

Homework #6

1. Start a new terminal, do “mkdir fs_test”, then do “cd fs_test”, then do “su” (password required), then do “fdisk /dev/sda”, then press “p”, then press ”q”, then do “exit”

- Take a screenshot

▪

```
mtak@mtak-virtual-machine:~$ mkdir fs_test
mtak@mtak-virtual-machine:~$ cd fs_test/
mtak@mtak-virtual-machine:~/fs_test$ su
Password:
root@mtak-virtual-machine:/home/mtak/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 (m for help): p

Disk /dev/sda: 100 GiB, 107374182400 bytes, 209715200 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: 84096C1E-3CD7-48A3-A311-5E4464ECA1A



| 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 | 209713151 | 208658432 | 99.5G | Linux filesystem |



Command (m for help): q

root@mtak-virtual-machine:/home/mtak/fs_test# exit
exit
mtak@mtak-virtual-machine:~/fs_test$
```

- What is the specification of the disk “sda”? (sectors, sector bytes, I/O size)

20971500 개의 섹터와, 그에따라 $512 * 20971500 = 107374182400$ sector bytes 를 가지고, I/O size 는 512bytes/ 512bytes 를 가진다.

- How many partitions are there?

3개

2. Do “clear”, then do “mkdir Intest”, then do “echo “test ln” | cat > testln”, then do “ln -s Intest Intest_s”, then do “ln -s testln testln_s”, then do “ls -l”

- Take a screenshot

```
mtak@mtak-virtual-machine:~/fs_test$ mkdir Intest
mtak@mtak-virtual-machine:~/fs_test$ echo test ln | cat > testln
mtak@mtak-virtual-machine:~/fs_test$ ln -s Intest/ Intest_s
mtak@mtak-virtual-machine:~/fs_test$ ln -s testln testln_s
mtak@mtak-virtual-machine:~/fs_test$ ls -l
total 8
drwxrwxr-x 2 mtak mtak 4096 5월 2 11:32 Intest
lrwxrwxrwx 1 mtak mtak 7 5월 2 11:33 Intest_s -> Intest/
-rw-rw-r-- 1 mtak mtak 8 5월 2 11:32 testln
lrwxrwxrwx 1 mtak mtak 6 5월 2 11:33 testln_s -> testln
mtak@mtak-virtual-machine:~/fs_test$
```

- What are the files named “Intest_s” and “testln_s”?

Intest_s 는 Intest 디렉토리를 가리키는 심볼릭 링크, testln_s 는 testln 파일을 가리키는 심볼릭 링크이다.

3. Do “clear”, then do “cat testln”, then do “cat testln_s”, then do “echo “addition” >> testln”, then do “echo “addition2” >> testln_s”, then do “cat testln”

