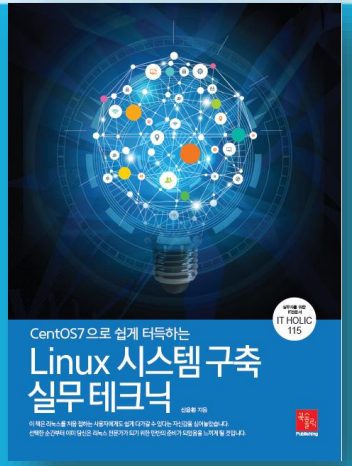
CHAPTER 05

파일 접근권한 설정



파일 속성

- 파일의 속성에 대해 이해하기

1. 파일의 종류에 대해 살펴봅니다.
2. 파일 소유자와 그룹 등 세부적인 파일 속성을 이해합니다.

■ 리눅스 시스템에서 사용되는 파일은 크게 디렉터리와 파일로 구분

■ 다중 시스템으로 접근권한을 부여하여 미인가자가 임의로 파일에 접근하더라도 훼손할 수 없도록 보안기능 제공

파일의 상세정보 출력하기

- 실습을 위해 chap_05 디렉터리를 생성한 다음 [예제 4-5]에서 실습했던 centos_vi.txt 파일을 centos_cp.txt 파일로 복사
- 5장에서 실습하는 모든 파일들은 chap_05 디렉터리에 저장

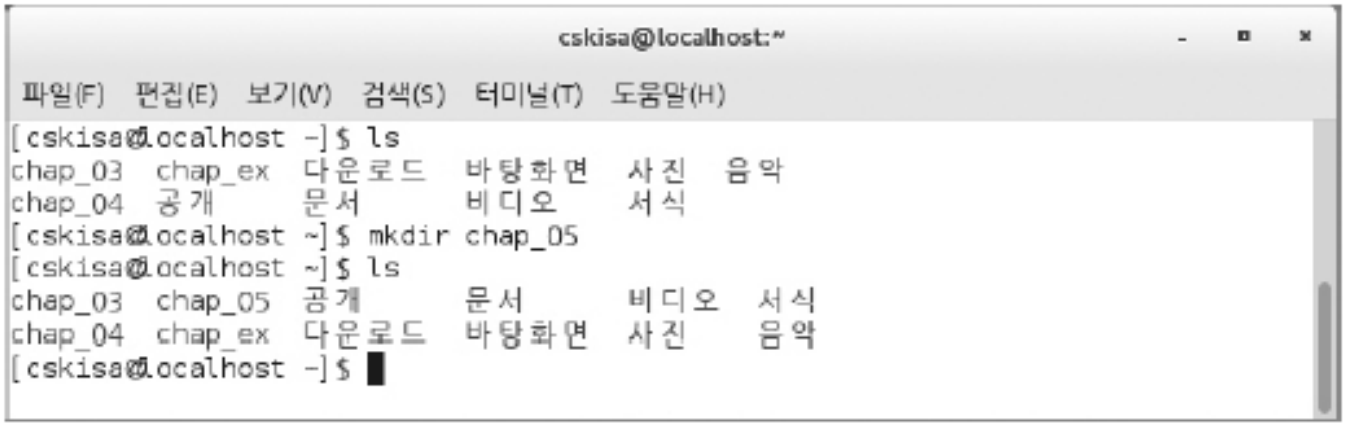
- 파일의 상세정보 출력

| 예제 5-1 |

- Step 01 | 사용자 홈 디렉터리에 chap_05 디렉터리를 새로 생성합니다.

```
$ mkdir chap_05
```

기능 새로운 chap_05 디렉터리 생성
형식 mkdir [새로 만들 디렉터리명 지정] Enter↵

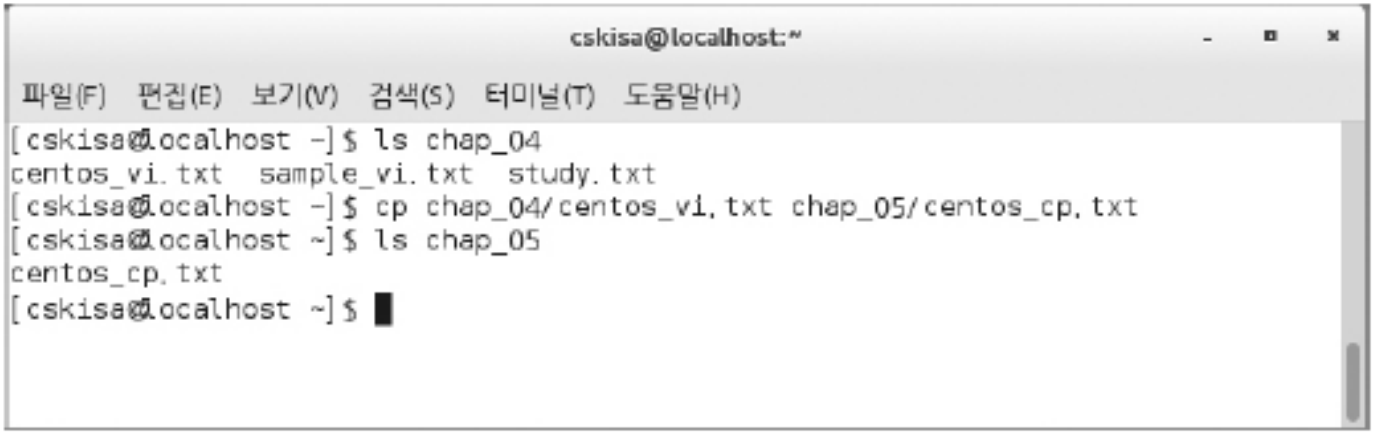


[그림 5-1] 사용자 홈 디렉터리에 chap_05 디렉터리 생성

- **Step 02** | chap_04 디렉터리 안에 있는 centos_vi.txt 파일을 cp 명령으로 chap_05 디렉터리에 centos_cp.txt 파일명으로 복사하고 ls 명령으로 chap_05 디렉터리에 복사한 centos_cp.txt 파일이 존재하는지를 확인합니다.

```
$ cp chap_04/centos_vi.txt chap_05/centos_cp.txt
```

기능 기존에 존재하는 centos_vi.txt 파일을 centos_cp.txt 파일로 복사
형식 cp [원본 파일명] [복사할 파일명] Enter↵

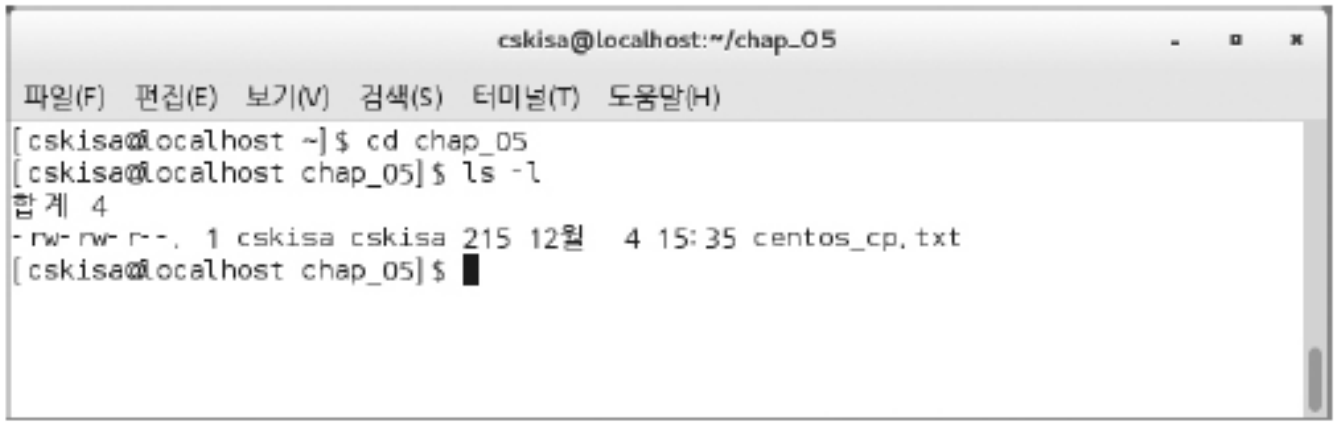


[그림 5-2] centos_vi.txt 파일을 centos_cp.txt 파일명으로 복사

- **Step 03** | 사용자 홈 디렉터리에서 chap_05 디렉터리로 이동한 다음 앞에서 복사한 파일 centos_cp.txt에 대한 상세한 정보를 출력하기 위해 ls -l 명령을 수행합니다.

```
$ ls -l
```

기능 파일의 정보를 상세하게 출력
형식 ls [옵션] Enter↵



[그림 5-3] centos_cp.txt 파일의 상세정보 출력

[그림 5-3]에서 보는 바와 같이 centos_cp.txt 파일에 대한 상세한 정보가 출력되었습니다. 파일을 마지막으로 수정한 날짜에는 연도가 생략되어 있는데 이는 파일이 최종적으로 금년에 수정되었음을 의미합니다. 출력된 파일의 정보에 대해 각 항목별로 상세하게 살펴보도록 하겠습니다.

```
-   rw- rw- r--  .  1   cskisa   cskisa   215   12월  4  15:35   centos_cp.txt
❶       ❷          ❸     ❹       ❺       ❻          ❼              ❽
```

이와 같은 파일의 상세한 정보에 대해 항목별로 [표 5-1]과 같이 정리하였습니다.

[표 5-1] 파일 속성에 대한 상세 정보

번호	속성 값	의미
❶	-	파일의 종류 (-는 일반파일, d는 디렉터리)
❷	rw- rw- r--	파일을 읽고(rw-) 쓰고(rw-) 실행(r--)할 수 있는 접근권한
❸	1	하드 링크의 수
❹	cskisa	파일 소유자의 로그인 ID
❺	cskisa	파일 소유자의 그룹 이름
❻	215	파일의 크기 (byte 단위)
❼	12월 4 15:35	파일을 최종 수정한 일자와 시각
❽	centos_cp.txt	파일명

① 파일의 종류

➢ 파일 속성의 첫 번째 항목은 파일의 종류를 표시

- \- : 일반 파일
- d : 디렉터리

➢ 파일의 종류를 확인하기 위한 명령

$ file

기능 지정한 파일의 종류를 알려줌
형식 file [디렉터리명 또는 파일명] Enter↵

➢ 파일의 종류를 파악하기 전에 현재 디렉터리의 위치 확인

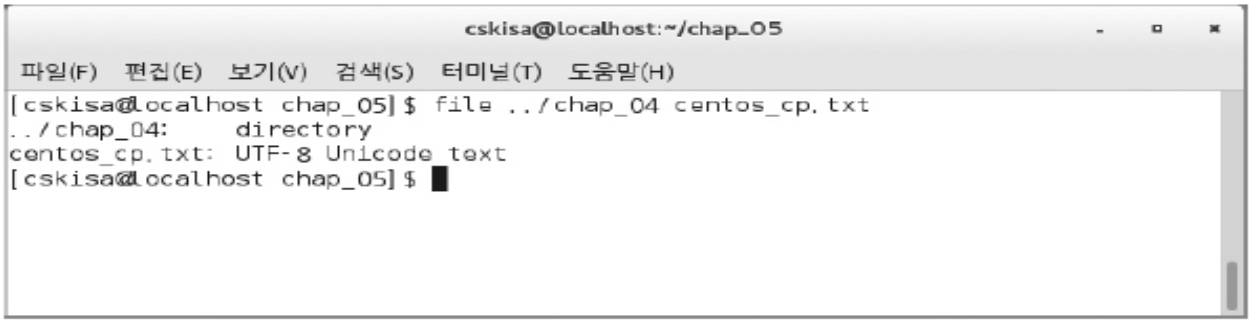
- 현재 디렉터리의 위치 : chap_05
- 상위 디렉터리로 이동 : ../

| 예제 5-2 |

현재 디렉터리의 위치에서 chap_04 디렉터리에 존재하는 centos_cp.txt 파일의 종류를 출력합니다.

```
$ file ../chap_04 centos_cp.txt
```

