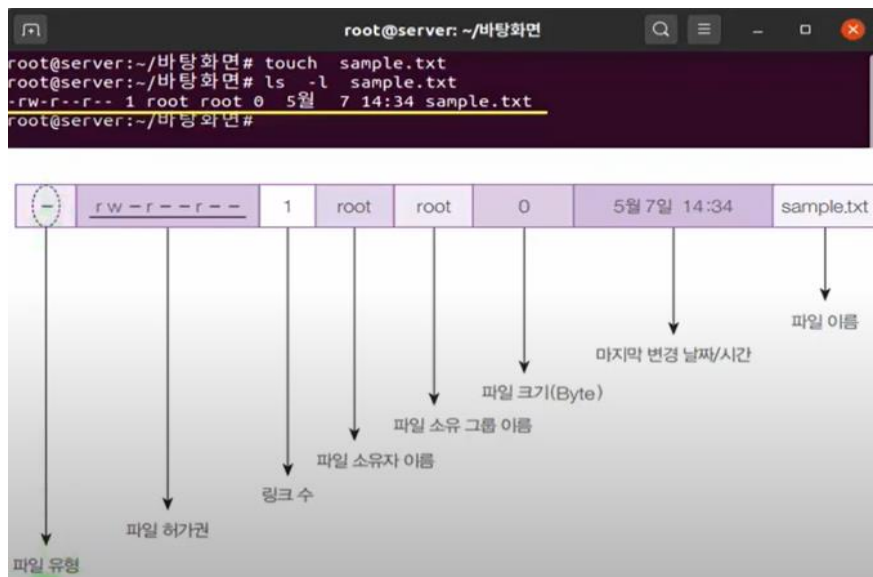


- Ch_6 _ 파일의 허가권과 소유권
 - Ch_7 _ 명령어:링크
 - 파일과 디렉터리의 소유와 허가권 (1)
- 파일의 리스트와 파일 속성



- 파일과 디렉터리의 소유와 허가권 (2)

➤ 파일 유형

디렉터리일 경우에는 d, 일반적인 파일일 경우에는 -가 표시

➤ 파일 허가권 (Permission)

"rw-", "r--", "r--" 3 개씩 끊어서 읽음 (r 은 read, w 는 write, x 는 execute 의 약자)

첫 번째 "rw-"는 소유자(user)의 파일접근 권한

두 번째의 "r--"는 그룹(Group)의 파일접근 권한

세 번째의 "r--"는 그 외의 사용자(other)의 파일접근 권한

숫자로도 표시 가능 (8 진수)

소유자(User)			그룹(Group)			그 외 사용자(Other)		
r	w	-	r	-	-	r	-	-
4	2	0	4	0	0	4	0	0
6			4			4		

- 파일과 디렉터리의 소유와 허가권 (3)

- chmod 명령

파일 허가권 변경 명령어

예) # chmod 777 sample.txt

- 파일 소유권(Ownership)

파일을 소유한 사용자와 그룹을 의미

- chown/chgrp 명령

파일의 소유권을 바꾸는 명령어 (root 사용자만 사용가능)

예) # chown ubuntu.ubuntu sample.txt 또는

소유자 소유그룹

chown ubuntu sample.txt 및 # chgrp ubuntu sample.txt

```

root@server:~#
root@server:~# ls -l test
-rw-r--r-- 1 root root 44  3월 11 16:29 test
root@server:~# whoami
root
root@server:~# ./test
bash: ./test: 허가 거부
root@server:~#

```

➔ 관리자 권한인 root 사용자로 test 파일 실행 시 허가 거부 발생

- nano 에디터로 test 파일 생성
- root 사용자로 nano 에디터로 생성된 test 파일 실행 > 허가 거부
- root 사용자 권한이 rw(읽기 쓰기) 만 허용, 실행 권한 없는 상태

```

root@server:~# chmod 755 test
root@server:~#
root@server:~# ls -l test
-rwxr-xr-x 1 root root 44  3월 11 16:29 test
root@server:~# ./test
./test: 줄 1: 안녕하세요: 명령어를 찾을 수 없음
backups cache crash lib local lock log mail metrics opt run snap spool tmp
root@server:~#

```

➔ 실행 권한 추가 후 실행 완료

- test 파일의 파일 허가권을 755 로 변경 ('chmod 755 test')