- Take a screenshot

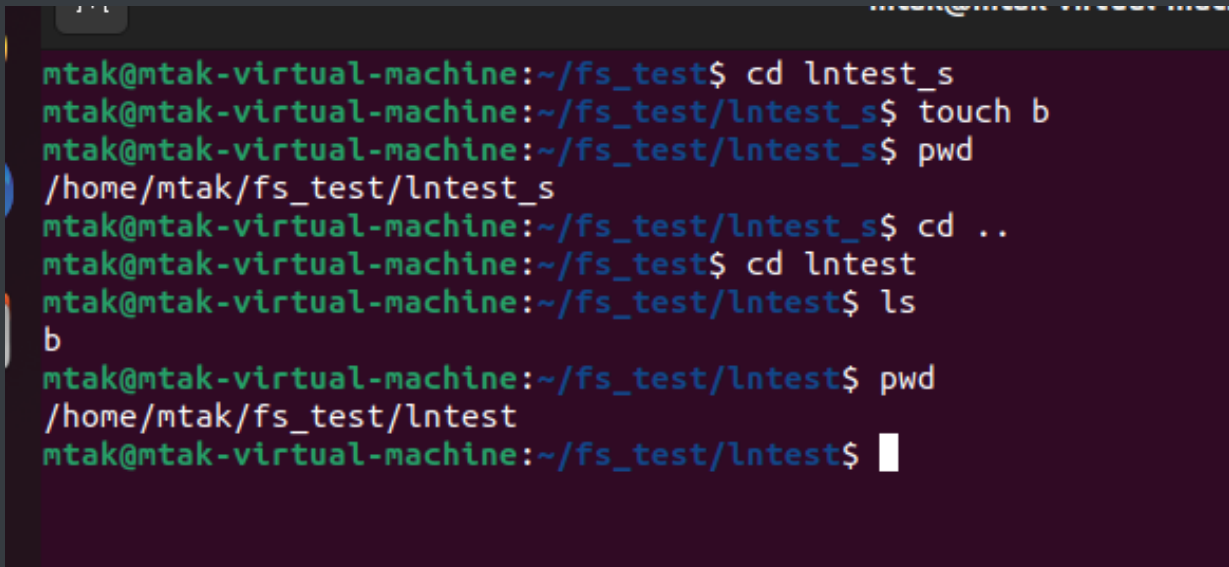
```
mtak@mtak-virtual-machine: ~/fs_test
mtak@mtak-virtual-machine:~/fs_test$ cat testln
test ln
mtak@mtak-virtual-machine:~/fs_test$ cat testln_s
test ln
mtak@mtak-virtual-machine:~/fs_test$ echo addition >> testln
mtak@mtak-virtual-machine:~/fs_test$ echo "addition2" >> testln_s
mtak@mtak-virtual-machine:~/fs_test$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

- What is the last result? Why?

testln_s 는 testln 을 가리키는 심볼릭 링크이기 때문에, 심볼릭 링크 testln_s 로 접근하여도 값이 변경된다.

4. Do “clear”, then do “cd lntest_s”, then do “touch b”, then do “pwd”, then do “cd ..”, then do “cd lntest”, then do “ls”, then do “pwd”

- Take a screenshot



```
mtak@mtak-virtual-machine:~/fs_test$ cd lntest_s
mtak@mtak-virtual-machine:~/fs_test/lntest_s$ touch b
mtak@mtak-virtual-machine:~/fs_test/lntest_s$ pwd
/home/mtak/fs_test/lntest_s
mtak@mtak-virtual-machine:~/fs_test/lntest_s$ cd ..
mtak@mtak-virtual-machine:~/fs_test$ cd lntest
mtak@mtak-virtual-machine:~/fs_test/lntest$ ls
b
mtak@mtak-virtual-machine:~/fs_test/lntest$ pwd
/home/mtak/fs_test/lntest
mtak@mtak-virtual-machine:~/fs_test/lntest$
```

- Compare the results between first “pwd” and second “pwd”.

첫번째 .pwd 는 심볼릭링크 lntest_s 로 디렉토리에 접근했다. 결과 :

/home/mtak/fs_test/lntest_s

두번째 pwd 는 lntest 에 직접 접근했다.

결과 : /home/mtak/fs_test/lntest_s

- What is the result of “ls”? Why?

심볼릭링크 lntest_s 로 접근하여 파일 b 를 만들었으므로 , lntest 에 파일 b 가 있다. ls 결과 : b

5. Do “cd ..”, then do “clear”, then do “ln testln testln_h”, then do “echo “addition3” >> testln_h”, then do “cat testln_s”, then do “ls -l”

- Take a screenshot