기능 chap_04와 centos_cp.txt 파일의 종류 출력
형식 file [디렉터리명] [파일명] Enter↵



[그림 5-4] chap_04와 centos_cp.txt 파일의 종류 출력

② 파일의 접근권한 표시

- 파일의 속성에서 두 번째 항목인 rw- rw- r—의미하며 파일에 접근하여 읽고 쓰는 권한을 부여하기 위함

  - rw- : 소유자 접근권한
  - rw- : 그룹 접근권한
  - r-- : 기타 사용자 접근권한

③ 하드 링크의 수

- 하드 링크는 하나의 파일에 대해 여러 개의 파일명을 부여할 수 있는 기능
- 3장에서 이미 살펴봤으므로 자세한 사항은 3장을 참조

④ 파일 소유자의 로그인 ID

- 리눅스 시스템에서 취급되는 모든 파일에는 각각의 소유자가 지정되어 있음
- 시스템과 관련된 파일은 대부분 루트계정이 소유자이고 일반 파일은 해달 파일을 생성한 사용자가 파일의 소유자가 됨
- Centos_cp.txt 파일의 경우 사용자 계정 cskisa 아이디로 접속된 상태에서 생성한 파일이므로 파일 소유자의 로그인 ID는 cskisa로 출력됨

⑤ 파일 소유자의 그룹이름

- 시스템 관리자가 사용자를 등록할 때 파일 소유자의 그룹이 결정
- 임의로 소유자가 속할 그룹을 바꿀 수는 없으며 변경하고자 할 경우에는 반드시 시스템 관리자에게 그룹 변경을 요청해야 함
- 그룹 변경에 대해 정의는 /etc/group 디렉터리에서 관리

- 현재 접속 중인 사용자가 속한 그룹 조회

```
$ groups
```

기능 현재 접속 중인 사용자가 속한 그룹을 알려줌

형식 groups [사용자명] Enter↵

- Groups 명령을 사용할 때 '사용자명'을 인자로 지정하면 그 사용자가 속한 그룹을 알려주지만
- 인자를 지정하지 않으면 현재 접속 중인 사용자가 속한 그룹을 알려줌

● 현재 접속 중인 사용자가 속한 그룹 조회

| 예제 5-3 |

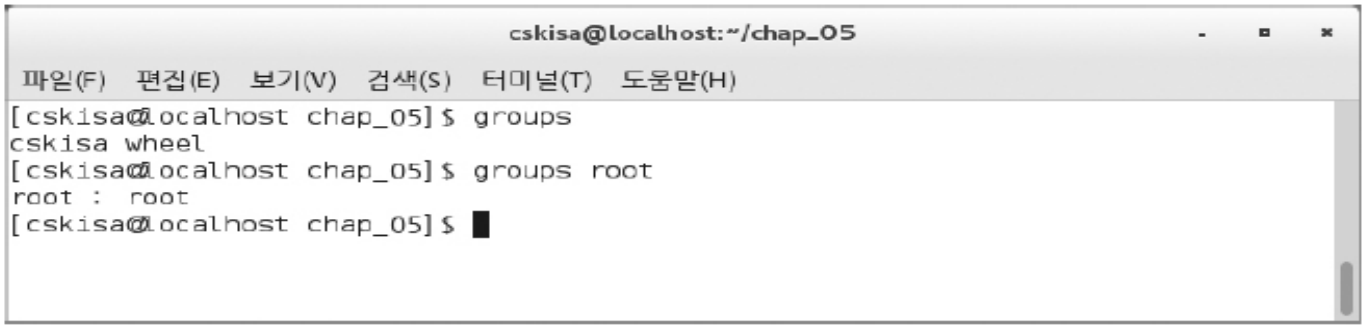
현재 접속 중인 사용자가 속한 그룹과 root 계정이 속한 그룹에 대한 정보를 출력합니다.

$ groups

기능 현재 접속 중인 사용자가 속한 그룹을 알려줌
형식 groups [사용자명] Enter↵

$ groups root

기능 현재 접속 중인 사용자가 속한 그룹을 알려줌
형식 groups [사용자명] Enter↵



[그림 5-5] 사용자와 root가 속한 그룹 확인

⑥ 파일의 크기

- 파일의 크기는 byte 단위로 알려줌
- [예제 5-1]의 출력결과에서 centos_cp.txt 파일의 크기가 215 byte로 출력

⑦ 파일이 수정된 마지막 일자와 시각

- 파일이 마지막으로 수정된 일자와 시각 표시
- 연도가 표시되지 않는 경우는 금년을 의미

⑧ 파일명

- 실질적으로 사용되고 있는 파일명
- 파일의 종류에 따라 디렉터리 또는 파일의 이름으로 출력

파일 접근권한

- 파일 접근권한에 대해 이해하기

1. 파일 접근권한의 종류에 대해서 살펴봅니다.
2. 파일 접근권한 표기방법에 대해 이해합니다.
3. 파일 접근권한 변경방법을 예제를 통해 실습합니다.

접근권한 종류

- 리눅스 시스템에서 사용되는 파일에 대한 접근권한은 읽기, 쓰기, 실행권한 등 3가지 권한으로 제한

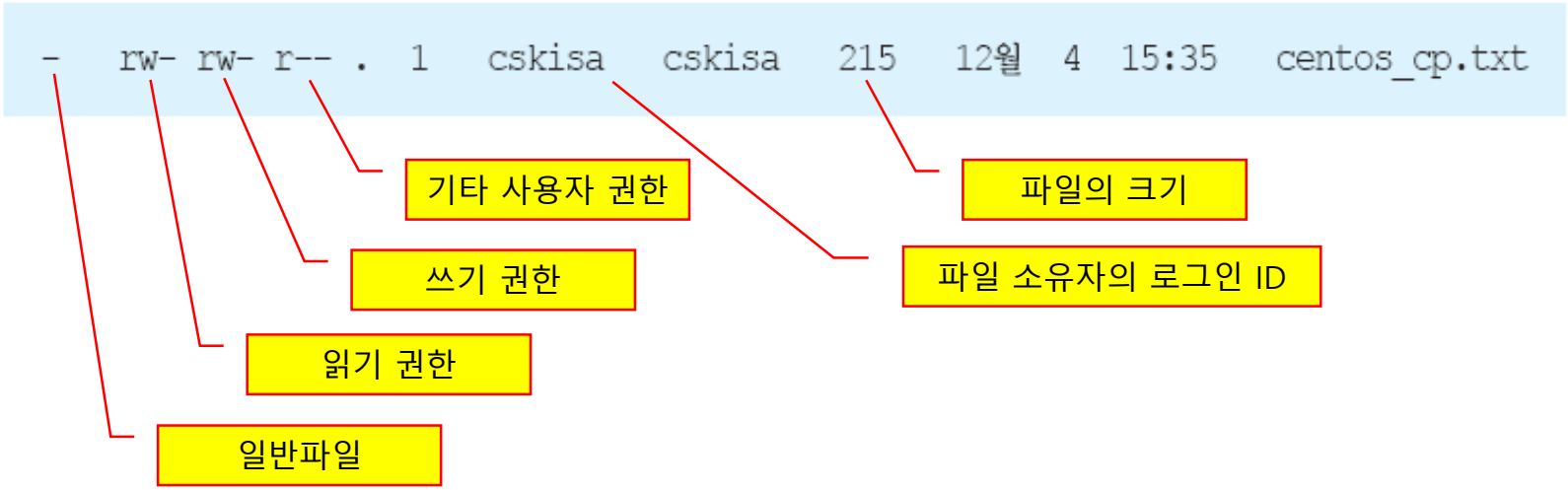
[표 5-2] 파일과 디렉터리 접근권한

접근권한	파일	디렉터리
읽기(r)	해당 파일을 읽거나 복사 가능	ls 명령으로 디렉터리의 목록을 확인 (옵션은 실행권한이 있을 경우만 허용)
쓰기(w)	파일을 수정, 이동, 삭제 가능 (디렉터리에 쓰기권한 있어야함)	해당 파일을 삭제하거나 생성할 수 있음
실행(x)	셸 스크립트 또는 실행파일에 대한 실행 가능	cd 명령 사용 가능하며 파일을 디렉터리로 이동 또는 복사 가능

접근권한 표기방법

- 리눅스 시스템에 접속한 사용자 카테고리별로 누가 어떤 파일에 대해 읽기, 쓰기, 실행권한을 할 수 있는지에 대해 문자로 표현

- [예제 5-1]에서 실습한 결과 centos_cp.txt 파일의 접근권한 출력



접근권한 표기방법

- 다양한 파일 접근권한의 조합

[표 5-3] 다양한 파일 접근권한 조합

접근권한	의미
rwx r-x r-x	소유자 권한(읽기, 쓰기, 실행), 그룹과 기타 사용자 권한(읽기, 실행)
r-x r-x r-x	소유자, 그룹, 기타 사용자(읽기, 실행) 권한만 부여
rw- --- ---	소유자 (읽기, 쓰기) 권한만 부여 그룹과 기타 사용자는 권한 없음
rw- rw- rw-	소유자, 그룹, 기타 사용자 모두(읽기, 쓰기) 권한만 부여
rwx rwx rwx	소유자, 그룹, 기타 사용자 모두(읽기, 쓰기, 실행) 권한 부여
rwx --- ---	소유자 (읽기, 쓰기, 실행) 권한만 부여
r-- --- ---	소유자 (읽기) 권한만 부여

접근권한 변경 명령 : chmod

- 접근권한 변경은 파일 소유자와 시스템 관리자만 변경 가능하며 파일 접근권한 변경 명령은 change mode의 약어인 chmod 명령을 사용

$ chmod

기능 디렉터리 또는 파일의 접근권한 변경
형식 chmod [옵션] Enter↵
옵션 -R : 하위 디렉터리까지 접근권한을 모두 변경

기호를 이용한 접근권한

- 파일 접근권한 변경을 위해 심볼릭 모드 사용 이해하기

1. 파일 접근권한을 변경하는 심볼릭 모드에 대해 이해합니다.
2. 심볼릭 모드에서 사용되는 문자와 기호를 살펴봅니다.
3. 심볼릭 모드로 파일 접근권한을 변경하는 방법에 대해 실습합니다.

