

# Homework #6

1. jonahyun@jonahyun-virtual-machine:~\$ mkdir fs\_test  
jonahyun@jonahyun-virtual-machine:~\$ cd fs\_test  
jonahyun@jonahyun-virtual-machine:~/fs\_test\$ su  
Password:  
root@jonahyun-virtual-machine:/home/jonahyun/fs\_test# fdisk /dev/sda

Welcome to fdisk (util-linux 2.37.2).  
Changes will remain in memory only, until you decide to write them.  
Be careful before using the write command.

This disk is currently in use - repartitioning is probably a bad idea.  
It's recommended to umount all file systems, and swapoff all swap  
partitions on this disk.

Command (n for help): p

Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors  
Disk model: VMware Virtual S  
Units: sectors of 1 \* 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
Disklabel type: gpt  
Disk identifier: 166697C8-9AC4-467B-8E41-EA7721AC26DF

Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	4095	2048	1M	BIOS boot
/dev/sda2	4096	1054719	1050624	513M	EFI System
/dev/sda3	1054720	41940991	40886272	19.5G	Linux filesystem

Command (n for help): q

root@jonahyun-virtual-machine:/home/jonahyun/fs\_test# exit  
exit

- 디스크 'sda' 의 사양
  - : sectors => 41943040
  - : sector bytes => 512bytes
  - : I/O size => 512bytes
- 파티션의 개수 : 3개

# Homework #6

2.

```
jonahyun@jonahyun-virtual-machine:~/fs_test$ mkdir lntest
jonahyun@jonahyun-virtual-machine:~/fs_test$ echo "test ln" | cat > testln
jonahyun@jonahyun-virtual-machine:~/fs_test$ ln -s lntest lntest_s
jonahyun@jonahyun-virtual-machine:~/fs_test$ ln -s testln testln_s
jonahyun@jonahyun-virtual-machine:~/fs_test$ ls -l
total 8
drwxrwxr-x 2 jonahyun jonahyun 4096 4월 26 00:06 lntest
lrwxrwxrwx 1 jonahyun jonahyun 6 4월 26 00:09 lntest_s -> lntest
-rw-rw-r-- 1 jonahyun jonahyun 8 4월 26 00:08 testln
lrwxrwxrwx 1 jonahyun jonahyun 6 4월 26 00:09 testln_s -> testln
```

What are the files named “lntest\_s” and “testln\_s”? : 각각 lntest 파일과 testln을 가리키는 심볼릭 링크의 이름이다.

## Homework #6

3.

```
jonahyun@jonahyun-virtual-machine:~/fs_test$ cat testln
test ln
jonahyun@jonahyun-virtual-machine:~/fs_test$ cat testln_s
test ln
jonahyun@jonahyun-virtual-machine:~/fs_test$ echo "addition" >> testln
jonahyun@jonahyun-virtual-machine:~/fs_test$ echo "addition2" >> testln_s
jonahyun@jonahyun-virtual-machine:~/fs_test$ cat testln
test ln
addition
addition2
```

마지막 결과는 addition2이다.

testln\_s는 testln을 가르키고 있기 때문에 testln의 내용을 출력할 때 testln의 내용과 testln\_s의 내용이 함께 출력되기 때문이다.

# Homework #6

4.

```
jonahyun@jonahyun-virtual-machine:~/fs_test$ cd Intest_s
jonahyun@jonahyun-virtual-machine:~/fs_test/Intest_s$ touch b
jonahyun@jonahyun-virtual-machine:~/fs_test/Intest_s$ pwd
/home/jonahyun/fs_test/Intest_s
jonahyun@jonahyun-virtual-machine:~/fs_test/Intest_s$ cd ..
jonahyun@jonahyun-virtual-machine:~/fs_test$ cd Intest
jonahyun@jonahyun-virtual-machine:~/fs_test/Intest$ ls
b
jonahyun@jonahyun-virtual-machine:~/fs_test/Intest$ pwd
/home/jonahyun/fs_test/Intest
jonahyun@jonahyun-virtual-machine:~/fs_test/Intest$ s
```

- 첫번째 결과는

/home/jonahyun/fs\_test/Intest\_s이고

두번째 결과는

/home/jonahyun/fs\_test/Intest이다.

- b 이다. 이유는 Intest\_s는 Intest을 가르키고 있다. 따라서 "Intest\_s" 디렉토리 내부에 "b" 디렉토리를 생성하면, 실제로는 "Intest" 디렉토리 내부에 "b" 디렉토리가 생성되는 것이기 때문이다.