```
mtak@mtak-virtual-machine:~/fs_test$ ln testln testln_h
mtak@mtak-virtual-machine:~/fs_test$ echo addition3 >> testln_h
mtak@mtak-virtual-machine:~/fs_test$ cat testln_h
test ln
addition
addition2
addition3
mtak@mtak-virtual-machine:~/fs_test$ cat testln_s
test ln
addition
addition2
addition3
mtak@mtak-virtual-machine:~/fs_test$ ls -l
total 12
drwxrwxr-x 2 mtak mtak 4096 5월 2 11:36 lntest
lrwxrwxrwx 1 mtak mtak 7 5월 2 11:33 lntest_s -> lntest/
-rw-rw-r-- 2 mtak mtak 37 5월 2 11:40 testln
-rw-rw-r-- 2 mtak mtak 37 5월 2 11:40 testln_h
lrwxrwxrwx 1 mtak mtak 6 5월 2 11:33 testln_s -> testln
mtak@mtak-virtual-machine:~/fs_test$
```

- What is the file “testln_h”?

하드 링크(hard link) 이며, testln 을 가리킨다.

- What is the difference between “testln_s” and “testln_h”

testln_s 는 심볼릭 링크이며 testln_h 는 하드링크이다.

추가로 명령어 ls -i 를 넣어서 확인해보면, testln_h 와 testln 의 inode 값은 2359934로 같은 것을 알 수 있다.

하지만 testln_s 는 inode 값이 2359936으로 다르다. 그이유는

testln_s 는 directory entry 가 파일(testln)의 이름을 가리키는 반면 Hard Link 는 directory entry 가 파일(testln)의 inode 를 가리킨다.

6. Do “clear”, then do “mv testln testln_moved”, then do “cat testln_h”, then do “cat testln_s”, then do “ls -l”, then do “ls -i”

- Take a screenshot

```

mtak@mtak-virtual-machine: ~/fs_test
mtak@mtak-virtual-machine:~/fs_test$ mv testln testln_moved
mtak@mtak-virtual-machine:~/fs_test$ cat testln_h
test ln
addition
addition2
addition3
mtak@mtak-virtual-machine:~/fs_test$ cat testln_s
cat: testln_s: No such file or directory
mtak@mtak-virtual-machine:~/fs_test$ ls -l
total 12
drwxrwxr-x 2 mtak mtak 4096 5월 2 11:36 lntest
lrwxrwxrwx 1 mtak mtak 7 5월 2 11:33 lntest_s -> lntest/
-rw-rw-r-- 2 mtak mtak 37 5월 2 11:40 testln_h
-rw-rw-r-- 2 mtak mtak 37 5월 2 11:40 testln_moved
lrwxrwxrwx 1 mtak mtak 6 5월 2 11:33 testln_s -> testln
mtak@mtak-virtual-machine:~/fs_test$ ls -li
2359932 lntest 2359935 lntest_s 2359934 testln_h 2359934 testln_moved 2359936 testln_s
mtak@mtak-virtual-machine:~/fs_test$

```

- Compare the results between the first “cat” and the second “cat”

첫번째 cat 은 하드링크를 통한 접근이다. inode 를 통한 접근이어서 출력이 가능하다.

두번째 cat 은 접근에 실패했다. 그 이유는 심볼릭 링크는 이전 파일의 이름 testln 을 가리키므로 변경된 testln_moved 에 접근하지 못했다.

- What is the difference between “testln_s” and “testln_h” from the results of “ls -l” and “ls -li”?

testln_s 는 가리키고 있는 파일의 이름을 보여준다. 하지만 testln_h 는 보여주지 않는다.

그 이유는 testln_h 는 inode 를 직접적으로 가리키고 있기 때문이다.

그리고 그림을 보면 testln_s 는 빨간색인데, 이 빨간색은 더이상 링크가 제 기능을 못한다라는 뜻이다.

그 이유는 testln 이 이름이 바뀌었기 때문이다.

ls -li 의 결과를 보면 , testln_h 와 testln_moved 의 inode 가 2359934로 일치하는 것을 볼 수 있다.

하지만 testln_s 는, inode 가 2359936으로 다르다.

- Do “clear”, then do “dd if=/dev/zero of=disk.img bs=1024 count=1400”, then do “mke2fs -F disk.img”, then do “mkdir disk”, then do “ls -l disk”, then do “su” (password required), then do “mount -o loop disk.img disk”, then do “exit”, then do “ls -l disk”

- Take a screenshot

```
mtak@mtak-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, 0.00252084 s, 569 MB/s
mtak@mtak-virtual-machine:~/fs_test$ mke2fs -F disk.img
mke2fs 1.46.5 (30-Dec-2021)
mke2fs: invalid blocks 'disk.img' on device '-F'
mtak@mtak-virtual-machine:~/fs_test$ df -h
!

```

mke2fs에 -F 가 없어서, mkfs로 대체했다.

