File System of Linux

Chonnam National University

School of Electronics and Computer Engineering

Kyungbaek Kim

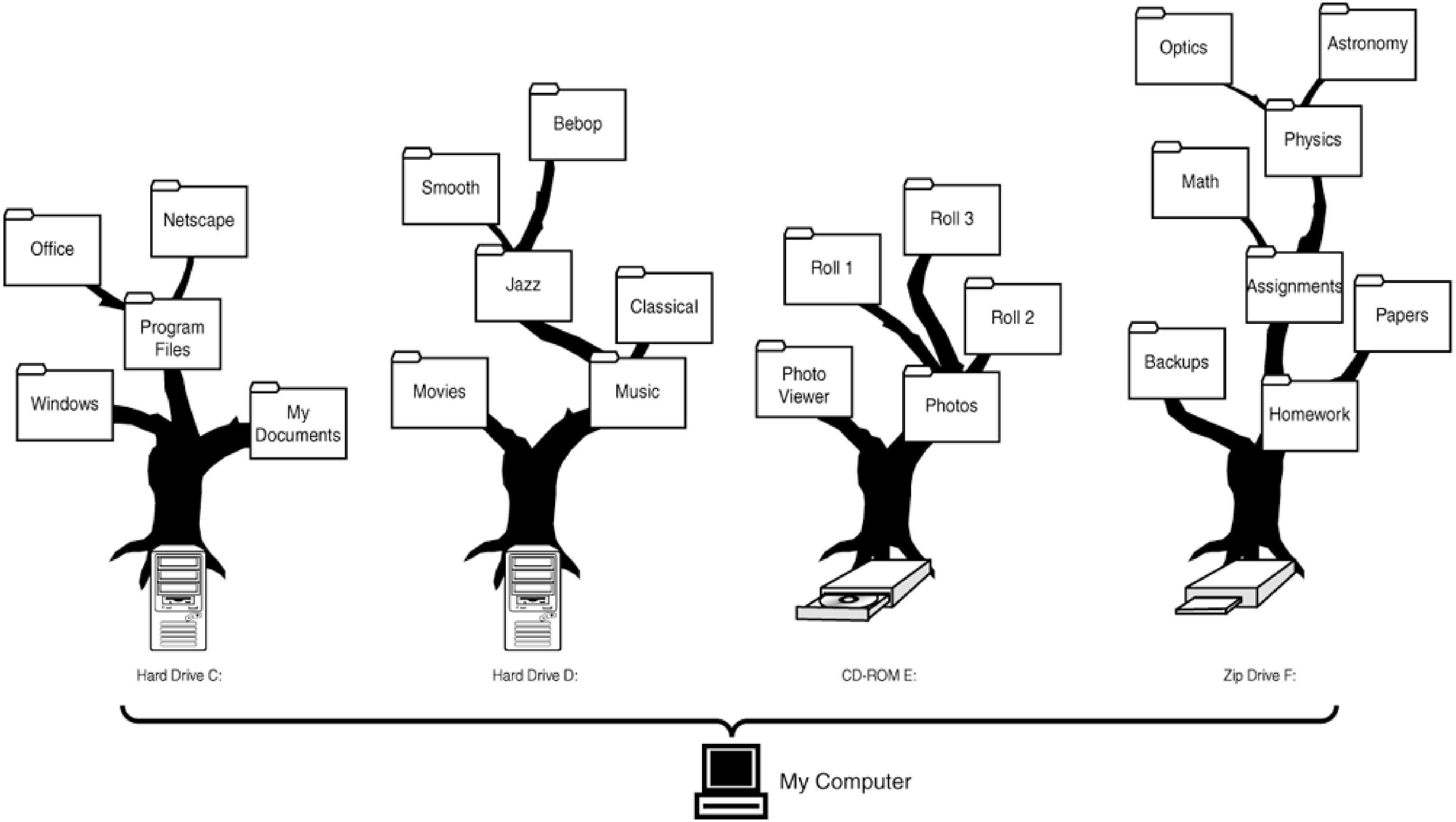
Filesystems

- Commonly used to refer to two distinct concepts
  - Unified filesystem
    - The hierarchy of directories and files which humans use to organize data on a system
  - Filesystem types
    - The formatting system which the kernel uses to store blocks of data on physical media such as disks