# Homework #6

5.

```
jonahyun@jonahyun-virtual-machine:~/fs_test$ ln testln testln_h
jonahyun@jonahyun-virtual-machine:~/fs_test$ echo "addition3" >> testln_h
jonahyun@jonahyun-virtual-machine:~/fs_test$ cat testln_s
test ln
addition
addition2
addition3
jonahyun@jonahyun-virtual-machine:~/fs_test$ ls -l
total 180
drwxrwxr-x 2 jonahyun jonahyun 4096 4월 26 00:13 disk
-rw-rw-r-- 1 jonahyun jonahyun 1433600 4월 26 00:36 disk.img
drwxrwxr-x 2 jonahyun jonahyun 4096 5월 1 17:39 lntest
lrwxrwxrwx 1 jonahyun jonahyun 6 4월 26 00:09 lntest_s -> lntest
-rw-rw-r-- 2 jonahyun jonahyun 37 5월 1 17:49 testln
-rw-rw-r-- 2 jonahyun jonahyun 37 5월 1 17:49 testln_h
lrwxrwxrwx 1 jonahyun jonahyun 6 4월 26 00:09 testln_s -> testln
```

- testln\_h : 생성될 하드 링크의 이름
- testln\_h는 하드링크를 생성한 것이고 testln\_s는 심볼릭 링크를 생성한 것이다.

하드 링크는 원본 파일과 링크된 파일이 같은 i-node를 가지고 있어 원본 파일과 링크된 파일이 완전히 같은 파일로 취급된다.

반면, 심볼릭 링크는 원본 파일이나 폴더를 가리키는 별도의 파일로, 원본 파일이나 폴더의 경로를 가지고 있다.

# Homework #6

6.

```
jonahyun@jonahyun-virtual-machine:~/fs_test$ mv testln testln_moved
jonahyun@jonahyun-virtual-machine:~/fs_test$ cat testln_h
test ln
addition
addition2
addition3
jonahyun@jonahyun-virtual-machine:~/fs_test$ cat testln_s
cat: testln_s: No such file or directory
jonahyun@jonahyun-virtual-machine:~/fs_test$ ls -l
total 180
drwxrwxr-x 2 jonahyun jonahyun 4096 4월 26 00:13 disk
-rw-rw-r-- 1 jonahyun jonahyun 1433600 4월 26 00:36 disk.img
drwxrwxr-x 2 jonahyun jonahyun 4096 5월 1 17:39 lntest
lrwxrwxrwx 1 jonahyun jonahyun 6 4월 26 00:09 lntest_s -> lntest
-rw-rw-r-- 2 jonahyun jonahyun 37 5월 1 17:49 testln_h
-rw-rw-r-- 2 jonahyun jonahyun 37 5월 1 17:49 testln_moved
lrwxrwxrwx 1 jonahyun jonahyun 6 4월 26 00:09 testln_s -> testln
jonahyun@jonahyun-virtual-machine:~/fs_test$ ls -li
538591 disk 538581 lntest 538583 testln_h 538587 testln_s
538589 disk.img 538585 lntest_s 538583 testln_moved
```

- 첫번째 cat은 하드 링크 이기 때문에 mv 명령어를 수행해도 testln 파일의 내용이 출력되지만, 두번째 cat은 심볼릭 링크 이기 때문에 이제 더 이상 원본 파일인 "testln"을 가리키지 않게 되어 "no such file"이라는 에러 메시지가 출력됩니다.
- ls -l과 ls -li를 출력한 결과 testln\_s는 Q 빨간색으로 출력되어 동작을 제대로 하지 않지만 testln\_h는 동작을 제대로 한다.

# Homework #6

7.

```
jonahyun@jonahyun-virtual-machine:~/fs_test$ dd if=/dev/zero of=disk.img bs=1024 count=1400
1400+0 records in
1400+0 records out
1433600 bytes (1.4 MB, 1.4 MiB) copied, 0.123015 s, 11.7 MB/s
jonahyun@jonahyun-virtual-machine:~/fs_test$ mke2fs -F disk.img
mke2fs 1.46.5 (30-Dec-2021)
Discarding device blocks: done
Creating filesystem with 350 4k blocks and 176 inodes

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

jonahyun@jonahyun-virtual-machine:~/fs_test$ mkdir disk
jonahyun@jonahyun-virtual-machine:~/fs_test$ ls -l disk
total 0
jonahyun@jonahyun-virtual-machine:~/fs_test$ su
Password:
root@jonahyun-virtual-machine:/home/jonahyun/fs_test# mount -o loop disk.img disk
root@jonahyun-virtual-machine:/home/jonahyun/fs_test# exit
exit
jonahyun@jonahyun-virtual-machine:~/fs_test$ ls -l disk
total 16
drwx----- 2 root root 16384 4월 26 00:13 lost+found
```

-dd의 기능: 블록 장치 간 데이터 복사를 수행하여 파일을 만든다.