심볼릭 모드

- 기호를 이용한 접근권한 변경방법인 심볼릭 모드는 사용자 카테고리 문자, 연산자 기호, 접근권한 문자를 사용하여 파일 접근권한 변경

- 심볼릭 모드에서 사용되는 문자와 기호

[표 5-4] 심볼릭 모드에서 사용되는 문자와 기호의 종류

구분	문자/기호	의미
사용자 카테고리 문자	u	파일 소유자
	g	파일 소유자가 속한 그룹
	o	파일 소유자와 그룹 이외의 기타 사용자
	a	파일을 사용하려는 전체 사용자
연산자 기호	+	파일 접근권한 부여
	-	파일 접근권한 제거
	=	파일 접근권한 설정
접근권한 문자	r	파일 읽기권한
	w	파일 쓰기권한
	x	파일 실행권한

- 심볼릭 모드에서 다양한 파일 접근권한

[표 5-5] 심볼릭 모드에서 다양한 파일 접근권한 조합

권한 표기	의미
u+w	파일 소유자에게 쓰기(w) 권한부여
u-w	파일 소유자에게 쓰기(w) 권한제거
u=rwx	파일 소유자에게 쓰기(w), 읽기(r), 실행(x) 권한설정
u+x,go+w	소유자에게 실행(x) 권한부여와 그룹 및 기타 사용자에게 쓰기(w) 권한부여
g+w	파일 그룹에게 쓰기(w) 권한부여
g+wx	파일 그룹에게 쓰기(w)와 실행(x) 권한부여
go+w	그룹과 기타 사용자에게 쓰기(w) 권한부여
+wx	파일을 사용하려는 모든 사용자에게 쓰기(w)와 실행(x) 권한부여
a+rwx	모든 사용자에게 쓰기(w), 읽기(r), 실행(x) 권한부여
o-r	파일 기타 사용자에게 읽기(r) 권한제거

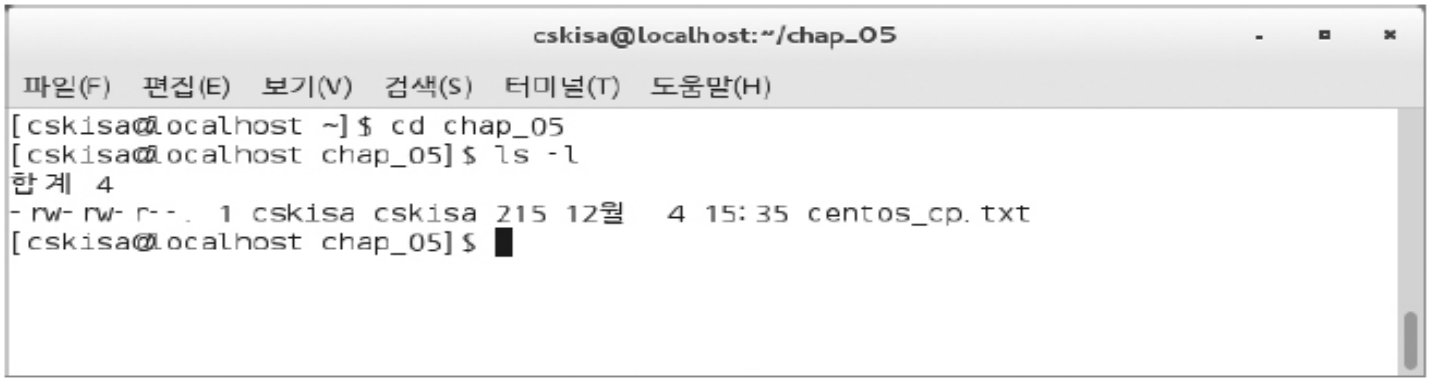
심볼릭 모드로 접근권한 변경

예제 5-4

- **Step 01** | 터미널 창에서 현재 디렉터리의 위치를 확인하여 chap_05 디렉터리 안으로 이동한 다음 디렉터리에 존재하는 centos_cp.txt 파일에 대해 ls -l 명령으로 파일 접근권한이 현재 어떻게 설정되어 있는지에 대한 상세정보를 출력합니다.

```
$ ls -l
```

기능 파일의 정보를 상세하게 출력
형식 ls [옵션] Enter↵



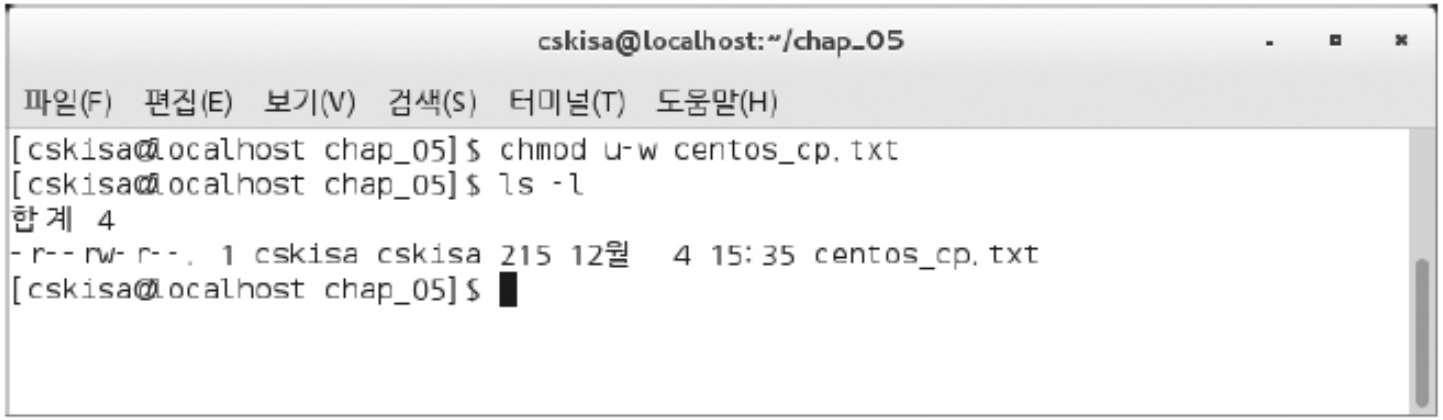
[그림 5-8] centos_cp.txt 파일의 상세정보 출력

- **Step 02** | centos_cp.txt 파일의 현재 접근권한은 rw- rw- r--으로 설정되어 있습니다. 여기에서 접근권한 첫 번째 항목인 소유자 권한 중 쓰기 권한을 제거하기 위해 문자와 연산자 기호를 u-w와 같이 선언하여 접근권한을 변경합니다.

```
$ chmod u-w centos_cp.txt
```

기능 파일의 접근권한 변경

형식 chmod [심볼릭 모드] [파일명] Enter↵



[그림 5-9] centos_cp.txt 파일 접근권한 변경

- **Step 03** | centos_cp.txt 파일에 설정되어 있던 접근권한은 rw- rw- r--에서 소유자 쓰기 권한제거 옵션 u-w를 통해 r-- rw- r--으로 접근권한이 변경 설정되었음을 확인합니다.

<접근권한 변경 전>

```
-   rw- rw- r-- .  1   cskisa   cskisa   215   12월  4  15:35   centos_cp.txt
```

▲ 당초

<접근권한 변경 후>

```
-   r-- rw- r-- .  1   cskisa   cskisa   215   12□  4  15:35   centos_cp.txt
```

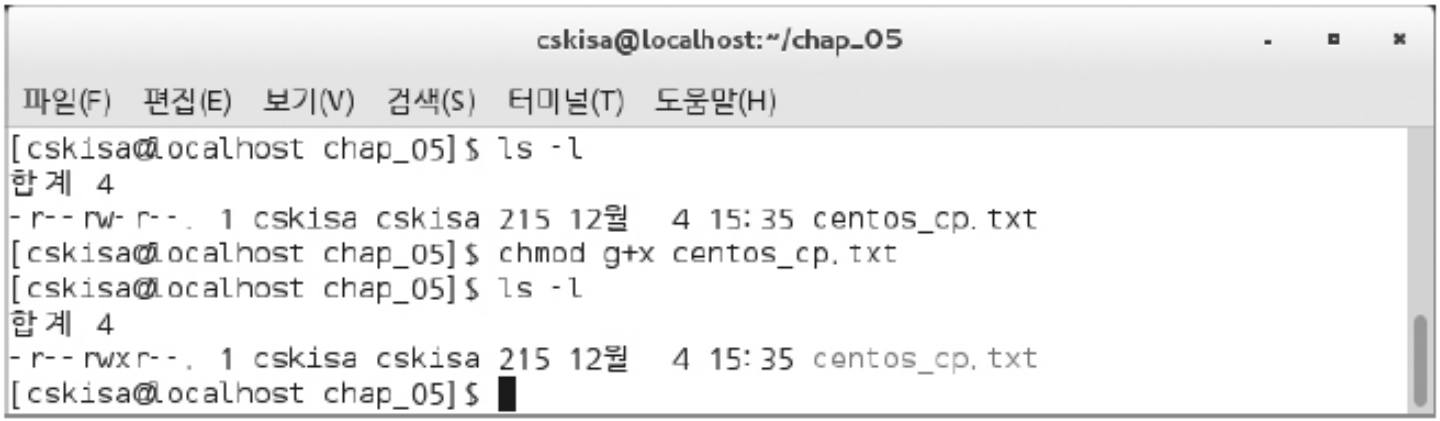
▲ 변경

- **Step 04** | centos_cp.txt 파일의 그룹에 심볼릭 모드 g+x를 지정하여 그룹에 대한 실행권한을 부여합니다.

```
$ chmod g+x centos_cp.txt
```

기능 파일의 접근권한 변경

형식 chmod [심볼릭 모드] [파일명] Enter↵

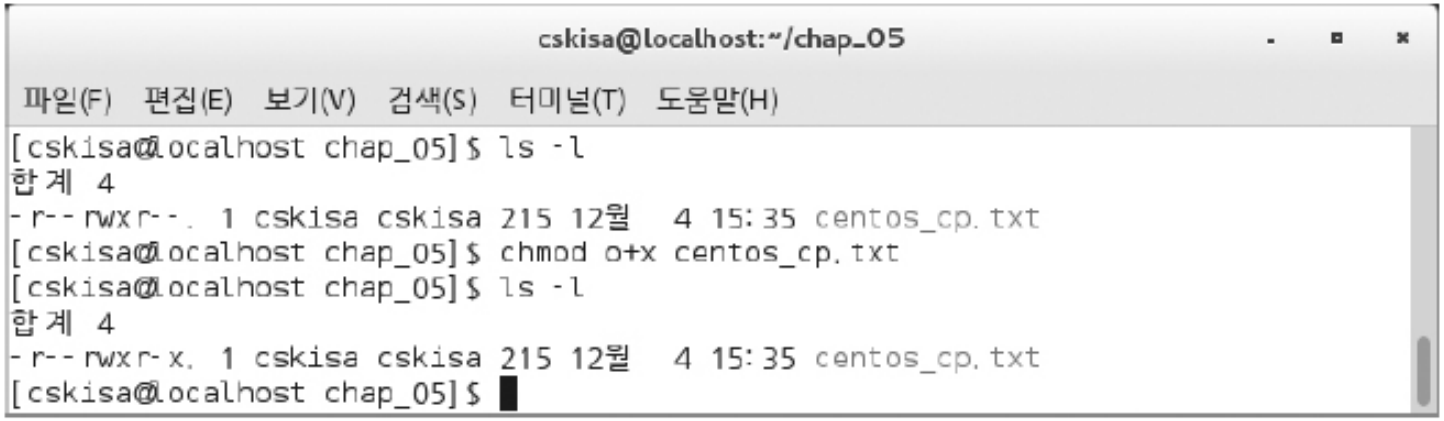


[그림 5-10] 그룹에 실행권한 부여

- **Step 05** | 이 단계에서는 심볼릭 모드 o+x를 지정하여 기타 사용자에게 파일에 대한 실행권한을 부여합니다.

```
$ chmod o+x centos_cp.txt
```