Unified Filesystem

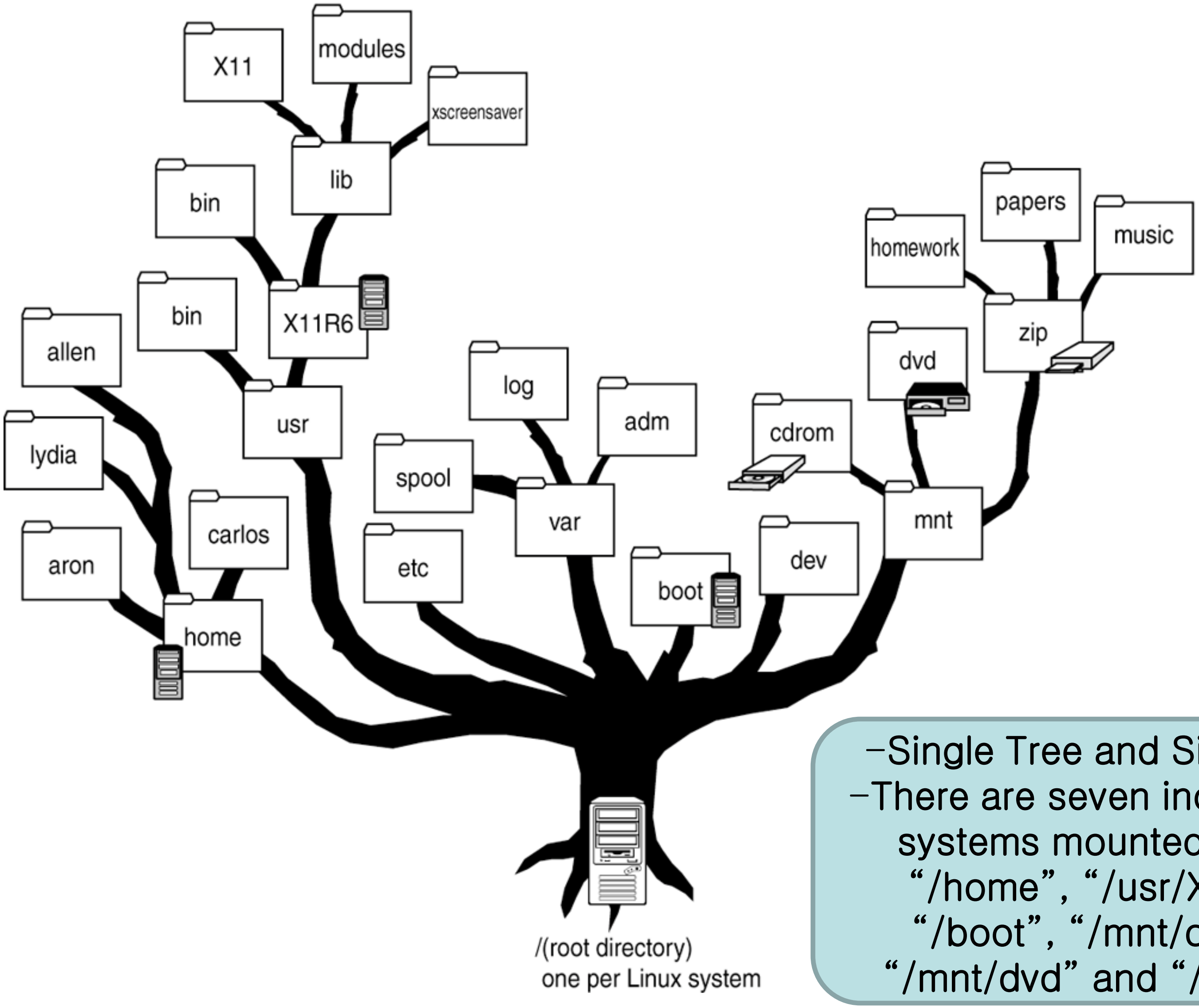
- Linux systems have a unified filesystem
  - Any file, on any disk or network share, can be accessed through a name beginning with “/”
    - The root
- Unified filesystem is made up of one or more individual filesystems
  - Each individual file system has its own root
  - That root can be grafted onto any directory in the unified filesystem
  - The directory where an individual filesystem is grafted into the unified filesystem is the individual filesystem’s mount point
- An individual file system lives on a physical device(such as a disk drive), through not necessarily on the same computer

