터리 생성 시 부여하지 않을 권한을 지정해놓는 것
- 마스크 값이 022일 경우 이는 ---w--w-이고, 그룹과 기타 사용자에게 쓰기 권한은 부여하지 않겠다는 뜻

```
user1@myubuntu:~/linux_ex/ch5$ umask -S
u=rwx,g=rwx,o=rx
user1@myubuntu:~/linux_ex/ch5$
```

05 기본 접근 권한 설정

■ 마스크 값 변경하기

```
user1@myubuntu:~/linux_ex/ch5$ umask 077
user1@myubuntu:~/linux_ex/ch5$ umask
0077
user1@myubuntu:~/linux_ex/ch5
```

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

```
user1@myubuntu:~/linux_ex/ch5$ touch linux.txt
user1@myubuntu:~/linux_ex/ch5$ ls -l
합계 8
-rw----- 1 user1 user1 0 11월 17 13:57 linux.txt
drwxrwxr-x 2 user1 user1 4096 11월 17 13:50 temp
-rwx----- 1 user1 user1 223 11월 17 13:25 test.txt
-rw-rw-r-- 1 user1 user1 0 11월 17 13:50 ubuntu.txt
user1@myubuntu:~/linux_ex/ch5$
```

05 기본 접근 권한 설정

■ 마스크 값의 적용 과정

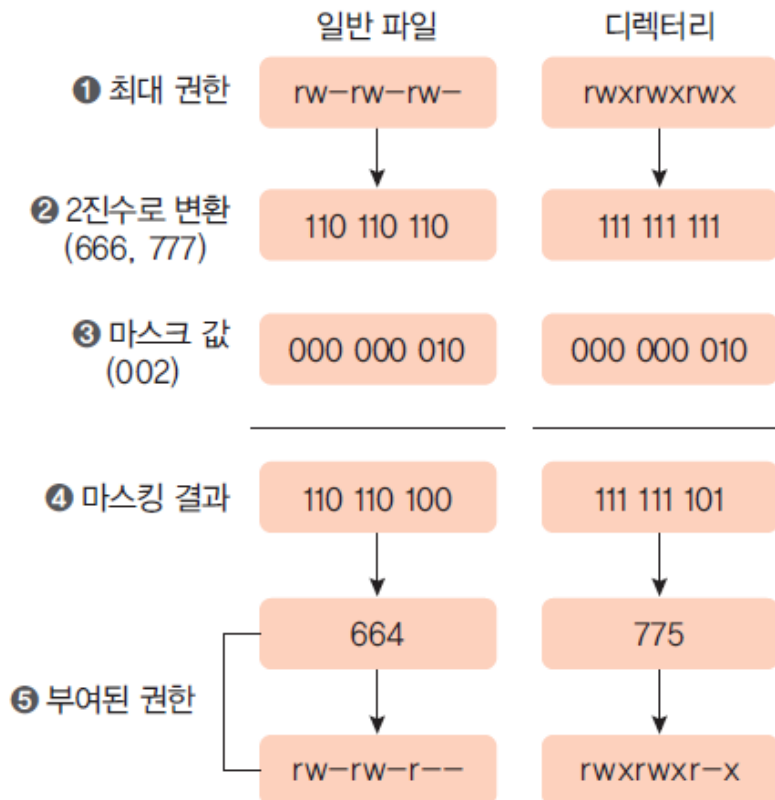


그림 5-8 마스크 값을 적용하는 과정

표 5-8 umask 진리표

| | | | | |
|--------|---|---|---|---|
| 요청 권한 | 1 | 1 | 0 | 0 |
| 마스크 | 1 | 0 | 1 | 0 |
| 부여된 권한 | 0 | 1 | 0 | 0 |

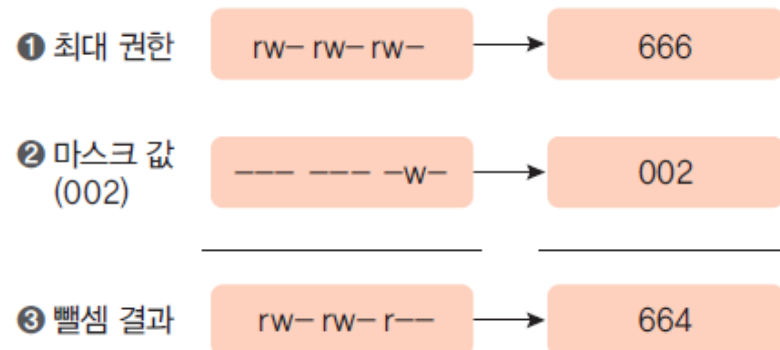


그림 5-9 마스크 값에 뺄셈을 적용하는 과정

05 기본 접근 권한 설정

■ 여러 가지 마스크 값

표 5-9 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 적용해봐야 함
- 마스크 값이 파일에는 적합하지만 디렉터리에는 적합하지 않을 수도 있음

■ 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 설정
- 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

```
user1@myubuntu:~/linux_ex/ch5$ touch set.exe
user1@myubuntu:~/linux_ex/ch5$ chmod 755 set.exe          → 실행 권한을 부여한다.
user1@myubuntu:~/linux_ex/ch5$ ls -l set.exe
-rwxr-xr-x 1 user1 user1 0 11월 17 14:09 set.exe
user1@myubuntu:~/linux_ex/ch5$
user1@myubuntu:~/linux_ex/ch5$ chmod 4755 set.exe
user1@myubuntu:~/linux_ex/ch5$ ls -l set.exe
-rwsr-xr-x 1 user1 user1 0 11월 17 14:09 set.exe
user1@myubuntu:~/linux_ex/ch5$
```

- SetUID가 설정되면 소유자의 실행 권한에 's'가 표시
- set.exe를 실행하면 항상 user1의 권한을 가지고 실행된다는 의미

```
user1@myubuntu:~/linux_ex/ch5$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 54224  8월 21 08:56 /usr/bin/passwd
user1@myubuntu:~/linux_ex/ch5$
```

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

■ 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

```
user1@myubuntu:~/linux_ex/ch5$ ls -ld /tmp
drwxrwxrwt 12 root root 4096 11월 17 13:36 /tmp
user1@myubuntu:~/linux_ex/ch5$
```