

8장 파일의 권한

강사 김영석

A top-down view of a wooden desk. On the desk, there is a silver laptop with a black keyboard, a pair of black-rimmed glasses, a white coffee cup with a yellow handle, and a small green succulent in a pot. The word 'CONTENT' is written in large, white, bold, sans-serif capital letters over the left side of the image.

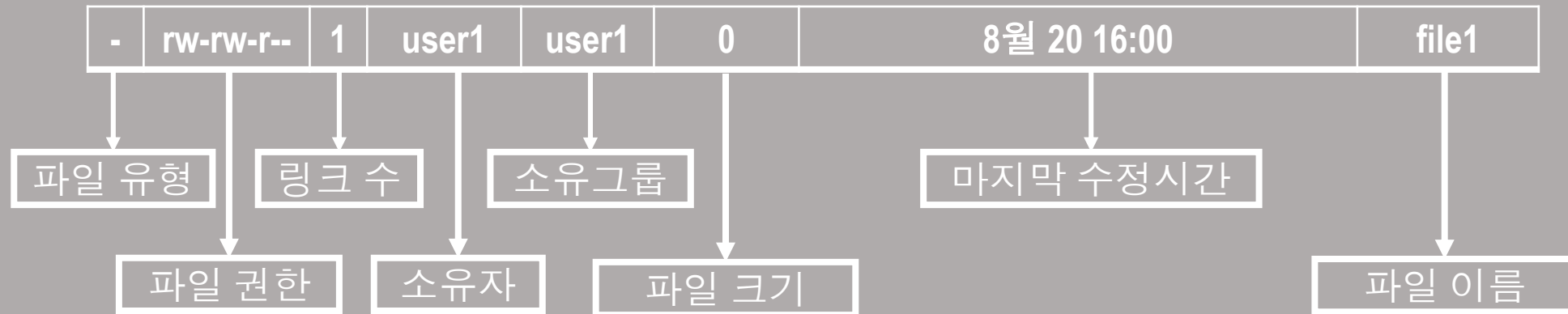
CONTENT

- 1 파일의 속성
- 2 파일의 허가권 및 소유권
- 3 ACL
- 4 umask
- 5 특수권한

1. 파일의 속성

❖ 파일의 속성

```
user1@Server1:~$ ls -l
합계 36
-rw-rw-r-- 1 user1 user1 0 8월 20 16:00 file1
drwx----- 3 user1 user1 4096 8월 17 16:39 snap
drwxr-xr-x 2 user1 user1 4096 8월 17 16:39 공개
```





❖ 파일 유형

파일 구분	파일 유형
d	디렉터리 파일
l	심볼릭 링크 파일
b	저장 장치 디바이스 파일
c	입출력 장치 디바이스 파일
-	일반 파일



❖ 파일 소유 권한

권한	권한 설명
r	읽기
w	쓰기
x	실행
-	권한 없음

2. 파일의 허가권 및 소유권

❖ 파일 허가권은 디렉터리 및 파일을 사용하거나 사용하지 못하도록 하는 권한이다.

```
-rw-rw-r-- 1 user1 user1 0 8월 20 16:00 file1
```

소유자

소유그룹

그외 사용자

권한	권한 설명
rwX	모든 권한
rw-	읽고 쓰기
r-X	읽고 실행
r--	읽기
---	모든 권한 없음

❖ 소유자와 소유 그룹 변경

chown	
기능	파일과 디렉터리의 소유자와 소유 그룹 변경
형식	chown [옵션][사용자 계정][파일명/디렉터리명]
옵션	-R : 서브 디렉터리의 소유자와 소유 그룹도 변경
사용 예	chown user1 file1 chown user1:gtest file1 chown -R user1 file1

```
user1@Server1:~$ sudo chown root:root file1
[sudo] user1 암호:
user1@Server1:~$ ls -l
합계 36
-rw-rw-r-- 1 root  root    0  8월 20 16:00 file1
```