MS Windows style File System



–Each device has its file system
–Many Trees and Many roots

Linux Unified File System

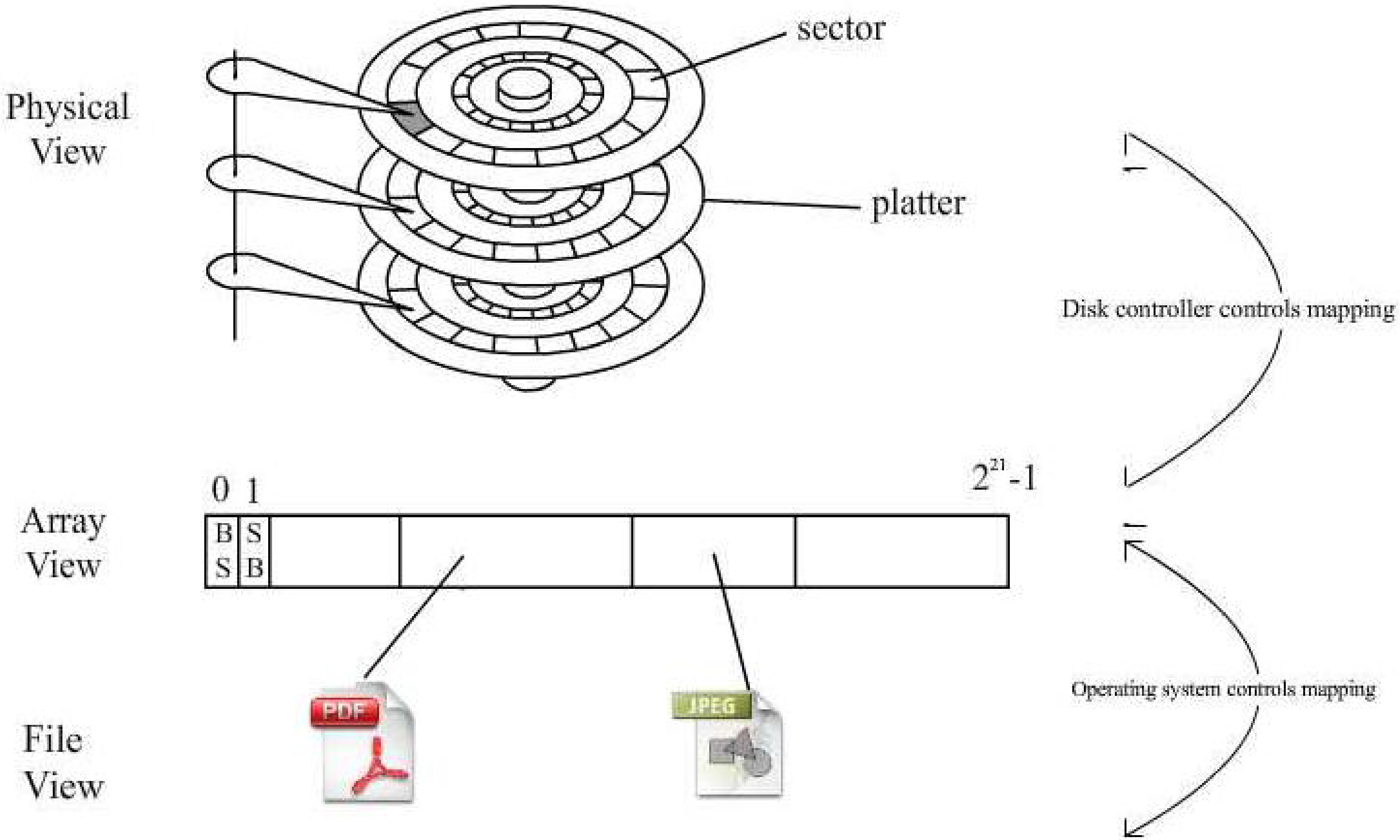


-Single Tree and Single Root
-There are seven individual file systems mounted on "/", "/home", "/usr/X11R6", "/boot", "/mnt/cdrom", "/mnt/dvd" and "/mnt/zip"

File Types

- Files directly contain data
- Directories provide a hierarchy of files: they contain both files and other directories
- *Files* and *directories* are both file types
- Other file types exist, including *device special files*
  - Device files provide a way of asking the kernel for access to a given physical device
  - The data that the device file seems to contain is actually the raw sequence of bytes or sectors on the device itself
  - Device files are, by convention, stored under the /dev directory

How to store data into your machine?