기능 파일의 접근권한 변경
형식 chmod [심볼릭 모드] [파일명] Enter↵



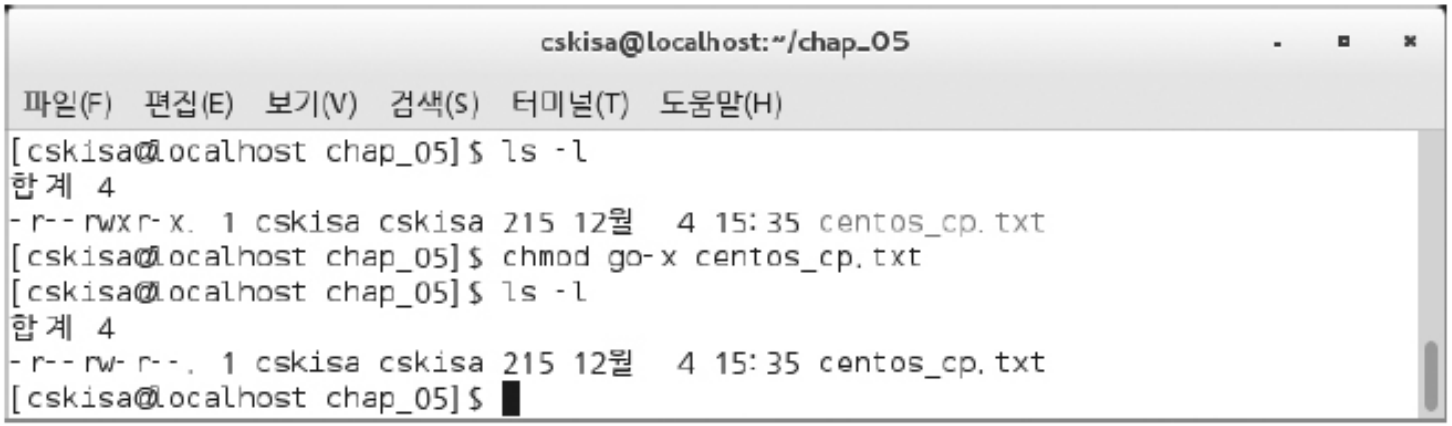
[그림 5-11] 기타 사용자에게 실행권한 부여

- **Step 06** | 다음은 centos_cp.txt 파일에 대한 그룹과 기타 사용자의 파일 실행권한을 심볼릭 모드 go-x를 지정하여 제거합니다.

```
$ chmod go-x centos_cp.txt
```

기능 파일의 접근권한 변경

형식 chmod [심볼릭 모드] [파일명] Enter↵



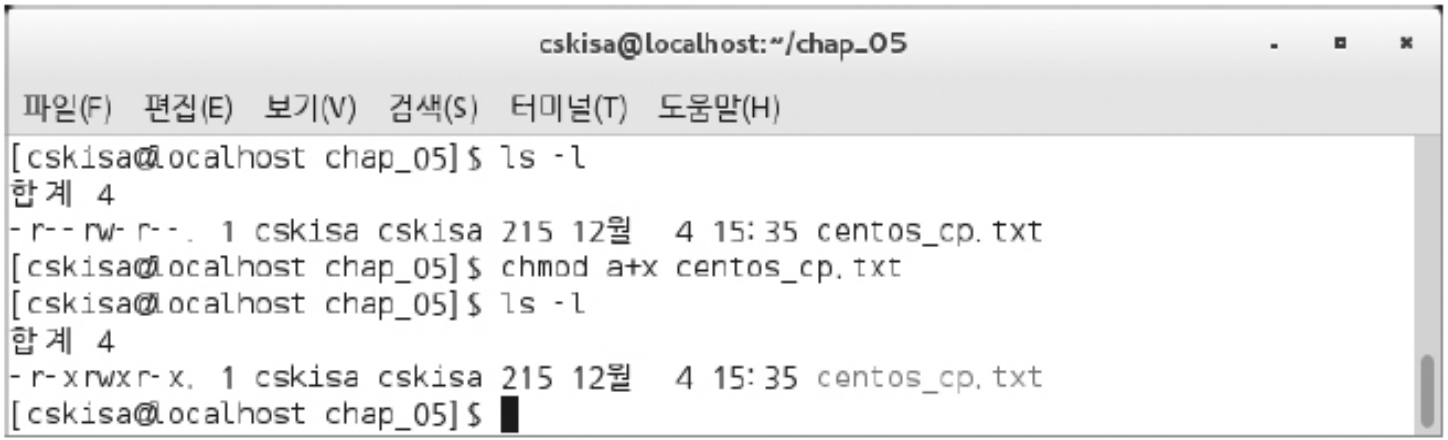
[그림 5-12] 그룹과 기타 사용자의 실행권한 제거

- **Step 07** | 이 단계에서는 심볼릭 모드 a+x를 지정하여 centos_cp.txt 파일을 모든 사용자에게 실행권한을 부여합니다.

```
$ chmod a+x centos_cp.txt
```

기능 파일의 접근권한 변경

형식 chmod [심볼릭 모드] [파일명] Enter↵



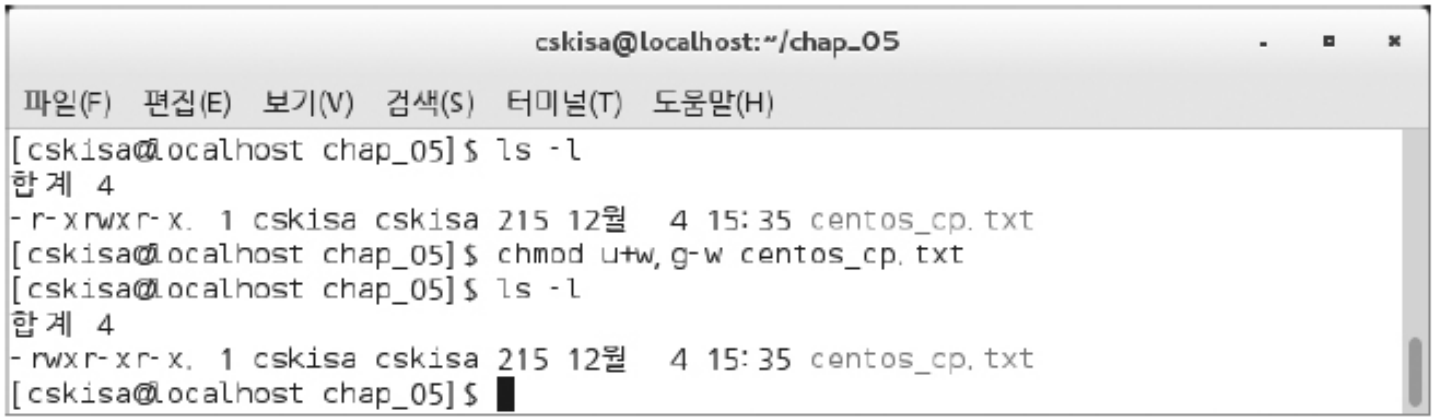
[그림 5-13] 모두에게 실행권한 부여

- **Step 08** | 마지막 단계로 centos_cp.txt 파일 소유자에게 u+w를 지정하여 파일에 대한 쓰기권한을 부여하고 그룹의 쓰기권한은 g-w를 지정하여 파일에 대한 쓰기권한을 각각 제거합니다.

```
$ chmod u+w,g-w centos_cp.txt
```

기능 파일의 접근권한 변경

형식 chmod [심볼릭 모드],[심볼릭 모드] [파일명] Enter↵



[그림 5-14] 소유자 쓰기권한 부여와 그룹의 쓰기권한 제거

실습 5-1 다음 항목에서 주어진 지시사항을 수행하시오.

1. 주어진 문장을 gedit 창에서 입력하기
2. 입력한 문장을 story.txt 파일명으로 저장하기
3. story.txt 파일의 현재 설정된 접근권한 출력하기
4. story.txt 파일의 접근권한을 심볼릭 모드로 파일 소유자에게 실행권한과 그룹 및 사용자에게 쓰기 권한 부여하기
5. story.txt 파일에 변경된 접근권한 출력하기
6. story.txt 파일의 접근권한을 심볼릭 모드로 파일 소유자에게 실행권한과 그룹 및 사용자에게 쓰기 권한 제거하기
7. story.txt 파일에 변경된 접근권한 출력하기

▼ 주어진 문장

리눅스 운영체제는
컴퓨터를 사용함에 있어
많은 편의성과 기능을 제공해 줍니다.

$ gedit

$ 지에디트 창에서 위 문장을 입력 후 story.txt 파일명으로 저장

$ ls –l

$ chmod u+x,go+w story.txt

$ ls –l

$ chmod u-x,go-w story.txt

$ ls –l

숫자를 이용한 접근권한

- 파일 접근권한 변경을 위한 숫자모드 사용 이해하기

1. 파일 접근권한을 변경하는 숫자 모드에 대해 살펴봅니다.
2. 숫자 모드에서 숫자로 환산하는 방법에 대해 이해합니다.
3. 숫자 모드로 파일 접근권한을 변경하는 방법에 대해 실습합니다.

숫자 모드

- 접근권한을 조정할 때 문자의 조합은 다소 많아지므로 복잡해지는 단점을 해결하기 좋은 방법인 숫자 모드 활용

- centos_cp.txt 파일의 접근권한을 숫자모드로 표현

```
-   rwx r-x r-x .  1   cskisa   cskisa   215   12월  4  15:35   centos_cp.txt
```

- 파일 접근권한과 숫자의 대응

[표 5-6] 파일 접근권한과 숫자의 대응관계

권한 표기	2진수	8진수	의미	접근권한 예시
rwx	111	7 (4+2+1)	읽기, 쓰기, 실행	rwx rwx rwx → 777
rw-	110	6 (4+2+0)	읽기, 쓰기	rwx r-x r-x → 755
r-w	101	5 (4+0+1)	읽기, 실행	rw- rw- rw- → 666
r--	100	4 (4+0+0)	읽기	r-x r-x r-x → 555
-wx	011	3 (0+2+1)	쓰기, 실행	rw- r-- r-- → 644
-w-	010	2 (0+2+0)	쓰기	rwx --- --- → 700
--x	001	1 (0+0+1)	실행	rw- r-- --- → 640
---	000	0 (0+0+0)	권한 없음	r-- --- --- → 400

실습 5-2 다음 항목에서 주어진 지시사항을 수행하시오.