```
mtak@mtak-virtual-machine:~/fs_test$ mkfs 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

mtak@mtak-virtual-machine:~/fs_test$
mtak@mtak-virtual-machine:~/fs_test$ ls -l disk
total 0
mtak@mtak-virtual-machine:~/fs_test$ su

```

```
root@mtak-virtual-machine:/home/mtak/fs_test# mount -o loop disk.img disk
root@mtak-virtual-machine:/home/mtak/fs_test# exit
exit
mtak@mtak-virtual-machine:~/fs_test$ ls -l disk
total 16
drwx----- 2 root root 16384 5월 3 09:50 lost+found
mtak@mtak-virtual-machine:~/fs_test$

```

- What is the functionality of “dd” command?
블록 단위로 파일을 복사하거나 파일을 변환할수 있는 명령어
이때 용량은 1024byte X 1400 으로, 약 1.4MB 이다
 - What is the functionality of “mke2fs” command?
리눅스에서 파티션 생성후, 주 파일 시스템인 ext2, 3, 4 타입의 리눅스 파일시스템을 생성하는 명령어이다. 윈도우에서 포맷이랑 비슷하다.
 - Compare the difference between the results of the first “ls” and the second “ls”. Why?
첫번째 ls 는 마운트가 안된 상황이다. 따라서 빈 디렉토리이지만,
두번째 ls 는 disk.img 를 disk 에
loop 로 마운트 했기 때문에, lost+found 라는 디렉토리가 나타났다.
8. Do “clear”, then do “df | grep /dev/sda”, then do “df -h | grep /dev/sda”, then “df -i | grep /dev/sda”, then do “df . | grep /dev/sda”

- Take a screenshot

```
mtak@mtak-virtual-machine:~/fs_test$ df | grep /dev/sda
/dev/sda3      19946096 10737312   8170244   57% /
/dev/sda2         524252    6216    518036    2% /boot/efi
mtak@mtak-virtual-machine:~/fs_test$ df -h | grep /dev/sda
/dev/sda3      20G   11G   7.8G   57% /
/dev/sda2      512M   6.1M   506M    2% /boot/efi
mtak@mtak-virtual-machine:~/fs_test$ df -i | grep /dev/sda
/dev/sda3      1277952 216412 1061540   17% /
/dev/sda2         0         0         0    - /boot/efi
mtak@mtak-virtual-machine:~/fs_test$ df . | grep /dev/sda
/dev/sda3      19946096 10737312   8170244   57% /
mtak@mtak-virtual-machine:~/fs_test$
```

- What are the options “-h” and “-i” for?

-h 는 사람이 보기 좋게 출력해준다. 옵션 없이 출력했을 때는 사람이 직관적으로 용량을 알기 어렵기 때문에,

-h 옵션으로 GB 나 MB 등 바이트 단위가 붙어서 보기 좋게 출력해준다.

-i 는 inode 의 남은공간, 사용공간, 사용 퍼센트를 출력한다.

9. Do “su”(password required), then do “umount disk”, then do “apt-get install quota” (press Y for all questions, Internet connection required!!), then do “mount -o loop,rw,usrquota,grpquota disk.img disk”, then do “mkdir disk/shared”, then do “chmod 777 disk/shared”, then do “chgrp neverland disk/shared”, then do “chmod g+s,o+t disk/shared”, do “quotacheck -cug disk”, then do “ls -l disk”. (Assumption : There are users named as peterpan and hook. Both users are members of a group named as neverland.)

- Take a screenshot