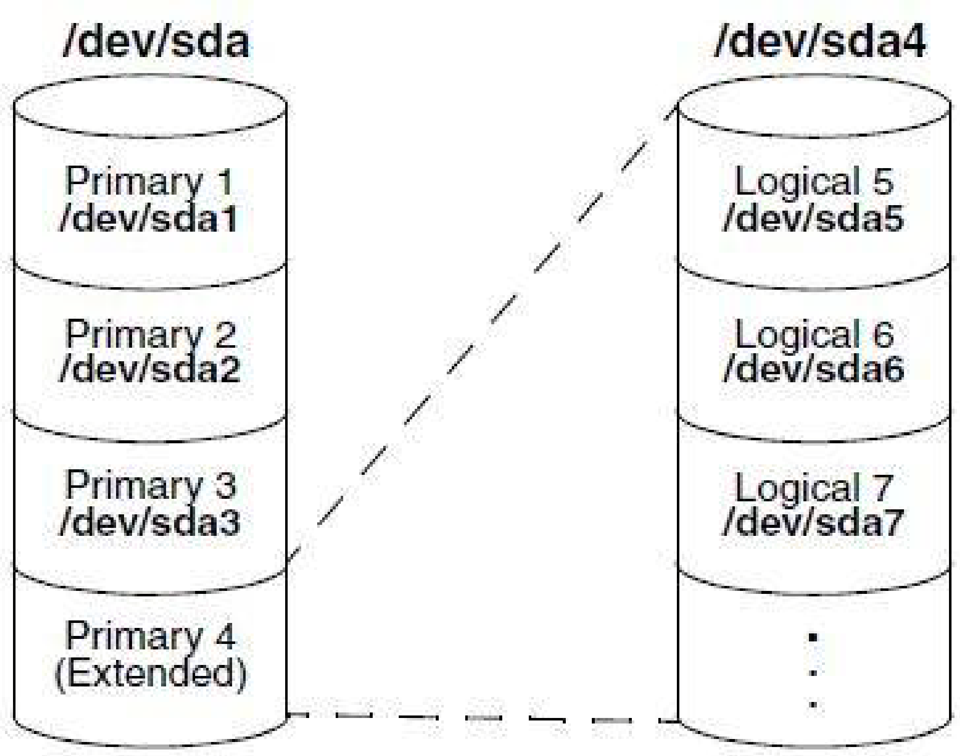
Disk Naming

- A hard disk provides a single large storage space
- The device files for IDE hard drives are **/dev/hda, /dev/hdb, /dev/hdc and /dev/hdd**
  - hda and hdb are the drives on the first IDE channel, hdc and hdd the ones on the second channel
  - The first drive on each channel is the IDE master, and the second is the IDE slave
- SCSI (Small Computer System Interface) disks are named **/dev/sda, /dev/sdb**, etc

Disks and Partitions

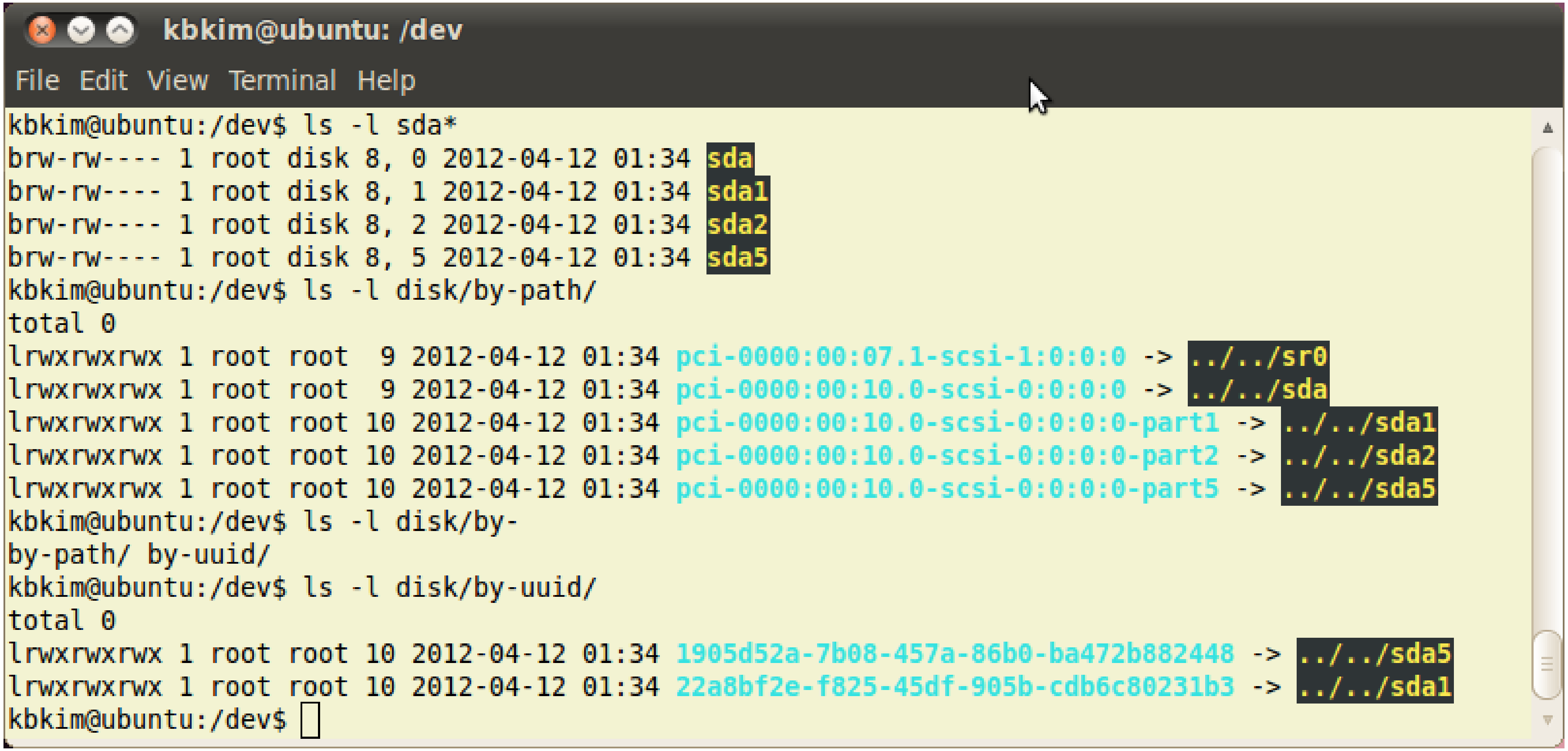
- Usually split into partitions
  - Information about partitions is stored in the partition table
  - Linux defaults using partition tables compatible with Microsoft Windows
  - For compatibility with Windows, at most four primary partitions can be made
    - *Primary partition ➔ only one filesystem*
    - Primary partitions are numbered from 1-4
  - But they can be extended partitions, which can themselves be split into smaller logical partitions
    - Extended partitions have their own partition table to store information about logical partitions
    - Logical partitions are numbered from 5