❖ 소유 그룹만 변경


chgrp	
기능	파일과 디렉터리의 소유 그룹을 변경
형식	chgrp [옵션][사용자 계정][파일명/디렉터리명]
옵션	-R : 서브 디렉터리의 소유자와 소유 그룹도 변경
사용 예	chgrp user1 file1

```
user1@Server1:~$ sudo chgrp gtest file1
user1@Server1:~$ ls -l
합계 36
-rw-rw-r-- 1 root  gtest    0  8월 20 16:00 file1
drwx----- 3 user1 user1 4096  8월 17 16:39 snap
```


❖ 파일 권한 변경

chmod	
기능	파일과 디렉터리의 권한 변경
형식	chmod [권한][파일명/디렉터리명]
사용 예	chmod 777 file1

권한	권한 설명	이진수 표현	십진수 표현
rwx	모든 권한	111	7
rw-	읽고 쓰기	110	6
r-x	읽고 실행	101	5
r--	읽기	100	4
---	권한 없음	000	0



```
user1@Server1:~$ ls -l file1
-rw-rw-r-- 1 root gtest 0  8월 20 16:00 file1
user1@Server1:~$ sudo chmod 777 file1
user1@Server1:~$ ls -l file1
-rwxrwxrwx 1 root gtest 0  8월 20 16:00 file1
user1@Server1:~$ sudo chmod 666 file1
user1@Server1:~$ ls -l file1
-rw-rw-rw- 1 root gtest 0  8월 20 16:00 file1
user1@Server1:~$ sudo chmod 444 file1
user1@Server1:~$ ls -l file1
-r--r--r-- 1 root gtest 0  8월 20 16:00 file1
```

Quiz

❖ /home/user1/linux_ex/ch8 디렉터리 생성 후 아래 내용을 실습한다.

- testNum.txt 파일을 생성한다.
- 기타 사용자에게 실행 권한을 부여한다.
- 그룹과 기타 사용자의 실행 권한을 제거한다.
- 모두에게 실행 권한을 부여한다.
- 소유자에게 쓰기 권한을 부여하고 그룹의 쓰기 권한은 제거한다.
- 소유자의 권한만 남기고 나머지 사용자의 권한은 모두 제거한다.

❖ 기호를 이용한 권한 변경

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


```
user1@Server1:~/바탕화면$ touch test.txt
user1@Server1:~/바탕화면$ ls -l
합계 0
-rw-rw-r-- 1 user1 user1 0  8월 30 16:55 test.txt
user1@Server1:~/바탕화면$ sudo chmod u-w test.txt
[sudo] user1 암호:
user1@Server1:~/바탕화면$ ls -l
합계 0
-r--rw-r-- 1 user1 user1 0  8월 30 16:55 test.txt
user1@Server1:~/바탕화면$ sudo chmod u+x test.txt
user1@Server1:~/바탕화면$ ls -l
합계 0
-r-xrw-r-- 1 user1 user1 0  8월 30 16:55 test.txt
user1@Server1:~/바탕화면$ sudo chmod g-r test.txt
user1@Server1:~/바탕화면$ ls -l
합계 0
-r-x-w-r-- 1 user1 user1 0  8월 30 16:55 test.txt
user1@Server1:~/바탕화면$ sudo chmod u+rw test.txt
user1@Server1:~/바탕화면$ ls -l
합계 0
-rwx-w-r-- 1 user1 user1 0  8월 30 16:55 test.txt
```

```
user1@Server1:~/바탕화면$ sudo chmod +rwx test.txt
user1@Server1:~/바탕화면$ ls -l
합계 0
-rwxrwxr-x 1 user1 user1 0  8월 30 16:55 test.txt
user1@Server1:~/바탕화면$ sudo chmod a-rwx test.txt
user1@Server1:~/바탕화면$ ls -l
합계 0
----- 1 user1 user1 0  8월 30 16:55 test.txt
user1@Server1:~/바탕화면$ sudo chmod u=rwx test.txt
user1@Server1:~/바탕화면$ ls -l
합계 0
-rwx----- 1 user1 user1 0  8월 30 16:55 test.txt
user1@Server1:~/바탕화면$ sudo chmod go=rx test.txt
user1@Server1:~/바탕화면$ ls -l
합계 0
-rwxr-xr-x 1 user1 user1 0  8월 30 16:55 test.txt
user1@Server1:~/바탕화면$ sudo chmod u-x,go-x test.txt
user1@Server1:~/바탕화면$ ls -l
합계 0
-rw-r--r-- 1 user1 user1 0  8월 30 16:55 test.txt
```