1. 특정 파일에 주어진 권한 -rwx r--w rwx 설명하기
2. 소유자 권한에 대한 권한값 숫자모드로 환산하기
3. 그룹 권한에 대한 권한값 숫자모드로 환산하기
4. 기타 사용자 권한에 대한 권한값 숫자모드로 환산하기
5. 권한묶음에 대한 권한값 숫자모드로 최종 환산하기

일반 파일 : -

소유자 권한 : 읽기, 쓰기, 실행 권한부여 (rwx)

그룹 권한 : 읽기와 쓰기 권한부여 (r-w)

기타 사용자 권한 : 읽기, 쓰기, 실행 권한부여 (rwx)

소유자 권한값 : 400 + 200 + 100 = 700

그룹 권한값 : 040 + 000 + 010 = 050

기타 사용자 권한 : 004 + 002 + 001 = 006

권한묶음에 대한 최종 권한값 : 700 + 050 + 006 = 756

숫자 모드로 접근권한 변경

- 숫자의 각 위치가 사용자 카테고리를 나타내고 있으므로 사용자 카테고리를 별도로 지정할 필요 없음(기호 모드와 차이점)

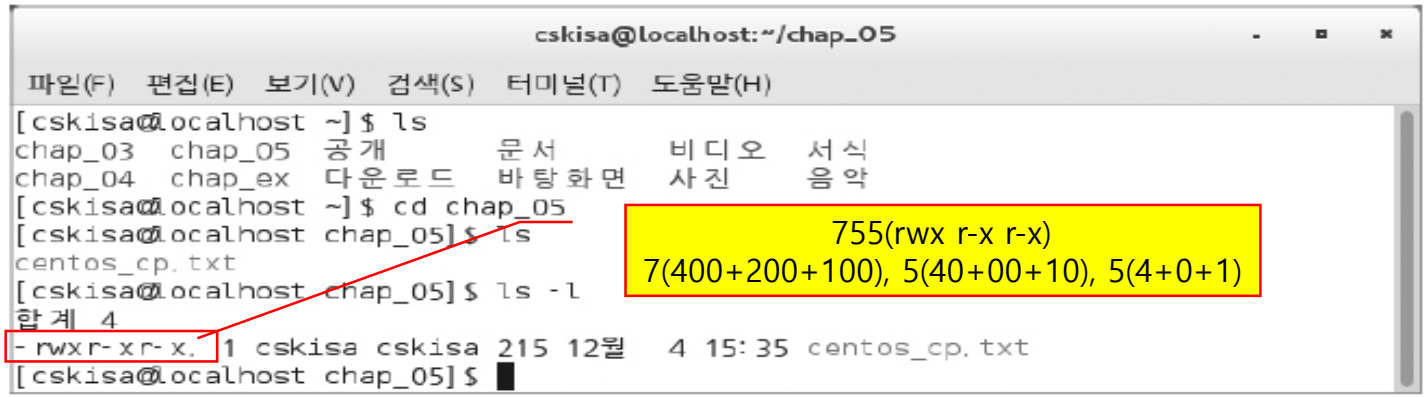
● 숫자 모드로 접근권한 변경

| 예제 5-5 |

● **Step 01** | 터미널 창에서 chap_05 디렉터리 안으로 이동한 다음 디렉터리에 존재하는 centos_cp.txt 파일에 대해 ls -l 명령으로 파일 접근권한이 설정되어 있는 현재 상태를 확인합니다.

$ ls -l

기능 파일의 정보를 상세하게 출력
형식 ls [옵션] Enter↵

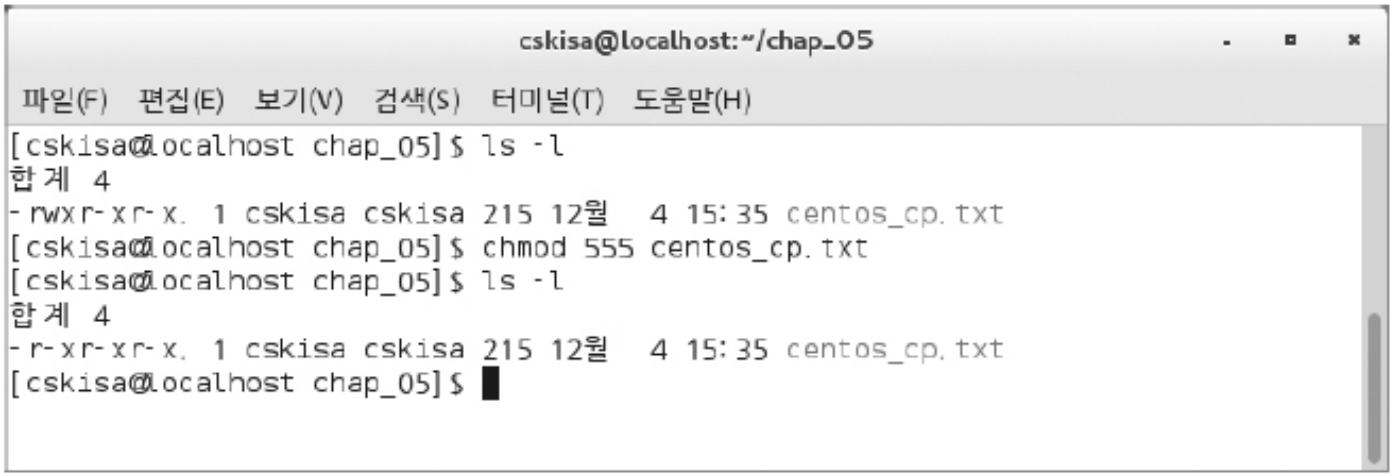


[그림 5-18] centos_cp.vi 파일의 접근권한 현재 상태 확인

- **Step 02** | centos_cp.txt 파일에 부여된 접근권한 rwx r-x r-x에서 소유자의 접근권한에서 쓰기권한을 제거하기 위해 숫자 모드 555(r-x r-x r-x)로 접근권한을 변경하면 다음과 같은 결과가 출력됩니다.

```
$ chmod 555 centos_cp.txt
```

기능 파일의 접근권한 변경
형식 chmod [숫자 모드] [파일명] Enter↵

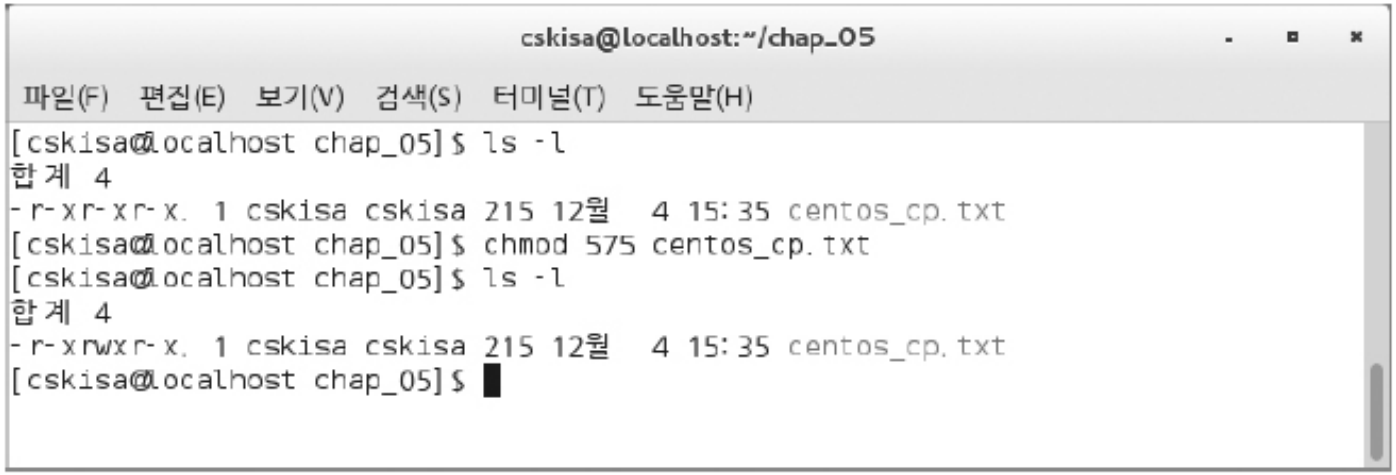


[그림 5-19] centos_cp.txt 파일의 접근권한 555로 변경

- **Step 03** | 이제 centos_cp.txt 파일에 변경된 접근권한 r-x r-x r-x에서 그룹의 접근권한에서 쓰기권한을 부여하기 위해 숫자 모드 575(r-x rwx r-x)로 접근권한을 변경해 봅니다.

```
$ chmod 575 centos_cp.txt
```

기능 파일의 접근권한 변경
형식 chmod [숫자 모드] [파일명] Enter↵



[그림 5-20] centos_cp.txt 파일의 접근권한 575로 변경

실습 5-3 다음 항목에서 주어진 지시사항을 수행하시오.

1. 새로운 빈파일 yesterday.txt 생성하기
2. yesterday.txt 파일에 설정되어 있는 접근권한 출력하기
3. yesterday.txt 파일에 대한 접근권한을 숫자모드를 이용하여 rwx rwx r-x로 변경하기
4. 접근권한이 변경된 yesterday.txt 파일의 접근권한 출력하기
5. yesterday.txt 파일삭제 여부를 물어가며 삭제하기
6. 삭제된 yesterday.txt 파일이 존재하는지 확인하기

```
$ touch yesterdat.txt
$ ls –l
$ chmod 775 yesterday.txt
$ ls –l
$ rm –i yesterday.txt
$ ls -l
```

기본 접근권한

- 파일 생성 시 부여되는 기본 접근권한 이해하기

1. 파일 생성 시 부여되는 접근권한에 대해 살펴봅니다.
2. 파일에 부여하지 않을 권한을 지정할 때의 마스크 값에 대해 이해합니다.
3. 마스크 값을 적용하여 기본 접근권한을 변경하는 방법에 대해 실습합니다.

기본 접근권한 확인

- 파일과 디렉터리에 대한 기본 접근권한은 리눅스 시스템에 설정된 기본 값에 따라 자동으로 설정됨
- 일반 파일인 경우에는 소유자와 그룹의 접근권한은 읽기(r)와 쓰기 권한(w)만 설정되고
- 디렉터리의 경우에는 소유자와 그룹은 읽기, 쓰기, 실행권한이 설정되며 기타 사용자는 읽기와 실행권한만 설정됨