```
mtak@mtak-virtual-machine:~/fs_test$ su
Password:
root@mtak-virtual-machine:/home/mtak/fs_test# umount disk
root@mtak-virtual-machine:/home/mtak/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 95 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 (132 kB/s)
Preconfiguring packages ...
Selecting previously unselected package quota.
(Reading database ... 197894 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@mtak-virtual-machine:/home/mtak/fs_test# moun -o loop,rw,usrquota,grpquota disk.img disk
Command 'moun' not found, did you mean:
  command 'muon' from deb muon (4:5.8.0-1ubuntu5)
  command 'mon' from deb mon (1.3.6-2)
  command 'mount' from deb mount (2.37.2-4ubuntu3)
Try: apt install <deb name>
root@mtak-virtual-machine:/home/mtak/fs_test# mount -o loop,rw,usrquota,grpquota disk.img disk
root@mtak-virtual-machine:/home/mtak/fs_test# mkdir disk/shared
root@mtak-virtual-machine:/home/mtak/fs_test# chmod 777 disk/shared/
root@mtak-virtual-machine:/home/mtak/fs_test# chgrp neverland disk/shared/
chgrp: invalid group: 'neverland'
root@mtak-virtual-machine:/home/mtak/fs_test# chmod g+s,o+t disk/shared/
root@mtak-virtual-machine:/home/mtak/fs_test# quotacheck -cug disk
root@mtak-virtual-machine:/home/mtak/fs_test# ls -l disk
total 36
-rw----- 1 root root 6144 5월 3 12:12 aquota.group
-rw----- 1 root root 6144 5월 3 12:12 aquota.user
drwx----- 2 root root 16384 5월 3 12:08 lost+found
drwxrwsrwt 2 root root 4096 5월 3 12:11 shared
root@mtak-virtual-machine:/home/mtak/fs_test# useradd peterpand -m -s /bin/bash
root@mtak-virtual-machine:/home/mtak/fs_test# useradd hook -m -s /bin/bash
root@mtak-virtual-machine:/home/mtak/fs_test# groupadd neverland
root@mtak-virtual-machine:/home/mtak/fs_test# usermod -a -G neverland peterpan
usermod: user 'peterpan' does not exist
root@mtak-virtual-machine:/home/mtak/fs_test# useradd peterpan -m -s /bin/bash -G neverland
root@mtak-virtual-machine:/home/mtak/fs_test# usermod -a -G neverland hook
root@mtak-virtual-machine:/home/mtak/fs_test# █

root@mtak-virtual-machine:/home/mtak/fs_test# chgrp neverland disk/shared/
root@mtak-virtual-machine:/home/mtak/fs_test# ls -l
total 192
drwxr-xr-x 4 root root 4096 5월 3 12:12 disk
-rw-rw-r-- 1 mtak mtak 1433600 5월 3 12:12 disk.img
root@mtak-virtual-machine:/home/mtak/fs_test# ls -l disk
total 36
-rw----- 1 root root 6144 5월 3 12:12 aquota.group
-rw----- 1 root root 6144 5월 3 12:12 aquota.user
drwx----- 2 root root 16384 5월 3 12:08 lost+found
drwxrwsrwt 2 root neverland 4096 5월 3 12:11 shared
root@mtak-virtual-machine:/home/mtak/fs_test#
```

- What are the properties of the new directory shared?

새로운 shared 는 접근 권한이 rwxrwsrwt 이고(모든 사람이 rwx 가능) guid 와 sticky 비트가 설정되어있다. 그리고 그룹 소유권이 root 에서 neverland 로 바뀌었다.

- What is the functionality of “quotacheck”?

기록파일을 가장 최근의 상태로 업데이트 시키는 명령어이다.

-c 옵션은 기존의 생성된 쿼터 파일을 읽지 않는다.

-u 옵션은 사용자 쿼터 관련 체크를 한다.

-g 옵션은 그룹 쿼터 관련 체크를 한다.

- Do “clear”, then do “setquota -u peterpan 2 4 2 4 disk”, then do “setquota -u hook 2 4 2 4 disk”, then do “setquota -g neverland 4 6 4 6 disk”, then do “repquota -v disk”, then do “repquota -vg disk”

- Take a screenshots for repquota commands