Quiz

- ❖ /home/user1/linux_ex/ch8 디렉터리에서 아래 내용을 실행한다.
 - 파일 testOp.txt 를 생성한다.
 - 그룹에 쓰기와 실행 권한을 부여한다.
 - 기타 사용자에게 실행 권한을 부여한다.
 - 그룹과 기타 사용자의 실행 권한을 제거한다.
 - 모두에게 실행 권한을 부여한다.
 - 소유자에게 쓰기 권한을 부여하고 그룹의 쓰기 권한은 제거 한다.

3. ACL

- ❖ ACL은 Access Control List 로 접근 제어 목록이다
- ❖ 파일 및 디렉터리의 접근에 대해서 권한 이외에 권한을 추가적으로 설정 할 수 있다.

getfacl	
기능	파일 및 디렉터리의 ACL을 확인
형식	getfacl [옵션][파일명/디렉터리명]
옵션	-a : 모든 권한 확인 -ACL 사용자 포함 -d : 기본 권한만 확인 -ACL 사용자 제외 -R : 디렉터리 인 경우 하위 내용까지 확인
사용 예	getfacl -a file1 getfacl -d file1 getfacl -R dir1



```
user1@Server1:~$ getfacl -a file1
```

```
# file: file1
```

```
# owner: root
```

```
# group: gtest
```

```
user::r--
```

```
group::r--
```

```
other::r--
```

```
user1@Server1:~$ getfacl -d file1
```

```
# file: file1
```

```
# owner: root
```

```
# group: gtest
```

```
user1@Server1:~$ getfacl -R snap
```

```
# file: snap
```

```
# owner: user1
```


```
# group: user1
```

```
user::rwx
```

```
group::---
```


```
other::---
```

setfacl	
기능	파일 및 디렉터리의 ACL을 설정
형식	setfacl [옵션] [파일명/디렉터리명]
옵션	-m : 권한 수정 -x : 권한 삭제 -b : 모든 권한을 삭제 -R : 디렉터리 인 경우 하위 내용까지 권한 수정
사용 예	setfacl -m u:user1:4 file1 setfacl -m user:user1:rwX file1 setfacl -x user1 file1 setfacl -b file1



```
user1@Server1:~$ sudo setfacl -m u:root:7 file1
user1@Server1:~$ sudo getfacl -a file1
# file: file1
# owner: root
# group: gtest
user::r--
user:root:rwX
group::r--
mask::rwX
other::r--
```

```
user1@Server1:~$ sudo setfacl -m user:user1:rwX,u:root:4 file1
user1@Server1:~$ sudo getfacl -a file1
# file: file1
# owner: root
# group: gtest
user::r--
user:root:r--
user:user1:rwX
group::r--
mask::rwX
other::r--
```

```
user1@Server1:~$ sudo setfacl -x user:user1 file1
user1@Server1:~$ sudo getfacl -a file1
# file: file1
# owner: root
# group: gtest
user::r--
user:root:r--
group::r--
mask::r--
other::r--
```

```
user1@Server1:~$ sudo setfacl -b file1
user1@Server1:~$ sudo getfacl -a file1
# file: file1
# owner: root
# group: gtest
user::r--
group::r--
other::r--
```

Quiz


❖ /home/user1/linux_ex/ch8 디렉터리에서 아래 내용을 실행한다.