-mke2fs의 기능: ext2, ext3 또는 ext4 파일 시스템을 생성한다.

-첫번째 ls와 두번째 ls의 차이: 첫 번째는 아무것도 뜨지 않지만 두 번째는 하나의 디렉토리가 있다.

이유는 “mount -o loop disk.img disk” 명령어로 인해 “disk.img” 파일 시스템이 “disk” 디렉토리에 마운트되며, 해당 디렉토리에서 파일 시스템의 파일 및 폴더를 탐색할 수 있기 때문이다.

# Homework #6

8.

```
jonahyun@jonahyun-virtual-machine:~/fs_test$ df | grep /dev/sda
/dev/sda3 19946096 13311224 5596332 71% /
/dev/sda2 524252 6216 518036 2% /boot/efi
jonahyun@jonahyun-virtual-machine:~/fs_test$ df -h | grep /dev/sda
/dev/sda3 20G 13G 5.4G 71% /
/dev/sda2 512M 6.1M 506M 2% /boot/efi
jonahyun@jonahyun-virtual-machine:~/fs_test$ df -i | grep /dev/sda
/dev/sda3 1277952 221107 1056845 18% /
/dev/sda2 0 0 0 - /boot/efi
jonahyun@jonahyun-virtual-machine:~/fs_test$ df . | grep /dev/sda
/dev/sda3 19946096 13311224 5596332 71% /
```

- -h : 사람이 이해하기 쉬운 단위로 출력한다.
- -i : inode 사용 현황을 출력한다. 사용가능한 i-node 수와 사용된 i-node 수를 보여준다.



# Homework #6

9.

```
jonahyun@jonahyun-virtual-machine: ~/fs_test$ su
Password:
root@jonahyun-virtual-machine: /home/jonahyun/fs_test# mount disk
mount: disk: not mounted.
root@jonahyun-virtual-machine: /home/jonahyun/fs_test# mount -o loop disk.img disk
root@jonahyun-virtual-machine: /home/jonahyun/fs_test# exit
exit
jonahyun@jonahyun-virtual-machine: ~/fs_test$ ls -l disk
total 16
drwx----- 2 root root 16384 4월 26 00:13 lost+found
jonahyun@jonahyun-virtual-machine: ~/fs_test$ su
Password:
root@jonahyun-virtual-machine: /home/jonahyun/fs_test# mount disk
root@jonahyun-virtual-machine: /home/jonahyun/fs_test# apt-get install quota
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  libnet-ldap-perl rpcbind default-mta | mail-transport-agent
The following NEW packages will be installed:
  quota
0 upgraded, 1 newly installed, 0 to remove and 104 not upgraded.
Need to get 213 kB of archives.
After this operation, 1,249 kB of additional disk space will be used.
Get:1 http://kr.archive.ubuntu.com/ubuntu jammy/main amd64 quota amd64 4.06-1build2 [213 kB]
Fetched 213 kB in 2s (93.0 kB/s)
Preconfiguring packages ...
Selecting previously unselected package quota.
(Reading database ... 281166 files and directories currently installed.)
Preparing to unpack .../quota_4.06-1build2_amd64.deb ...
Unpacking quota (4.06-1build2) ...
Setting up quota (4.06-1build2) ...
Processing triggers for man-db (2.10.2-1) ...
root@jonahyun-virtual-machine: /home/jonahyun/fs_test# mount -o loop,rw,usrquota,grpquota disk.in
9 disk
root@jonahyun-virtual-machine: /home/jonahyun/fs_test# mkdir disk/shared
root@jonahyun-virtual-machine: /home/jonahyun/fs_test# chmod 777 disk/shared
root@jonahyun-virtual-machine: /home/jonahyun/fs_test# chgrp neverland disk/shared
root@jonahyun-virtual-machine: /home/jonahyun/fs_test# chmod g+s,o+t disk/shared
root@jonahyun-virtual-machine: /home/jonahyun/fs_test# quotacheck -cug disk
root@jonahyun-virtual-machine: /home/jonahyun/fs_test# ls -l disk
total 36
-rw----- 1 root root 7168 5월 1 18:25 aquota.group
-rw----- 1 root root 6144 5월 1 18:25 aquota.user
drwx----- 2 root root 16384 4월 26 00:13 lost+found
drwxrwxrwt 2 root neverland 4096 5월 1 18:19 [shared]
```