```
root@mtak-virtual-machine:/home/mtak/fs_test# setquota -u peterpan 2 4 2 4 disk
root@mtak-virtual-machine:/home/mtak/fs_test# do setquota -u hook 2 4 2 4 disk
bash: syntax error near unexpected token `do'
root@mtak-virtual-machine:/home/mtak/fs_test# setquota -u hook 2 4 2 4 disk
root@mtak-virtual-machine:/home/mtak/fs_test# setquota -g neverland 4 6 4 6 disk
root@mtak-virtual-machine:/home/mtak/fs_test# repquota -v disk
Command 'repquota' not found, did you mean:
  command 'repquota' from deb quota (4.06-1build2)
Try: apt install <deb name>
root@mtak-virtual-machine:/home/mtak/fs_test# repquota -v disk
*** Report for user quotas on device /dev/loop16
Block grace time: 7days; Inode grace time: 7days
```

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

```

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

root@mtak-virtual-machine:/home/mtak/fs_test# repquota -vg disk
*** Report for group quotas on device /dev/loop16
Block grace time: 7days; Inode grace time: 7days
```

Group		used	Block limits			File limits			
			soft	hard	grace	used	soft	hard	grace
root	--	24	0	0		3	0	0	
neverland	--	0	4	6		0	4	6	

```

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

root@mtak-virtual-machine:/home/mtak/fs_test#
```

What is the functionality of “setquota”? Explain the options “-u” and “-g” and arguments.

-u 옵션 : 유저의 사용량 한도변경

-g 옵션 : 그룹의 사용량 한도변경

11. Do “clear”, then do “quotaon disk”, then do “su peterpan”, then do “touch disk/shared/a”, then do “touch disk/shared/b”, then do “touch disk/shared/c”, then do “quota peterpan”, then do “touch disk/shared/d”, then do “touch disk/shared/e”, then do “quota -g neverland”, then do “exit”

■ Take a screenshot

```
root@mtak-virtual-machine:/home/mtak/fs_test# quotaon disk
root@mtak-virtual-machine:/home/mtak/fs_test# su peterpan
peterpan@mtak-virtual-machine:/home/mtak/fs_test$ touch disk/shared/a
peterpan@mtak-virtual-machine:/home/mtak/fs_test$ touch disk/shared/b
peterpan@mtak-virtual-machine:/home/mtak/fs_test$ touch disk/shared/c
peterpan@mtak-virtual-machine:/home/mtak/fs_test$ quota peterpan
quota: Cannot resolve mountpoint path /home/mtak/fs_test/disk: Permission denied
peterpan@mtak-virtual-machine:/home/mtak/fs_test$ touch disk/shared/d
peterpan@mtak-virtual-machine:/home/mtak/fs_test$ touch disk/shared/e
touch: cannot touch 'disk/shared/e': Disk quota exceeded
peterpan@mtak-virtual-machine:/home/mtak/fs_test$ quota -g neverland
quota: Cannot resolve mountpoint path /home/mtak/fs_test/disk: Permission denied
peterpan@mtak-virtual-machine:/home/mtak/fs_test$ exit
exit
root@mtak-virtual-machine:/home/mtak/fs_test#
```

■ What is the difference between the soft limit and hard limit?

soft limit 는 사용자가 limit 를 초과할 수 있지만

hard limit 는 사용자가 limit 를 초과할 수 없다.

12. Do “clear”, then do “su hook”, then do “touch disk/shared/h1”, then do “touch disk/shared/h2”, then do “quota hook”, then do “quota -g neverland”, then do “exit”, then do “quotaoff disk”, then do “du -h disk”, then do “umount disk”

- Take a screenshot