- work 디렉터리와 sample_f.txt 일반파일을 생성한 다음 디렉터리와 일반 파일에 설정된 기본 접근권한 확인

| 예제 5-6 |

- **Step 01** | 현재 디렉터리에 빈파일 sample_f.txt를 생성합니다.

```
$ touch sample_f.txt
```

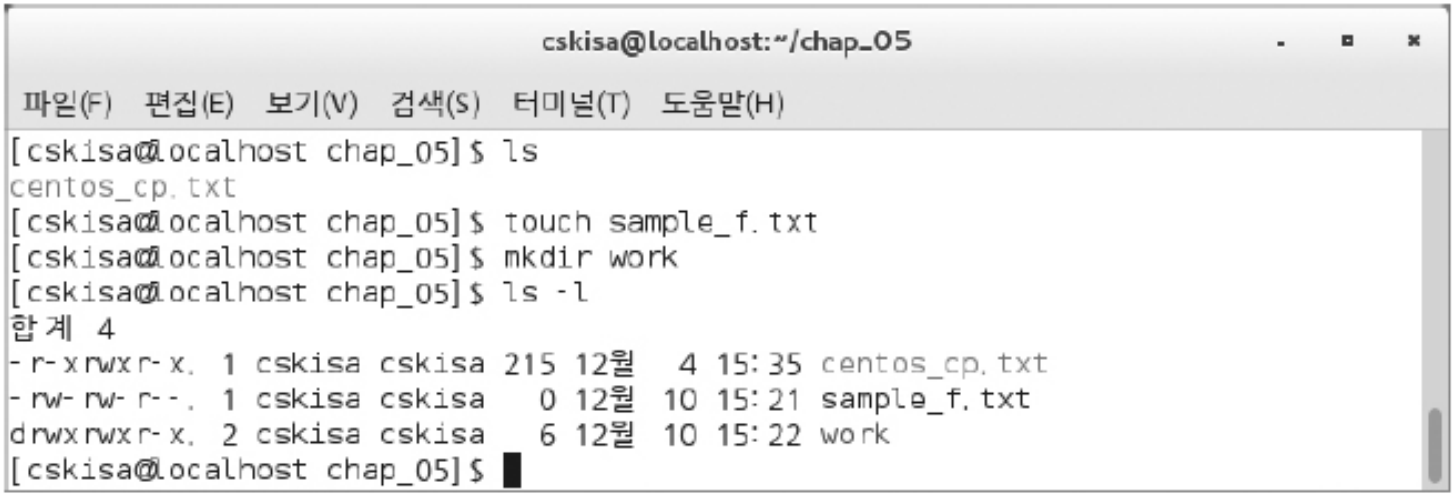
기능 빈파일 생성
형식 touch [일반 파일명] Enter↵

- **Step 02** | 현재 디렉터리에 새로운 work 디렉터리를 생성합니다.

```
$ mkdir work
```

기능 새 디렉터리 생성
형식 mkdir [디렉터리명] Enter↵

● **Step 03** | 새로 생성한 sample_f.txt 빈파일과 work 디렉터리가 존재하는지를 확인합니다.



[그림 5-21] 빈 파일과 새 디렉터리 생성

[예제 5-6]에서 실습한 바와 같이 일반파일 sample_f.txt는 rw- rw- r--와 같이 권한이 설정되어 있으며 work 디렉터리는 rwx rwx r-x와 같이 기본 접근권한이 자동으로 설정되어 있는 것을 확인하였습니다. 리눅스 시스템 환경에 현재 설정된 기본 접근권한을 확인하기 위해서는 umask 명령을 사용하며 사용형식은 다음과 같습니다.

$ umask

기능 리눅스 시스템에 현재 설정되어 있는 기본 접근권한을 출력하거나 변경
형식 umask [옵션] [마스크 값] Enter↵
옵션 -s : 마스크 값을 rwx와 같이 문자로 출력

- 현재 리눅스 시스템 환경에 설정된 기본 접근권한 조회

| 예제 5-7 |

현재 사용 중인 리눅스 시스템에 설정되어 있는 파일의 기본 접근권한에 대한 정보를 출력합니다.

$ umask -S

기능 리눅스 시스템에 현재 설정되어 있는 기본 접근권한을 출력하거나 변경
형식 umask [옵션] [마스크 값] Enter↵

```
cskisa@localhost:~/chap_05
파일(F)  편집(E)  보기(V)  검색(S)  터미널(T)  도움말(H)
[cskisa@localhost chap_05]$ umask -S
u=rwx,g=rwx,o=rx
[cskisa@localhost chap_05]$ umask
0002
[cskisa@localhost chap_05]$
```

[그림 5-22] 리눅스 시스템에 현재 설정된 파일의 기본 접근권한

[예제 5-7]에서 실습한 결과에서 볼 수 있듯이 현재 있는 위치가 chap_05 디렉터리이므로 현재 디렉터리에 설정되어 있는 일반 접근권한을 살펴보면 소유자 권한과 그룹 권한은 읽기, 쓰기, 실행이 모두 부여되어 있고 기타 사용자의 경우에는 읽기와 실행 권한만 부여되어 있음을 확인할 수 있습니다.

umask 명령과 옵션 -S를 함께 실행할 경우에는 현재 리눅스 시스템에 설정된 디렉터리의 기본 접근권한이 u=rwx, g=rwx, o=rx와 같이 문자로 출력되었고 옵션 없이 umask 명령만 사용할 경우에는 숫자 0002가 출력되었습니다. 출력된 0002 숫자는 마스크 값을 의미합니다.

마스크 값 적용

- 마스크 값은 파일이나 디렉터리를 생성할 때 부여하지 않을 권한에 대해서 지정하는 값을 의미
- 인자 없이 umask 명령만 사용하게 되면 현재 설정된 기본 마스크 값을 확인할 수 있음
- umask란 파일이 생성될 때 사용할 파일의 권한에 대해 mask를 인코딩하는 시스템 변수를 의미
- 이러한 권한 값을 퍼미션이라고도 하며 퍼미션은 보통 4자리(레드햇 7.2이하 버전은 3자리)로 표현

- 포미션의 4자리 숫자는 각각의 고유권한을 나타내며 맨 앞자리는 특수 퍼미션 <SetUid, SetGid>을 의미하고
- 다음 자리는 소유자 권한, 그 다음은 그룹 권한, 맨 마지막에는 기타 사용자 권한을 부여
- umask bit 형식

권한 구분	SetUid, SetGid	소유자 권한	그룹 권한	기타 사용자 권한
umask bit	0	0	0	2

프로그램 파일과 프로세스들에게
접근하는 것을 허용하는 기능을 수행

- 리눅스 시스템에서 마스크 값의 적용은 파일과 디렉터리에 따라 각각 다르게 적용됨

- 포미션의 4자리 중에서 특수 퍼미션 <SetUid, SetGid>을 의미하는 맨 앞의 0을 제외하고 002에 대한 의미에 대해서만 살펴보면

[표 5-7] 마스크 값 테이블

마스크 값	0	1	2	3	4	5	6	7
파일의 기본권한	6	6	4	4	2	2	0	0
디렉터리의 기본권한	7	7	6	4	3	2	1	0

- 좀 더 쉽게 이해 할 수 있도록 파일과 디렉터리에 대한 권한설정과 해당 모드의 코드 값을 아래 표와 같이 정리

[표 5-8] 파일과 디렉터리에 대한 기본권한과 해당모드 값

마스크 값		000	001	002	022
파일	권한 값	666	666	664	644
	기본 권한	-rw- rw- rw-	-rw- rw- rw-	-rw- rw- r--	-rw- r-- r--
디렉터리	권한 값	777	776	775	755
	기본 권한	drwx rwx rwx	drwx rwx rw-	drwx rwx r-x	drwx r-x r-x

기본 접근권한

- 리눅스 시스템에서 기본 값을 변경하면 기본 접근권한도 변경되도록 설정되어 있으며
- 기본 접근권한을 출력하거나 변경할 때 사용되는 umask 명령과 함께 마스크 값을 적용하면 기본 접근권한을 변경할 수 있음

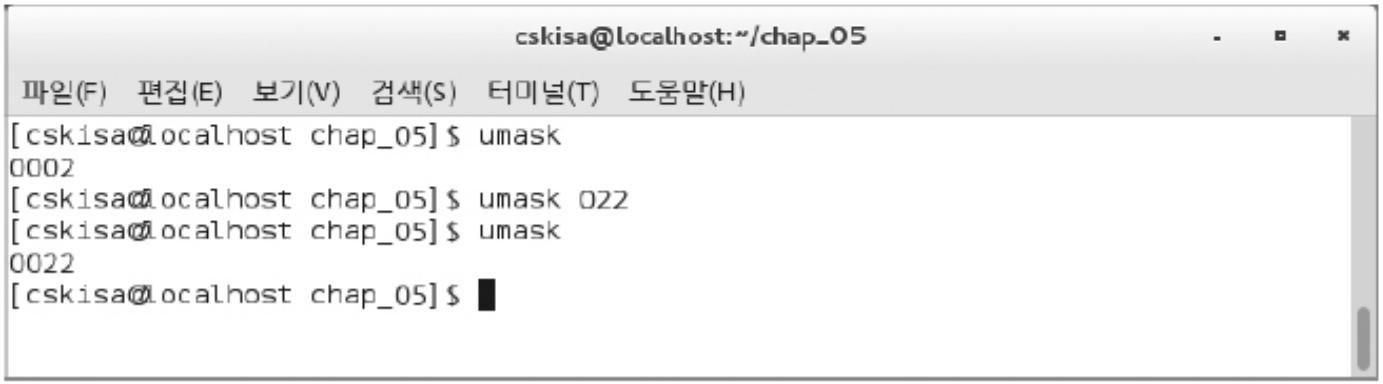
- 파일과 디렉터리에 대한 기본 접근권한 변경

| 예제 5-8 |

- **Step 01** | 현재 설정된 리눅스 시스템의 기본 접근권한을 출력한 다음 기본 접근권한을 마스크 값 022를 적용하여 기본 접근권한을 변경하고 변경된 마스크 값을 확인합니다.

$ umask 022

기능 리눅스 시스템에 현재 설정되어 있는 기본 접근권한을 출력하거나 변경
형식 umask [마스크 값] Enter↵



[그림 5-24] 마스크 값 022로 기본 접근권한

- **Step 02** | 파일 sample_022.txt와 work_022 디렉터리를 touch 명령과 mkdir 명령으로 생성한 다음 ls -l 명령을 수행하여 기본 접근권한을 출력합니다.

$ touch sample_022.txt

기능 빈파일 생성
형식 touch [일반 파일명] Enter↵

$ mkdir work_022

기능 새 디렉터리 생성
형식 mkdir [디렉터리명] Enter↵

```
cskisa@localhost:~/chap_05
파일(F)  편집(E)  보기(V)  검색(S)  터미널(T)  도움말(H)
[cskisa@localhost chap_05]$ touch sample_022.txt
[cskisa@localhost chap_05]$ mkdir work_022
[cskisa@localhost chap_05]$ ls -l
합계 4
-r-xrwxr-x. 1 cskisa cskisa 215 12월  4 15:35 centos_cp.txt
-rw-r--r--. 1 cskisa cskisa   0 12월 10 17:50 sample_022.txt
-rw-rw-r--. 1 cskisa cskisa   0 12월 10 15:21 sample_f.txt
drwxrwxr-x. 2 cskisa cskisa   6 12월 10 15:22 work
drwxr-xr-x. 2 cskisa cskisa   6 12월 10 17:50 work_022
[cskisa@localhost chap_05]$
```

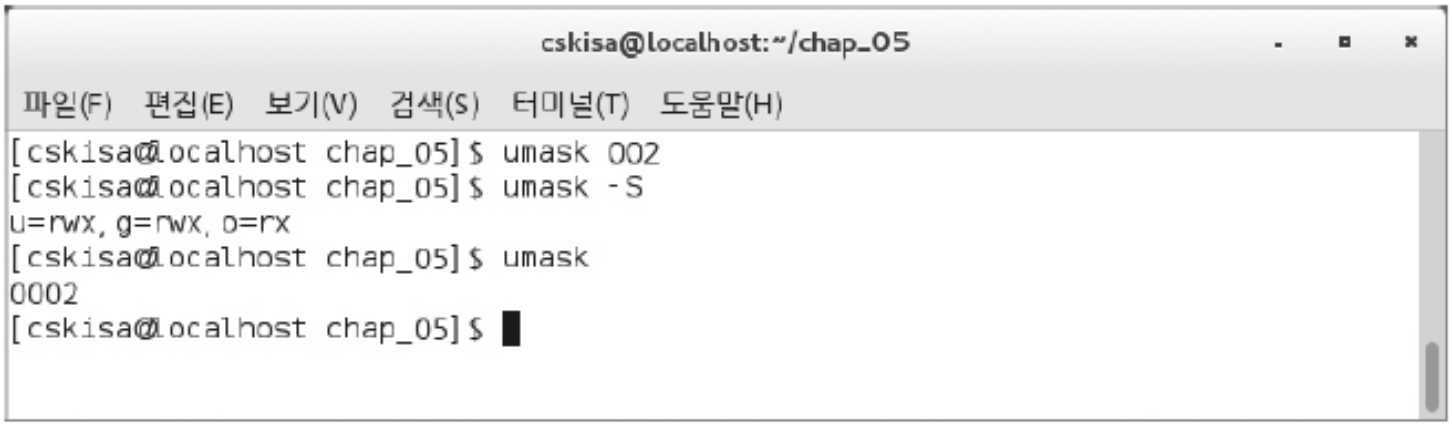
[그림 5-25] sample_022 파일과 work_022 디렉터리 생성

- **Step 03** | 기본 접근권한을 022로 변경 설정한 마스크 값을 원래 설정되어 있던 기본 마스크 값 002로 다시 변경한 다음 umask 명령으로 변경된 기본 접근권한을 확인합니다.