- acltest.txt 파일을 생성한다.
- acltest.txt 의 소유자, 소유 그룹을 root 로 변경한다.
- acltest.txt 의 권한은 사용자만 모든 권한을 준다.
- acl 을 이용하여 user1 에게 읽기 권한을 주고 권한을 확인한다.
- acl 을 이용하여 user1 에게 읽기 쓰기 권한을 주고 권한을 확인한다.
- user1 의 모든 권한을 삭제한다.
- acl 을 이용하여 user1 그룹에게 읽기 권한을 주고 권한을 확인한다.
- acl 을 이용하여 user1 그룹에게 읽기 쓰기 권한을 주고 권한을 확인한다.
- 0

4. umask

- ❖ umask는 파일이나 디렉터리 생성할 때 권한을 설정해 주는 기능을 한다.
- ❖ 파일 생성 시 기본 권한은 0666, 디렉터리 생성 시 기본 권한은 0777이다. umask의 기본 값을 0002이다.
- ❖ umask의 0002 값을 빼면 파일이 생성 되었을 때 0664가 되고, 디렉터리는 0775가 된다.

```
user1@Server1:~$ umask
0002
user1@Server1:~$ touch file2
user1@Server1:~$ ls -l file2
-rw-rw-r-- 1 user1 user1 0  8월 20 20:26 file2
```



```
user1@Server1:~$ umask 0022
user1@Server1:~$ touch file2
user1@Server1:~$ ls -l file2
-rw-r--r-- 1 user1 user1 0  8월 20 20:30 file2
user1@Server1:~$ mkdir aaa
user1@Server1:~$ ls -l
합계 44
drwxrwxr-x 2 user1 user1 4096  8월 20 20:31 aaa
user1@Server1:~$ umask 0022
user1@Server1:~$ mkdir bbb
user1@Server1:~$ ls -l
합계 48
drwxrwxr-x 2 user1 user1 4096  8월 20 20:31 aaa
drwxr-xr-x 2 user1 user1 4096  8월 20 20:32 bbb
```

Quiz


- ❖ /home/user1/linux_ex/ch8 디렉터리에서 아래의 내용을 실행한다.
 - 현재의 기본 접근 권한을 확인한다.
 - 그룹과 기타 사용자가 읽기와 실행을 할 수 없도록 기본 접근 권한을 변경한다.
 - testUmask.txt 파일을 생성하고 기본 접근 권한이 변경되었는지 확인한다.

5. 특수 권한

❖ setuid

- umask 에서도 봤듯이 본래 권한을 0666 이나 0777 과 같이 4자리의 숫자로 이루어져 있다. 특수 권한은 제일 앞에 있는 권한이라고 보면 된다.
- 그 중 setuid는 root 사용자만 접근 할 수 있는 파일이나 명령에 대해 일반 사용자로 접근 허용 하도록 하는 권한이다.
- sudo 명령과 마찬가지로 실행한 순간만 권한을 빌려오는 것이라고 이해하면 쉽다.
- 하지만 보안의 취약하기 때문에 setuid 권한을 사용하는 것은 최소화 해야 한다.

setuid	
사용 예	chmod 4744 file1 chmod u+s file1



```
user1@Server1:~/public$ tail /etc/shadow
tail: 읽기를 위해 '/etc/shadow'을(를) 열 수 없음: 허가 거부
user1@Server1:~/public$ sudo chmod 4640 /etc/shadow
user1@Server1:~/public$ tail /etc/shadow
tail: 읽기를 위해 '/etc/shadow'을(를) 열 수 없음: 허가 거부
user1@Server1:~/public$ ls -l /bin/tail
-rwxr-xr-x 1 root root 68120  2월  8  2022 /bin/tail
user1@Server1:~/public$ sudo chmod 4755 /bin/tail
user1@Server1:~/public$ tail /etc/shadow
fwupd-refresh:*:19576:0:99999:7:::
nm-openvpn:*:19576:0:99999:7:::
saned:*:19576:0:99999:7:::
```



```
user1@Server1:~$ ls -l /etc/shadow
-rw-r--r-- 1 root shadow 1536  8월 20 15:07 /etc/shadow
user1@Server1:~$ sudo chmod 4744 /etc/shadow
user1@Server1:~$ ls -l /etc/shadow
-rwsr--r-- 1 root shadow 1536  8월 20 15:07 /etc/shadow
user1@Server1:~$ sudo chmod 644 /etc/shadow
user1@Server1:~$ ls -l /etc/shadow
-rw-r--r-- 1 root shadow 1536  8월 20 15:07 /etc/shadow
user1@Server1:~$ sudo chmod u+s /etc/shadow
user1@Server1:~$ ls -l /etc/shadow
-rwSr--r-- 1 root shadow 1536  8월 20 15:07 /etc/shadow
user1@Server1:~$ sudo chmod u-s /etc/shadow
user1@Server1:~$ ls -l /etc/shadow
-rw-r--r-- 1 root shadow 1536  8월 20 15:07 /etc/shadow
```


실행 권한이 있으면 s, 실행 권한이 없으면 S 로 표시된다.