- shared의 속성 : permission :3777

루트 권한을 가지고 neverland라는 그룹에 그룹소유권이 있다.

소유자(owner)는 읽기, 쓰기, 실행 권한이 있고, 그룹(group)은 읽기, 쓰기 권한이 있으며, setuid와 setgid가 설정되어 실행 파일에 대한 특수한 실행 권한이 부여되고,

다른 사용자(other)는 읽기, 쓰기 권한이 있으며, sticky bit가 설정되어 디렉토리 내 파일의 삭제 권한이 제한된다.

- "quotacheck"는 Linux 시스템에서 디스크 사용량을 검사하고, 사용자별(quota), 그룹별(group quota) 및 파일 시스템 전체(overall)에 대한 용량 제한 정보를 설정하는 명령어이다.

# Homework #6

10

```
root@jonahyun-virtual-machine:/home/jonahyun/fs_test# repquota -v disk
*** Report for user quotas on device /dev/loop5
Block grace time: 7days; Inode grace time: 7days
```

User		Block limits				File limits			
		used	soft	hard	grace	used	soft	hard	grace
root	--	24	0	0		3	0	0	
peterpan	--	0	2	4		0	2	4	
hook	--	0	2	4		0	2	4	

```
Statistics:
Total blocks: 7
Data blocks: 1
Entries: 3
Used average: 3.000000

root@jonahyun-virtual-machine:/home/jonahyun/fs_test# repquota -vg disk
*** Report for group quotas on device /dev/loop5
Block grace time: 7days; Inode grace time: 7days
```

Group		Block limits				File limits			
		used	soft	hard	grace	used	soft	hard	grace
root	--	20	0	0		2	0	0	
neverland	--	4	4	6		1	4	6	

```
Statistics:
Total blocks: 7
Data blocks: 1
Entries: 2
Used average: 2.000000
```

- setquota는 리눅스에서 사용자 또는 그룹에 대한 디스크 할당량을 설정하는 명령어이다.
  - -u 옵션은 사용자에게 대해서 디스크 할당량을 절하고
  - -g 옵션은 그룹에 대해서 디스크 할당량을 설정한다.
  - 2 4: 소프트 제한(Soft limit)을 2개의 파일로 하고 하드 제한(Hard limit)을 4개의 파일로 합니다. 이는 사용자가 생성할 수 있는 파일의 개수를 제한하는 것을 의미한다.
- > 소프트 제한은 사용자가 할당량을 초과하더라도 일시적으로 사용이 가능한 용량을 의미하며, 하드 제한은 사용자가 할당량을 초과할 수 없는 최대 용량을 의미한다.
- > 4 6 4 6 인자도 비슷한 맥락이다.

# Homework #6

11.

```
root@jonahyun-virtual-machine:/home/jonahyun/fs_test# quotaon disk
root@jonahyun-virtual-machine:/home/jonahyun/fs_test# su peterpan
peterpan@jonahyun-virtual-machine:/home/jonahyun/fs_test$ touch disk/shared/a
peterpan@jonahyun-virtual-machine:/home/jonahyun/fs_test$ touch disk/shared/b
peterpan@jonahyun-virtual-machine:/home/jonahyun/fs_test$ touch disk/shared/c
peterpan@jonahyun-virtual-machine:/home/jonahyun/fs_test$ quota peterpan
Disk quotas for user peterpan (uid 1001):
    Filesystem  blocks    quota   limit   grace   files   quota   limit   grace
    /dev/loop5      0         2         4         3*         2         4    7days
peterpan@jonahyun-virtual-machine:/home/jonahyun/fs_test$ touch disk/shared/d
peterpan@jonahyun-virtual-machine:/home/jonahyun/fs_test$ touch disk/shared/e
touch: cannot touch 'disk/shared/e': Disk quota exceeded
peterpan@jonahyun-virtual-machine:/home/jonahyun/fs_test$ quota -g neverland
Disk quotas for group neverland (gid 1003):
    Filesystem  blocks    quota   limit   grace   files   quota   limit   grace
    /dev/loop5      4         4         6         5*         4         6    7days
peterpan@jonahyun-virtual-machine:/home/jonahyun/fs_test$ exit
exit
root@jonahyun-virtual-machine:/home/jonahyun/fs_test#
```