Primary vs Extended Partition



- The devices /dev/hda, etc., refer to whole hard disks, not partitions
  - Add the partition number to refer to a specific partition
  - /dev/hda1 is the first partition on the first IDE disk

Example of SCSI disks

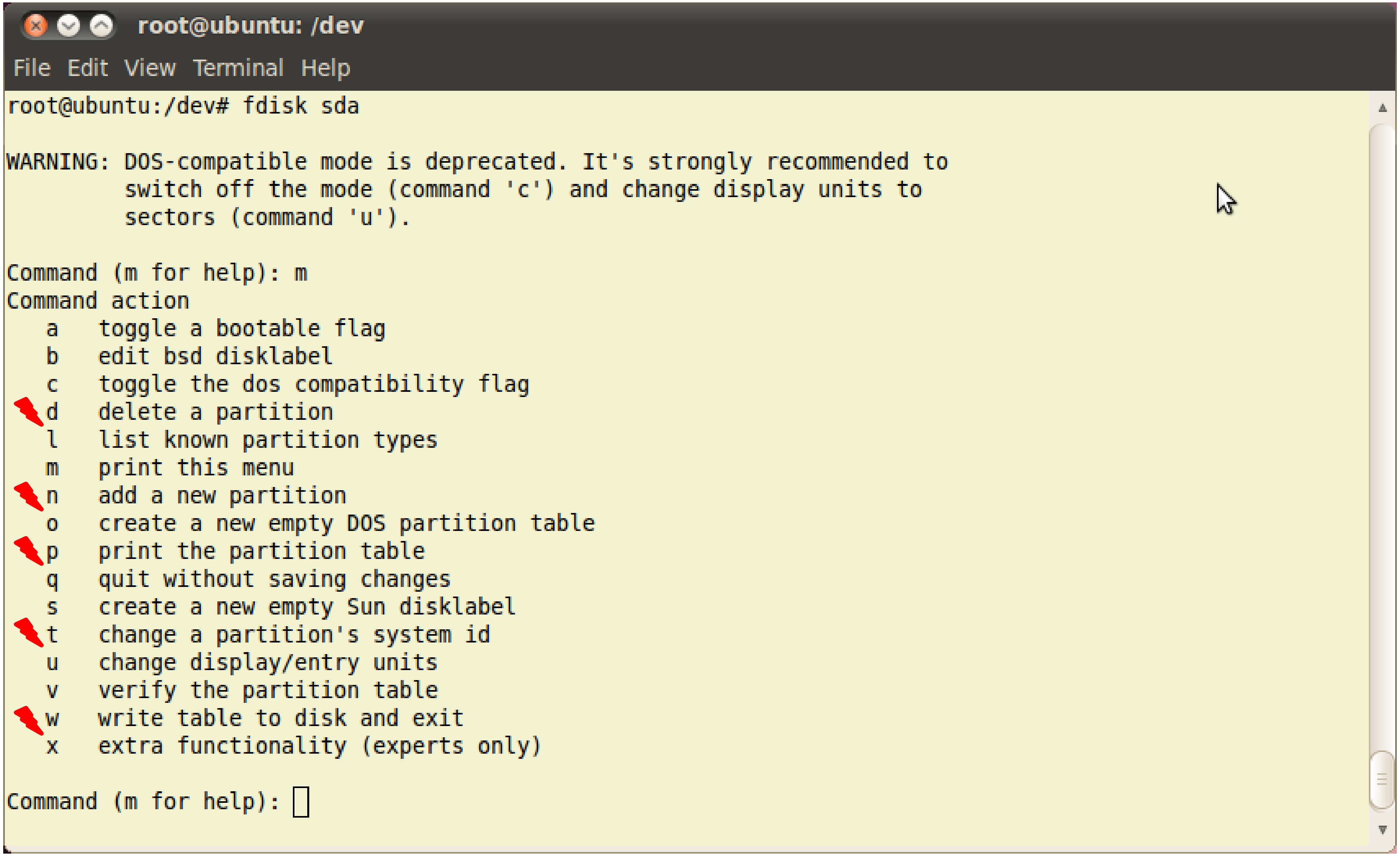


```
kbkim@ubuntu:/dev$ ls -l sda*
brw-rw---- 1 root disk 8, 0 2012-04-12 01:34 sda
brw-rw---- 1 root disk 8, 1 2012-04-12 01:34 sda1
brw-rw---- 1 root disk 8, 2 2012-04-12 01:34 sda2
brw-rw---- 1 root disk 8, 5 2012-04-12 01:34 sda5
kbkim@ubuntu:/dev$ ls -l disk/by-path/
total 0
lrwxrwxrwx 1 root root  9 2012-04-12 01:34 pci-0000:00:07.1-scsi-1:0:0:0 -> ../../sr0
lrwxrwxrwx 1 root root  9 2012-04-12 01:34 pci-0000:00:10.0-scsi-0:0:0:0 -> ../../sda
lrwxrwxrwx 1 root root 10 2012-04-12 01:34 pci-0000:00:10.0-scsi-0:0:0:0-part1 -> ../../sda1
lrwxrwxrwx 1 root root 10 2012-04-12 01:34 pci-0000:00:10.0-scsi-0:0:0:0-part2 -> ../../sda2
lrwxrwxrwx 1 root root 10 2012-04-12 01:34 pci-0000:00:10.0-scsi-0:0:0:0-part5 -> ../../sda5
kbkim@ubuntu:/dev$ ls -l disk/by-
by-path/ by-uuid/
kbkim@ubuntu:/dev$ ls -l disk/by-uuid/
total 0
lrwxrwxrwx 1 root root 10 2012-04-12 01:34 1905d52a-7b08-457a-86b0-ba472b882448 -> ../../sda5
lrwxrwxrwx 1 root root 10 2012-04-12 01:34 22a8bf2e-f825-45df-905b-cdb6c80231b3 -> ../../sda1
kbkim@ubuntu:/dev$ 
```