$ umask 002

기능 리눅스 시스템에 현재 설정되어 있는 기본 접근권한을 출력하거나 변경

형식 umask [마스크 값] Enter↵



[그림 5-26] 마스크 값 002로 원래 기본 접근권한으로 변경

① 파일에 대한 기본 접근권한 : sample_f.txt 파일과 sample_022.txt 파일

umask 022 명령을 수행하기 전에 생성했던 sample_f.txt 파일은 -rw- rw- r--과 같이 설정되어 있는 반면 umask 022 명령을 수행한 다음 생성한 sample_022.txt 파일은 -rw- r-- r--과 같이 다르게 출력된 것을 확인할 수 있습니다. 즉 그룹권한이 rw-과 r--으로 기본 접근권한이 각각 다르게 설정되어 있음을 알 수 있습니다.

② 디렉터리에 대한 기본 접근권한 : work 디렉터리와 work_022 디렉터리

umask 022 명령을 수행하기 전에 생성했던 work 디렉터리는 drwx rwx r-x와 같이 설정되어 있는 반면 umask 022 명령을 수행한 다음 생성한 work_022 디렉터리는 drwx r-x r-x와 같이 다르게 출력된 것을 확인할 수 있습니다. 즉 그룹권한에 대한 설정이 rwx와 r-x로 기본 접근권한이 각각 다르게 설정되어 있음을 알 수 있습니다.

- **Step 04** | 기본 마스크 값을 변경하기 전과 변경한 후에 생성된 파일과 디렉터리에 대한 기본 접근권한에 어떠한 변화가 있는지를 ls -l 명령으로 확인합니다.

```
cskisa@localhost:~/chap_05
파일(F)  편집(E)  보기(V)  검색(S)  터미널(T)  도움말(H)
[cskisa@localhost chap_05]$ ls -l
합계 4
-r-xrwxr-x. 1 cskisa cskisa 215 12월  4 15:35 centos_cp.txt
-rw-r--r--. 1 cskisa cskisa   0 12월 10 17:50 sample_022.txt
-rw-rw-r--. 1 cskisa cskisa   0 12월 10 15:21 sample_f.txt
drwxrwxr-x. 2 cskisa cskisa   6 12월 10 15:22 work
drwxr-xr-x. 2 cskisa cskisa   6 12월 10 17:50 work_022
[cskisa@localhost chap_05]$
```

[그림 5-27] 마스크 값을 변경해도 기존 생성한 파일과 디렉터리에는 변화 없음

[예제 5-8]을 실습한 결과 umask 022 명령으로 생성했던 파일과 디렉터리에 부여된 기본 접근권한은 마스크 값을 umask 002 명령으로 다시 변경해도 기존에 생성한 파일과 디렉터리에는 생성 당시의 기본 접근권한이 적용되어 있기 때문에 접근권한에 대한 변화는 없습니다.

실습 5-4 다음 항목에서 주어진 지시사항을 수행하시오.

1. 현재 리눅스 시스템에 설정된 기본 접근권한 마스크 값을 출력하기
2. 새로운 디렉터리 zone_d 생성하기
3. 새로운 빈파일 first.txt 생성하기
4. 새로 생성한 디렉터리와 파일에 대한 파일 접근권한 출력하기
5. 리눅스 시스템의 파일에 대한 기본 접근권한이 -rw- rw- rw-로 설정되도록 마스크 값을 변경하기
6. 마스크 값 변경 후 zone_d 디렉터리와 first.txt 파일에 대한 권한변경 여부 확인하기
7. 당초에 설정되어 있던 기본 접근권한 마스크 값으로 다시 설정하기

```
$ umask
$ mkdir zone_d
$ touch first.txt
$ ls –l
$ umask 001
$ ls –l
$ umask 002
```

특수 접근권한

- 파일 생성 시 부여되는 특수 접근권한 이해하기

1. 특수 접근권한의 종류에 대해 살펴봅니다.
2. 특수 접근권한 설정 값에 대해 이해합니다.
3. 특수 접근권한 설정 값을 적용하는 방법에 대해 실습합니다.

■ 특수 접근권한은 SetUid, SetGid, Sticky Bit 세 가지가 있음

[표 5-9] 특수 접근권한 설정 값

접근권한	SetUid	SetGid	Sticky Bit
권한설정 값	4000	2000	1000

[표 5-10] 특수 접근권한과 권한 모드

접근권한			소유자 권한			그룹 권한			기타 사용자 권한		
SetUid	SetGid	Sticky Bit	r	w	x	r	w	x	r	w	x
4000	2000	1000	400	200	100	40	20	10	4	2	1

SetUid

- 특정 파일에 SetUid가 설정되어 있다면 다른 사용자들이 그 파일을 실행하였을 경우와
- 실행되는 동안에는 실행시킨 사용자의 권한(아이디의 권한)이 아닌 파일 소유자 권한으로 실행하게 됨
- SetUid를 사용하여 파일의 소유자 권한을 실행할 수 있도록 설정하려면 접근권한에서 맨 앞자리에 4를 설정해야 함

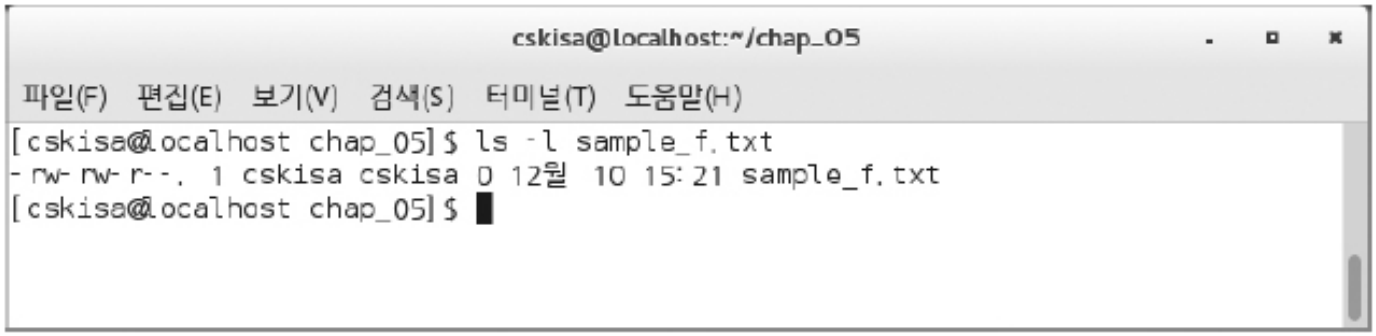
- 특수 접근권한 SetUid 적용

| 예제 5-9 |

- **Step 01** | chap_05 디렉터리에 존재하는 sample_f.txt 파일의 권한을 확인합니다.

```
$ ls -l sample_f.txt
```

기능 파일의 정보를 상세하게 출력
형식 ls [옵션] [파일명] Enter↵



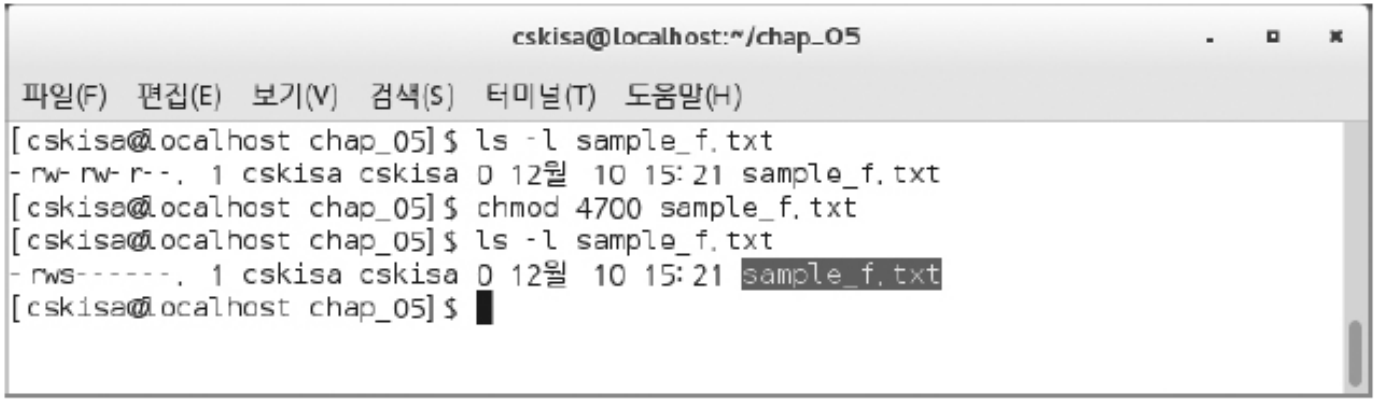
[그림 5-28] sample_f.txt 파일의 상세정보 출력

- **Step 02** | sample_f.txt 파일의 권한이 664(rw- rw- r--)로 설정되어 있는 권한 값을 확인한 다음 권한 값을 4700(- rws --- ---)으로 변경하여 적용합니다.

```
$ chmod 4700 sample_f.txt
```

기능 파일의 접근권한 변경

형식 chmod [SetUid + 숫자 모드] [파일명] Enter↵



[그림 5-29] SetUid를 적용하여 sample_f.txt 파일에 특수 접근권한 설정

[예제 5-9]를 통해 sample_f.txt 파일에 특수 접근권한 SetUid를 적용한 결과 접근권한은 rws --- ---으로 변경되었습니다. 여기에서 주의 깊게 살펴봐야 할 부분은 소유자 권한이 rws로 변경되었다는 부분입니다. 이 부분에 대해 좀 더 자세하게 살펴보면 일반 접근권한으로 설정된 sample_f.txt 파일의 숫자 모드는 644 (rw- rw- r--)로 설정되어 있었지만 chmod 명령으로 접근권한 값을 4700으로 변경하여 설정하였더니 소유자 권한이 rwx가 아닌 rws로 설정되었음을 확인하였습니다.