```
root@mtak-virtual-machine:/home/mtak/fs_test# su hook
hook@mtak-virtual-machine:/home/mtak/fs_test$ touch disk/shrared/h1
touch: cannot touch 'disk/shrared/h1': No such file or directory
hook@mtak-virtual-machine:/home/mtak/fs_test$ touch disk/shrared/h1
touch: cannot touch 'disk/shrared/h1': No such file or directory
hook@mtak-virtual-machine:/home/mtak/fs_test$ touch disk/shared/h1
hook@mtak-virtual-machine:/home/mtak/fs_test$ touch disk/shared/h2
hook@mtak-virtual-machine:/home/mtak/fs_test$ quota hook
quota: Cannot resolve mountpoint path /home/mtak/fs_test/disk: Permission denied
hook@mtak-virtual-machine:/home/mtak/fs_test$ quota -g neverland
quota: Cannot resolve mountpoint path /home/mtak/fs_test/disk: Permission denied
hook@mtak-virtual-machine:/home/mtak/fs_test$ eit
Command 'eit' not found, did you mean:
  command 'vit' from deb vit (2.1.0-2)
  command 'git' from deb git (1:2.34.1-1ubuntu1.9)
  command 'edit' from deb mailcap (3.70+nmu1ubuntu1)
  command 'eid' from deb id-utils (4.6.28-20200521ss15dab)
  command 'eet' from deb libeet-bin (1.26.2-1)
  command 'ent' from deb ent (1.2debian-3)
  command 'wit' from deb wit (3.01a-4)
Try: apt install <deb name>
hook@mtak-virtual-machine:/home/mtak/fs_test$ exit
exit
root@mtak-virtual-machine:/home/mtak/fs_test# quotaoff disk
root@mtak-virtual-machine:/home/mtak/fs_test# du -h disk
16K      disk/lost+found
4.0K      disk/shared
40K      disk
root@mtak-virtual-machine:/home/mtak/fs_test# umount disk
root@mtak-virtual-machine:/home/mtak/fs_test#
```

- What happens during creating h1 and h2? Why?

h1 은 정상적으로 생성되었으나, h2 는 Disk quota exceeded. 즉 사용량을 초과해서 생성이 되지 않았다. quota 명령어로 hook 과 neverland 의 사용량을 출력한 결과, 그룹 할당량이 초과되어서 (전에 peterpan 이 만든 파일로 인해) hook 은 더 만들 수 있지만 더 이상 생성하지 못했다.

- What is the command “du” for?

Disk Usage 란 뜻의 명령어 이다. 지정된 디렉토리의 디스크 사용량을 표시하는 명령어 이다. -h 옵션은 사람이 보기 편하게 출력해준다. 따라서 용량 뒤에 단위가 붙은 것을 알 수 있다.

== Problems ==

1. Make a local file named as “backup_disk.img” and initialize an “ext4” filesystem in the file whose capacity is about 10MB with 1024 Byte block size. The filesystem in the “backup_disk.img” file is mounted on the local directory “backup_disk”.

- List the required commands
- su : root 권한으로 mount
dd if=/dev/zero of=backup_disk.img bs=1024 count=10240
: 1024 byte = 1KB , 1MB = 1024*1KB 이므로 count =10240
mkfs -F -t ext4 backup_disk.img : 파일시스템을 만드는 과정
mkdir backup_disk : 마운팅할 디렉토리
mount -o loop backup_disk.img backup_disk : loop 로 마운팅
exit : root 로 할일을 끝냈으므로 root 를 종료 시킴.
- Evaluate your operation by using “ls -hl backup_disk.img”, “blkid backup_disk.img”, “ls -hl backup_disk”, and “df -T backup_disk”
타입 ext4 로 정확하고, 이름 틀린거 없고, 용량도 틀리지 않았다.