fdisk command

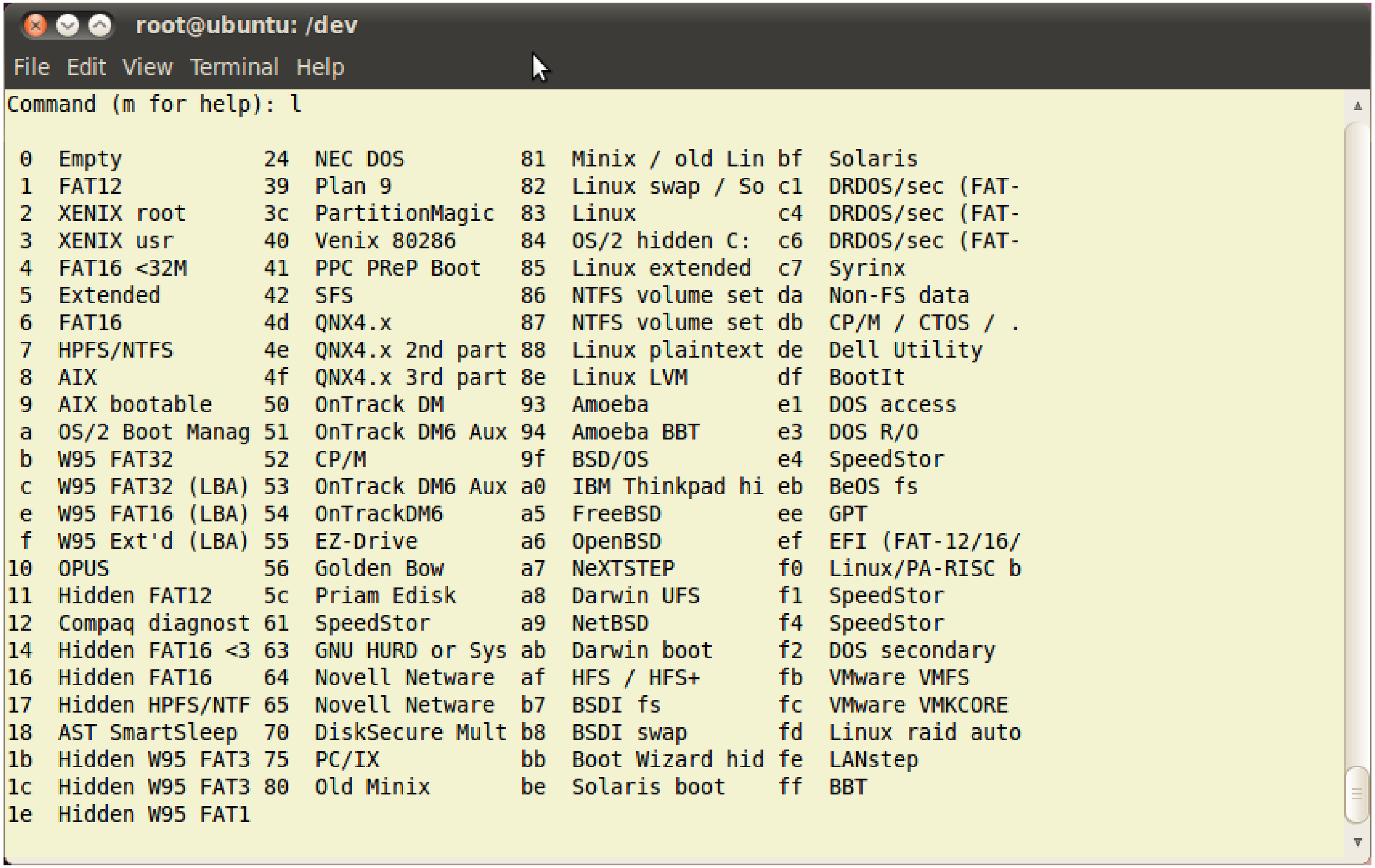
- “fdisk” is used to create, delete and change the partitions on a disk
- “fdisk” reads one-letter commands from the user
  - Type “m” to get a list of commands
  - Use “p” to show what partitions currently exist
  - Use “q” to quit without alerting anything
  - Use “w” to quit and write the changes