SetGid

- 특수 권한이 설정된 파일을 실행할 경우 해당 파일이 실행되는 동안에는 파일 소유 그룹의 권한으로 실행되도록 적용해 주는 기능
- SetGid는 접근권한의 맨 앞자리에 2를 설정해야 하며 이와 같이 설정하면 그룹 권한에서 소유 그룹의 권한 값에
- 실행의 값인 x 대신 s 기호가 표기되고 그룹에 실행 권한이 없을 경우에는 대문자 S가 표기됨
- 특수 접근권한 설정에 있어서 절대모드 표현방법으로 적용할 경우 일반 권한 값에 8진수 2000을 설정하여 해당 파일의 그룹권한 변경

- 특수 접근권한 SetGid 적용

예제 5-10

- Step 01 | 현재 디렉터리에 존재하는 sample_f.txt 파일의 상세한 정보를 출력합니다.

```
$ ls -l sample_f.txt
```

기능 파일의 상세한 내용 출력
형식 ls [옵션] [파일명] Enter↵

- Step 02 | sample_f.txt 파일에 대한 그룹 소유자 권한을 2700으로 변경합니다.

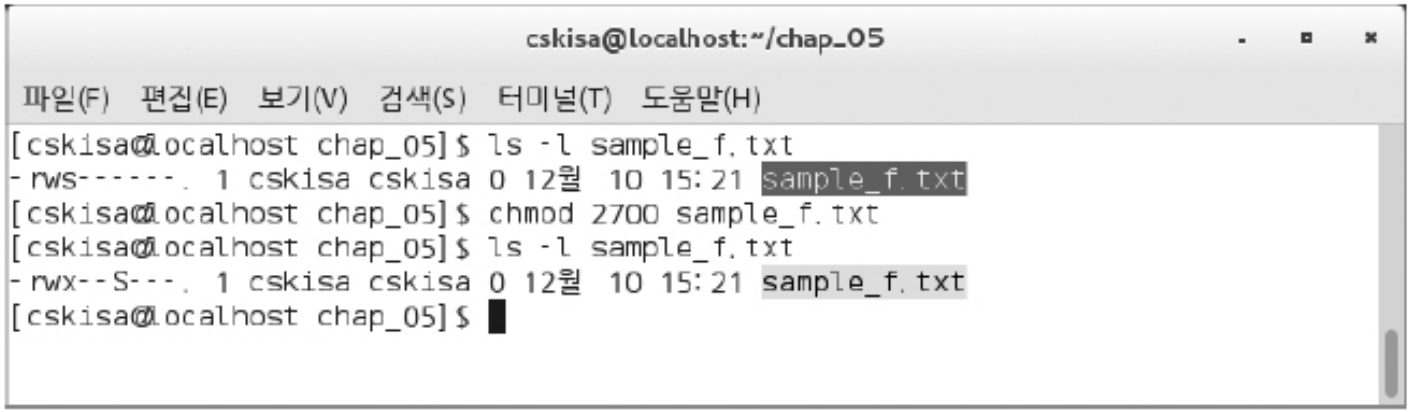
```
$ chmod 2700 sample_f.txt
```

기능 파일의 접근권한 변경
형식 chmod [SetGid + 숫자 모드] [파일명] Enter↵

● **Step 03** | 특수 접근권한을 변경 설정한 sample_f.txt 파일의 상세한 정보를 출력합니다.

$ ls -l sample_f.txt

기능 파일의 상세한 내용 출력
형식 ls [옵션] [파일명] Enter↵



[그림 5-30] SetGid를 적용하여 sample_f.txt 파일에 특수 접근권한 설정

[예제 5-10]에서 실습한 결과에서 볼 수 있듯이 sample_f.txt 파일에 SetGid의 설정 값을 2700으로 지정하여 그룹 소유자의 특수 접근권한을 변경하였습니다. 실습 파일인 sample_f.txt 파일은 특수 접근권한을 설정하기 전에 설정된 권한은 rws --- ---에서 rwx --S ---으로 변경된 것을 확인하였습니다. 그리고 SetGid 설정 후 그룹권한에 표기된 권한묶음이 대문자 S가 표기된 것은 그룹에 대한 실행권한이 없기 때문입니다. 소유자에 대한 권한묶음도 rws에서 rwx로 변경되어 적용되었습니다.

Sticky Bit

- 일반 파일이 아닌 디렉터리에 대해 특수권한을 설정하는 기능으로 아무런 제약 없이 누구나 디렉터리에 파일을 생성할 수 있음
- 이 디렉터리에 생성되는 파일은 파일을 생성한 소유자의 파일로 귀속되며 다른 사용자는 이 파일에 대해 어떠한 이유라도 삭제할 수 없음
- Sticky Bit는 접근권한의 맨 앞자리에 8진수 1000을 설정해야 하며
- Sticky Bit가 설정되면 기타 사용자의 실행권한에 t가 표시되고 실행권한이 없는 파일의 경우에는 대문자 T가 표기됨

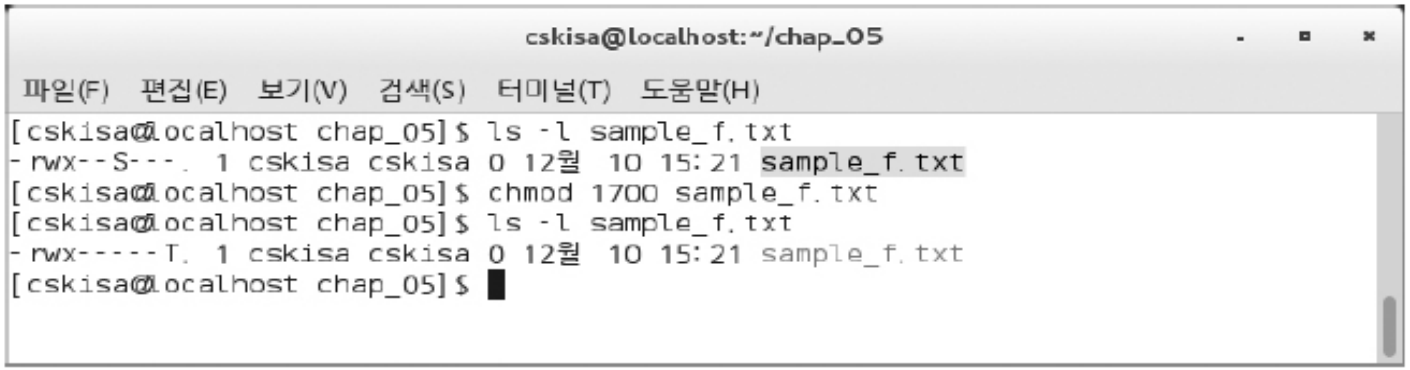
- 특수 접근권한 Sticky Bit 적용

예제 5-11

- **Step 01** | chap_05 디렉터리에 존재하는 sample_f.txt 파일의 권한설정 값을 ls -l 명령으로 확인한 다음 Sticky Bit의 설정 값을 1700으로 설정한 다음 특수 권한이 설정된 이 파일에 대한 권한묶음을 확인합니다.

```
$ chmod 1700 sample_f.txt
```

기능 파일의 접근권한 변경
형식 chmod [Sticky + 숫자 모드] [파일명] Enter



[그림 5-31] Sticky Bit를 적용하여 sample_f.txt 파일에 특수 접근권한 설정

- **Step 02** | chap_05 디렉터리에 존재하는 work 디렉터리의 권한설정 값을 ls -ld 명령으로 확인합니다.

```
$ ls -ld work
```

기능 디렉터리의 정보를 상세하게 출력
형식 ls [옵션] [디렉터리명] Enter↵

- **Step 03** | Sticky Bit의 설정 값을 1755로 설정한 다음 work 디렉터리에 설정된 특수권한을 확인합니다.

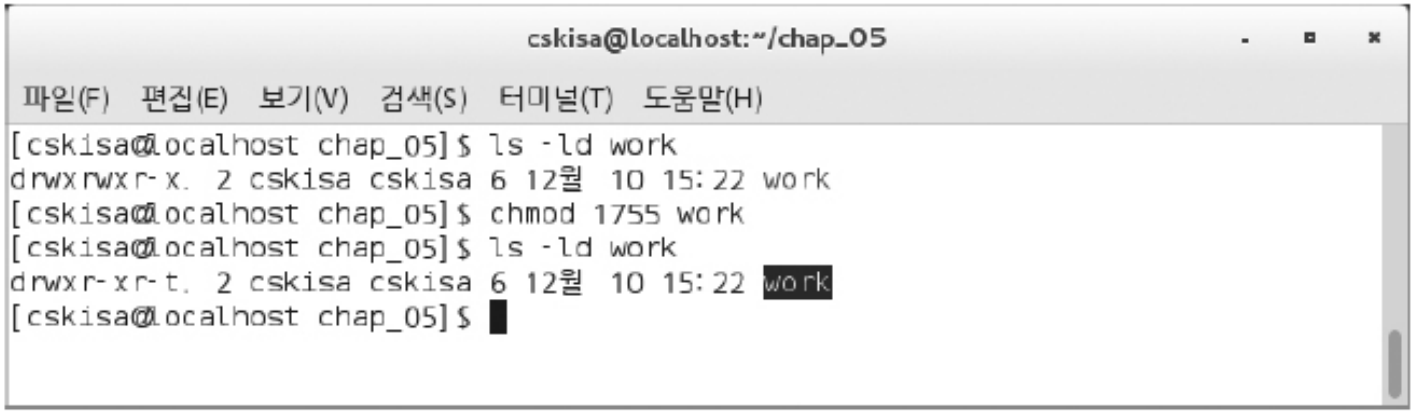
```
$ chmod 1755 work
```

기능 디렉터리의 접근권한 변경
형식 chmod [Sticky Bit + 숫자 모드] [디렉터리명] Enter↵

- **Step 04** | Sticky Bit의 설정 값을 변경 적용한 work 디렉터리의 권한설정 값을 ls -ld 명령으로 확인합니다.

```
$ ls -ld work
```

기능 디렉터리의 정보를 상세하게 출력
형식 ls [옵션] [디렉터리명] Enter↵



[그림 5-32] Sticky Bit를 적용하여 work 디렉터리에 특수 접근권한 설정

[예제 5-11]의 Step 01에서 실습한 sample_f.txt 파일에 Sticky Bit 설정 값을 1700으로 설정하였더니 파일에 대한 특수 접근권한이 rwx --- --T로 표기되었습니다. 이와 같이 표기된 이유는 기타 사용자에게 실행권한이 없는 파일이기 때문에 대문자 T가 표기된 것입니다.

그리고 Step 02에서는 실습한 work 디렉터리에 Sticky Bit 설정 값을 1755로 설정하였더니 디렉터리에 대한 특수 접근권한이 rwx rwx r-x에서 rwx r-x r-t로 변경된 것을 확인하였습니다. 좀 더 자세히 살펴보면 기타 사용자의 권한묶음이 r-x에서 r-t와 같이 표기된 것은 work 디렉터리를 다른 사용자가 실행할 수 있는 권한이 부여된 디렉터리임을 의미합니다.

Chapter 05

최상의 노력에 따른 인고의 가치는 반드시 증명될 수 있습니다!

Thank You