- 소프트웨어 리미트: 사용자가 자신의 하드 디스크 공간 및 파일 수 사용량을 확인할 수 있는 경고 기능이다. 사용자는 소프트웨어 리미트를 초과하면 경고 메시지를 받고, 일부 작업을 계속 수행할 수 있다.
- 하드 리미트: 사용자의 하드 디스크 공간 및 파일 수 사용량을 제한하는 제한 기능이다. 사용자는 하드 리미트를 초과하면 파일을 생성하거나 변경할 수 없다.
- 따라서 소프트웨어 리미트는 사용자에게 경고를 주는 역할을 하고, 하드 리미트는 사용자의 자원 사용량을 제한하는 역할을 한다.

# Homework #6

12.

```
root@jonahyun-virtual-machine:/home/jonahyun/fs_test# su hook
hook@jonahyun-virtual-machine:/home/jonahyun/fs_test$ touch disk/shared/h1
hook@jonahyun-virtual-machine:/home/jonahyun/fs_test$ touch disk/shared/h2
touch: cannot touch 'disk/shared/h2': Disk quota exceeded
hook@jonahyun-virtual-machine:/home/jonahyun/fs_test$ quota hook
Disk quotas for user hook (uid 1002):
    Filesystem  blocks    quota   limit   grace   files   quota   limit   grace
    /dev/loop5      0         2       4         1         2         4
hook@jonahyun-virtual-machine:/home/jonahyun/fs_test$ quota -g neverland
Disk quotas for group neverland (gid 1003):
    Filesystem  blocks    quota   limit   grace   files   quota   limit   grace
    /dev/loop5      4         4       6         6*        4         6   6days
hook@jonahyun-virtual-machine:/home/jonahyun/fs_test$ exit
exit
root@jonahyun-virtual-machine:/home/jonahyun/fs_test# quotaoff disk
root@jonahyun-virtual-machine:/home/jonahyun/fs_test# du -h disk
16K    disk/lost+found
4.0K    disk/shared
40K    disk
root@jonahyun-virtual-machine:/home/jonahyun/fs_test# umount disk
```

- h1은 잘 만들어졌으나 h2는 에러 메시지가 뜨며 파일이 생성되지 않았다. 이유는 hook의 용량은 그룹의 하드 제한을 통해 최대 6개 까지만 파일을 생성할 수 있는데 h2를 만드면 7개로 용량 초과 이기 때문이다.
- du 명령어는 디렉토리 또는 파일의 용량을 보여주는 명령어이다.

# Homework #6 - problems

1.

```
-dd if=/dev/zero  
of=backup_disk.img bs=1024  
count=10000  
-mkfs.ext4 -b 1024 backup_disk.img  
-mkdir backup_disk  
-mount -t ext4 -o loop  
backup_disk.img backup_disk
```

```
root@jonahyun-virtual-machine:/home/jonahyun# ls -hl backup_disk.img  
-rw-r--r-- 1 root root 10M 5월 1 19:20 backup_disk.img  
root@jonahyun-virtual-machine:/home/jonahyun# blkid backup_disk.img  
backup_disk.img: UUID="963977a9-61ac-4bb9-8dfe-f7e4c5cb9284" BLOCK_SIZE="1024" T  
YPE="ext4"  
root@jonahyun-virtual-machine:/home/jonahyun# ls -hl backup_disl  
ls: cannot access 'backup_disl': No such file or directory  
root@jonahyun-virtual-machine:/home/jonahyun# ls -hl backup_disk  
total 12K  
drwx----- 2 root root 12K 5월 1 19:14 lost+found  
root@jonahyun-virtual-machine:/home/jonahyun# df -T backup_disk  
Filesystem      Type 1K-blocks  Used Available Use% Mounted on  
/dev/loop5      ext4    8409      14      7679     1% /home/jonahyun/backup_disk
```

## Homework #6 - problems

1.

“ls -hl backup\_disk.img”:  
“backup\_disk.img”는 용량이 10MB  
이다.