Command option of fdisk



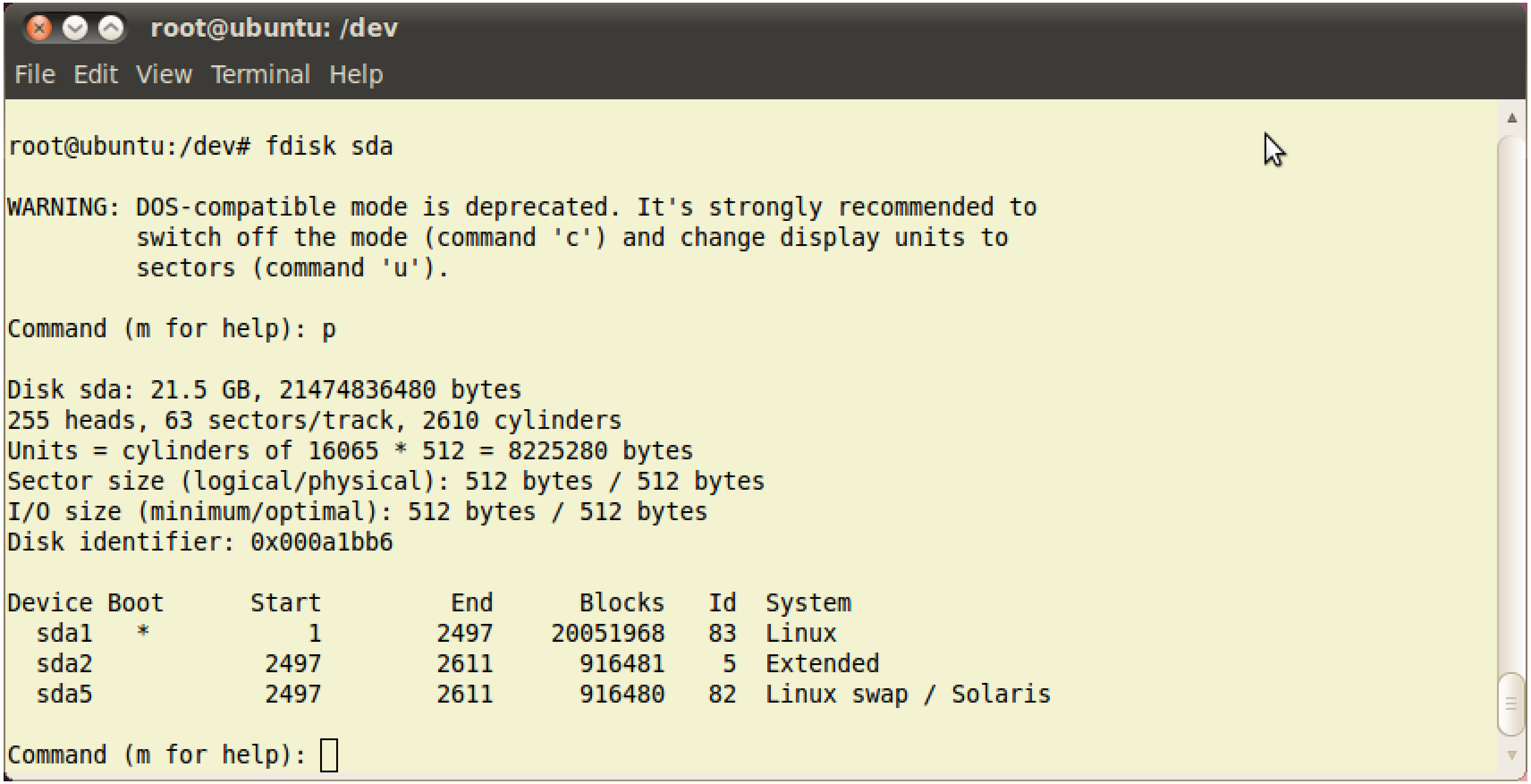
```
root@ubuntu:/dev# fdisk sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)

Command (m for help): 
```