❖ setgid

- setuid 은 사용자가 root 권한으로 실행하는 것이고, setgid 는 root 그룹으로 실행 하도록 해주는 특수 권한이다.

setgid	
사용 예	chmod 2744 file1 chmod g+s file1

```
user1@Server1:~$ su
암호:
root@Server1:/home/user1# mkdir public
root@Server1:/home/user1# chmod 2777 public/
root@Server1:/home/user1# su user1
user1@Server1:~$ cd public
user1@Server1:~/public$ touch aaa.txt
user1@Server1:~/public$ ls -l
합계 0
-rw-rw-r-- 1 user1 root 0  8월 21 17:58 aaa.txt
```

```
user1@Server1:~$ sudo chmod g-s public/
user1@Server1:~$ ls -ld public/
drwxrwxrwx 2 root root 4096  8월  21 18:03 public/
user1@Server1:~$ cd public/
user1@Server1:~/public$ touch bbb.txt
user1@Server1:~/public$ ls -l
합계 0
-rw-rw-r-- 1 user1 root  0  8월  21 18:03 aaa.txt
-rw-rw-r-- 1 user1 user1 0  8월  21 18:04 bbb.txt
```





❖ sticky

- 특정 디렉터리를 누구나 자유롭게 사용할 수 있게 하기 위해서 사용한다.
- sticky 비트가 디렉터리에 적용되면 디렉터리 소유자나 파일 소유자 또는 슈퍼 유저가 아닌 사용자들은 파일을 삭제하거나 이름을 변경하지 못하도록 막지만 파일 또는 디렉터리는 누구나 생성 할 수 있다.

```
user1@Server1:~$ sudo chmod 1777 public/  
user1@Server1:~$ ls -ld public/  
drwxrwxrwt 2 root root 4096  8월 21 18:04 public/
```



```
user1@Server1:~$ cd public
user1@Server1:~/public$ touch ccc.txt
user1@Server1:~/public$ ls -l
합계 0
-rw-rw-r-- 1 user1 root 0  8월 21 18:03 aaa.txt
-rw-rw-r-- 1 user1 user1 0  8월 21 18:04 bbb.txt
-rw-rw-r-- 1 user1 user1 0  8월 21 18:09 ccc.txt
user1@Server1:~/public$ mkdir test
user1@Server1:~/public$ rm -rf test
user1@Server1:~/public$ rm -rf ccc.txt
user1@Server1:~/public$ ls -l
합계 0
-rw-rw-r-- 1 user1 root 0  8월 21 18:03 aaa.txt
-rw-rw-r-- 1 user1 user1 0  8월 21 18:04 bbb.txt
```



```
user1@Server1:~/public$ sudo useradd -m user2
```

```
user1@Server1:~/public$ sudo passwd user2
```

새 암호:

잘못된 비밀번호: 암호가 앞뒤 어느쪽에서 읽어도 같은 문맥입니다

새 암호 다시 입력:

passwd: 암호를 성공적으로 업데이트했습니다

```
user1@Server1:~/public$ su user2
```

암호:

```
$ ls -l
```

합계 0

```
-rw-rw-r-- 1 user1 root 0  8월 21 18:03 aaa.txt
```

```
-rw-rw-r-- 1 user1 user1 0  8월 21 18:04 bbb.txt
```

```
$ rm -rf aaa.txt
```

rm: 'aaa.txt'을(를) 제거할 수 없습니다: 명령을 허용하지 않음

A photograph of a server room with rows of server racks on both sides of a central aisle. The racks have glass doors and internal components are visible, with many small blue lights glowing. The ceiling has several long, rectangular light fixtures. The overall atmosphere is dimly lit, emphasizing the blue light from the servers.

수고하셨습니다.