“blkid backup\_disk.img” :파일시스  
템이 “ext4” 인 것을 확인

“ls -hl backup\_disk” :backup\_disk  
의 권한이 root가 된 것을 확인

“df -T backup\_disk” : 디스크 사용  
현황 파악

- 소유권 변경 : jonahyun 에서  
root로 변경됨
- 사이즈 변경 : 16K로 늘어남.

## Homework #6 - problems

2.

-"mount" 명령에서 "sync" 옵션은 파일 시스템의 동기화 방식을 설정하는 옵션이다.

이 옵션을 사용하면 파일 시스템이 동기식으로 마운트되어 디스크에 데이터가 즉시 기록된다.

이것은 데이터 손실을 방지하고 파일 시스템이 불안정한 경우에도 데이터의 일관성을 유지하는 데 도움이 된다.

-"sync" 옵션은 데이터의 무결성이나 안정성이 중요한 시스템에서 사용하는 것이 좋다.

예를 들어 중요한 서버, 데이터베이스 또는 파일 서버에서 사용할 수 있다.

# Homework #6 - problems

3.

1. `dd if=/dev/zero of=vfat_disk.img  
bs=1042 count=10240`

2. `mkfs -t fat vfat_disk.img`

3. `sudo mount -o  
loop,uid=1000,gid=1000  
vfat_disk.img backup_disk`

4.

```
jonahyun@jonahyun-virtual-machine:~$ ln c c_h
jonahyun@jonahyun-virtual-machine:~$ ln -s c c_s
jonahyun@jonahyun-virtual-machine:~$ ln backup_disk/a a_h
ln: failed to create hard link 'a_h' => 'backup_disk/a': Invalid cross-device link
jonahyun@jonahyun-virtual-machine:~$ ln -s backup_disk/a a_s
jonahyun@jonahyun-virtual-machine:~$ ln backup_disk/a backup_disk/a_h
ln: failed to create hard link 'backup_disk/a_h': File exists
jonahyun@jonahyun-virtual-machine:~$ ln -s backup_disk/a backup_disk/a_s
ln: failed to create symbolic link 'backup_disk/a_s': File exists
jonahyun@jonahyun-virtual-machine:~$ ln backup_disk_2/b backup_disk/b_h
ln: failed to create hard link 'backup_disk/b_h' => 'backup_disk_2/b': Invalid cross-device link
jonahyun@jonahyun-virtual-machine:~$ ln backup_disk_2/b backup_disk_2/b_h
jonahyun@jonahyun-virtual-machine:~$ ln -s backup_disk_2/b backup_disk/b_s
jonahyun@jonahyun-virtual-machine:~$
```



# Homework #6 - problems

5.

- 1) 소프트 링크는 링크 파일을 통해 원본 파일에 대한 경로를 참조한다. 즉, 링크 파일은 원본 파일에 대한 새로운 이름을 가리키는데, 링크 파일이 가리키는 원본 파일이 삭제되면 링크 파일은 무효화된다. 반면, 하드 링크는 원본 파일 자체에 대한 링크를 만든다. 원본 파일과 하드 링크 파일은 동일한 inode를 공유하며, 이는 둘 중 하나가 삭제되더라도 다른 파일은 여전히 원본 파일을 가리키는 것을 의미한다.
- 2) 소프트 링크는 다른 파일 시스템에 대한 링크를 만들 수 있다. 하드 링크는 같은 파일 시스템에만 적용된다.
- 3) 소프트 링크는 원본 파일과 링크 파일 간의 차이를 유지하기 위해 ls 명령어로 확인할 때, 링크 파일 이름 앞에 l 문자가 나타난다. 하드 링크는 원본 파일과 링크 파일 간의 차이가 없기 때문에 파일 유형이나 이름 앞에 표시되는 문자가 달라지지 않는다.
- 4) 원본 파일에 대한 하드 링크는 일반적으로 파일 이름의 개수를 줄이고 디스크 공간을 절약할 수 있다. 그러나 원본 파일을 삭제하기 위해서는 모든 하드 링크를 제거해야 하기 때문에 사용에 주의해야 한다. 소프트 링크는 하드 링크보다 덜 효율적이지만, 파일 시스템 간의 링크 생성을 가능하게 한다.

- 하드 링크는 일반 복사 파일과는 달리 원본 파일과 동일한 inode를 가지며, 해당 inode의 레퍼런스 카운트를 증가시켜 파일을 가리키는 다른 이름을 만드는 것이다.