-rw-r--r-- > -rwx-r-x-r-x 로 변경됨

- 변경 후 root 사용자로 실행 완료

```

root@server:~# chown ubuntu test
root@server:~#
root@server:~# ls -l test
-rwxr-xr-x 1 ubuntu root 44  3월 11 16:29 test
root@server:~#
root@server:~# chgrp ubuntu test
root@server:~# ls -l test
-rwxr-xr-x 1 ubuntu ubuntu 44  3월 11 16:29 test
root@server:~#

```

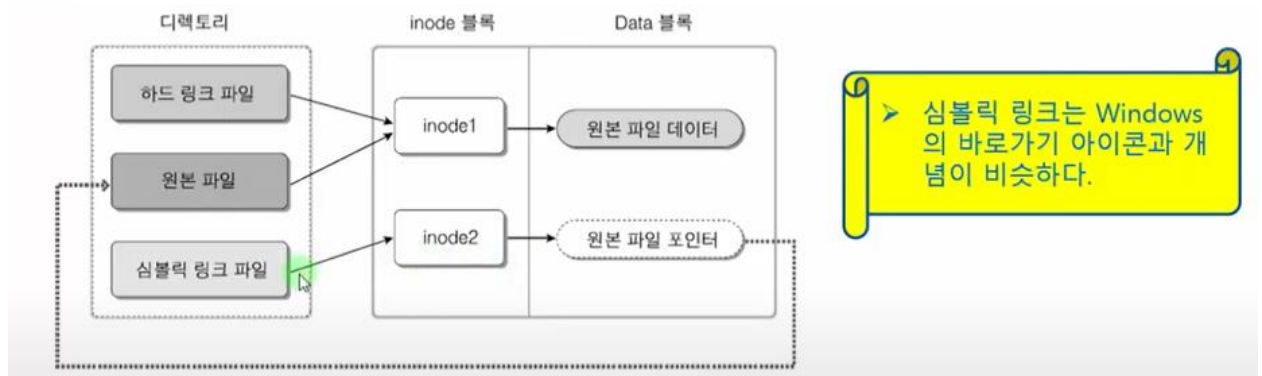
➔ ubuntu 사용자로 소유권 변경

- chown 명령어를 사용하여 test 파일의 소유권을 ubuntu 사용자로 변경

- chgrp 명령어를 사용하여 test 파일의 소유권을 ubuntu 그룹으로 변경

- 링크

- 파일의 링크(Link)에는 하드 링크(Hard Link)와 심볼릭 링크(Symbolic Link 또는 Soft Link) 두 가지가 있음



- 하드 링크를 생성하면 "하드 링크파일"만 하나 생성되며 같은 inode1을 사용 (명령: # ln 링크대상파일이름 링크파일이름)

- 심볼릭 링크를 생성하면 새로운 inode2를 만들고, 데이터는 원본 파일을 연결하는 효과 (명령 : # ln -s 링크대상파일이름 링크파일이름)

- linktest 디렉토리에 basefile 과 basefile 에 연결되는 hardlink, softlink 생성

- 연결 확인 후 basefile 을 다른 디렉토리로 이동 (mv 명령어)

```
root@server:~/linktest# mv basefile ../
root@server:~/linktest# ls
hardlink  softlink
root@server:~/linktest# cat hardlink
원본 파일입니다...
root@server:~/linktest# cat softlink
cat: softlink: 그런 파일이나 디렉터리가 없습니다
root@server:~/linktest#
```

- ➔ hardlink 는 작동되나 softlink 는 파일을 찾지 못함
- hardlink 는 원본 파일과 같은 inode1 에 연결하여 원본 파일 데이터와 연결된 반면 softlink 는 inode2 에 연결되어 원본 파일 포인터로 원본 파일을 찾아야 하기 때문에 다른 디렉터리로 이동한 원본 파일을 찾지 못함

```
root@server:~/linktest# mv ../basefile .
root@server:~/linktest# cat hardlink
원본 파일입니다...
root@server:~/linktest# cat softlink
원본 파일입니다...
root@server:~/linktest#
```

- ➔ 원본 파일이 원래 디렉터리(linktest) 로 돌아오니 softlink 도 정상 작동