- Describe how the ownership and size of backup_disk directory change.
root 로 로그인해서, chgrp 명령어와 chown 명령어로 ownership 을 변경할 수 있다.

2. What is the objective of “sync” option in “mount” command? When do we need to take care of using “sync” option?

파일 시스템의 입출력이 동기적으로 입/출력되게 설정한다.

3. There is a local file named as “vfat_disk.img” and initialize a “vfat” filesystem in the file whose capacity is about 10MB with 1024 Byte block size. The filesystem in the “vfat_disk.img” file is mounted on the local directory “backup_disk”. In here, you want to give ownership of this disk to a normal user whose UID is 1000 and GID is 1000. What are the required commands? (Hint : mkfs.fat, mount options for vfat: uid, gid, umask. Please try to use “man mount” to find this option)

su : root 권한

mkdir backup_disk(디렉토리가 존재한다면, backup_disk 의 상위 디렉토리로 이동.)

dd if=/dev/zero of=vfat_disk.img bs=1023 count=10240

mkfs -F -t vfat vfat_disk.img

chown mtak(UID1000 의 유저) backup_disk

chgrp mtak backup_disk

mount -o loop vfat_disk.img backup_disk

4. Let’s assume that a file “a” locates under “backup_disk”, a file “b” locates under “backup_disk_2” and a file “c” locates in the local directory. Evaluate whether the following operations correctly works or not. (Assumption: “backup_disk” and “backup_disk_2” are mounted on different media. The current directory is the parent

directory of “backup_disk” and “backup_disk_2”. The following commands are executed on the current directory.)

1. `ln c c_h`

- 현재 디렉토리의 파일 `c` 의 하드링크 `c_h` 생성. 잘 작동함.

2. `ln -s c c_s`

-현재 디렉토리의 파일 `c` 에 심볼릭링크 `c_s` 생성. 잘 작동함.

3. `ln backup_disk/a a_h`

- 디렉토리 `backup_disk` 의 파일 `a` 에 대한 하드링크 `a_h` 를 현재 디렉토리에 생성. 잘 작동함.

4. `ln -s backup_disk/a a_s`

-디렉토리 `backup_disk` 의 파일 `a` 에 대한 심볼릭링크 `a_s` 를 현재 디렉토리에 생성. 잘 작동함.

5. `ln backup_disk/a backup_disk/a_h`

- 디렉토리 `backup_disk` 의 파일 `a` 에 대한 하드링크 `a_h` 를 `backup_disk` 디렉토리 안에 생성. 잘 작동함.

6. `ln -s backup_disk/a backup_disk/a_s`

- 디렉토리 `backup_disk` 의 파일 `a` 에 대한 심볼릭 링크 `a_s` 를 `backup_disk` 디렉토리 안에 생성. 하지만 작동하지 않는다.

7. `ln backup_disk_2/b backup_disk/b_h`

- 디렉토리 `backup_disk_2` 의 파일 `b` 에 대한 하드링크를 `backup_disk` 디렉토리 안에 생성. 잘 작동함.

8. `ln backup_disk_2/b backup_disk_2/b_h`

- 디렉토리 `backup_disk_2` 의 파일 `b` 에 대한 하드링크를 `backup_disk_2` 디렉토리 안에 생성. 잘 작동함.

9. `ln -s backup_disk_2/b backup_disk/b_s`

- 디렉토리 backup_disk_2 의 파일 b 에 대한 심볼릭링크 b_s 를 backup_disk 디렉토리 안에 생성. 하지만 작동하지 않는다.

5.

a) State a few aspects that differ soft links from hard links.

symbolic link 파일은 원본의 이름을 가리키는 반면 Hard Link 파일은 원본의 inode 를 가리킨다
고로 symbolic은 inode를 하나 더 가질 뿐 아니라, 원본의 이름이 바뀌면 좌표를 잃어버린다.

b) How hard link is different from normal copied file? Answer in one sentence.

hard link는 원본의 inode를 복사한 것이다.