Known Filesystem partition types



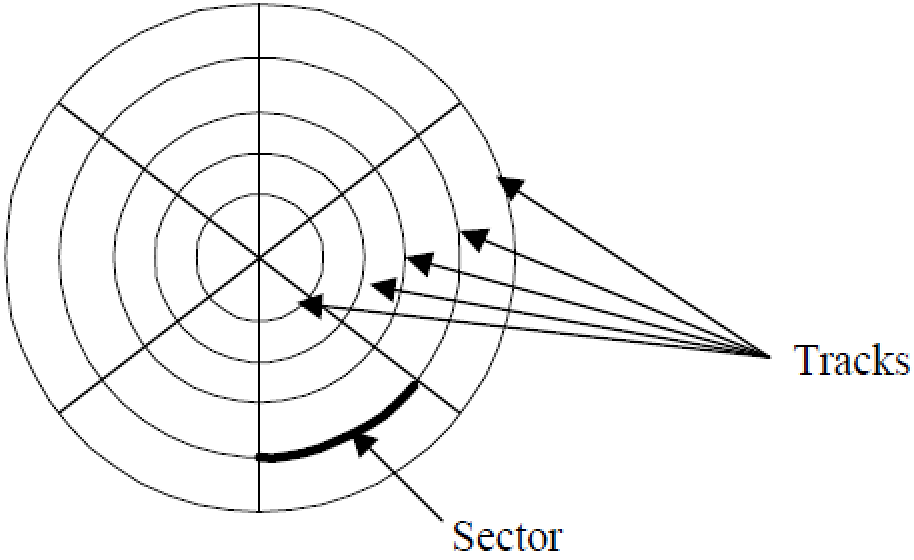
```
Command (m for help): l

 0  Empty           24  NEC DOS         81  Minix / old Lin bf  Solaris
 1  FAT12           39  Plan 9          82  Linux swap / So c1  DRDOS/sec (FAT-
 2  XENIX root      3c  PartitionMagic  83  Linux           c4  DRDOS/sec (FAT-
 3  XENIX usr       40  Venix 80286     84  OS/2 hidden C:  c6  DRDOS/sec (FAT-
 4  FAT16 <32M      41  PPC PReP Boot   85  Linux extended  c7  Syrinx
 5  Extended        42  SFS             86  NTFS volume set da  Non-FS data
 6  FAT16           4d  QNX4.x          87  NTFS volume set db  CP/M / CTOS / .
 7  HPFS/NTFS       4e  QNX4.x 2nd part 88  Linux plaintext de  Dell Utility
 8  AIX             4f  QNX4.x 3rd part 8e  Linux LVM       df  BootIt
 9  AIX bootable    50  OnTrack DM      93  Amoeba          e1  DOS access
 a  OS/2 Boot Manag 51  OnTrack DM6 Aux 94  Amoeba BBT      e3  DOS R/O
 b  W95 FAT32       52  CP/M            9f  BSD/OS          e4  SpeedStor
 c  W95 FAT32 (LBA) 53  OnTrack DM6 Aux a0  IBM Thinkpad hi eb  BeOS fs
 e  W95 FAT16 (LBA) 54  OnTrackDM6      a5  FreeBSD         ee  GPT
 f  W95 Ext'd (LBA) 55  EZ-Drive        a6  OpenBSD         ef  EFI (FAT-12/16/
10  OPUS            56  Golden Bow      a7  NeXTSTEP        f0  Linux/PA-RISC b
11  Hidden FAT12    5c  Priam Edisk     a8  Darwin UFS      f1  SpeedStor
12  Compaq diagnost 61  SpeedStor       a9  NetBSD          f4  SpeedStor
14  Hidden FAT16 <3 63  GNU HURD or Sys ab  Darwin boot     f2  DOS secondary
16  Hidden FAT16    64  Novell Netware  af  HFS / HFS+      fb  VMware VMFS
17  Hidden HPFS/NTF 65  Novell Netware  b7  BSDI fs         fc  VMware VMKCORE
18  AST SmartSleep  70  DiskSecure Mult b8  BSDI swap       fd  Linux raid auto
1b  Hidden W95 FAT3 75  PC/IX           bb  Boot Wizard hid fe  LANstep
1c  Hidden W95 FAT3 80  Old Minix       be  Solaris boot    ff  BBT
1e  Hidden W95 FAT1
```

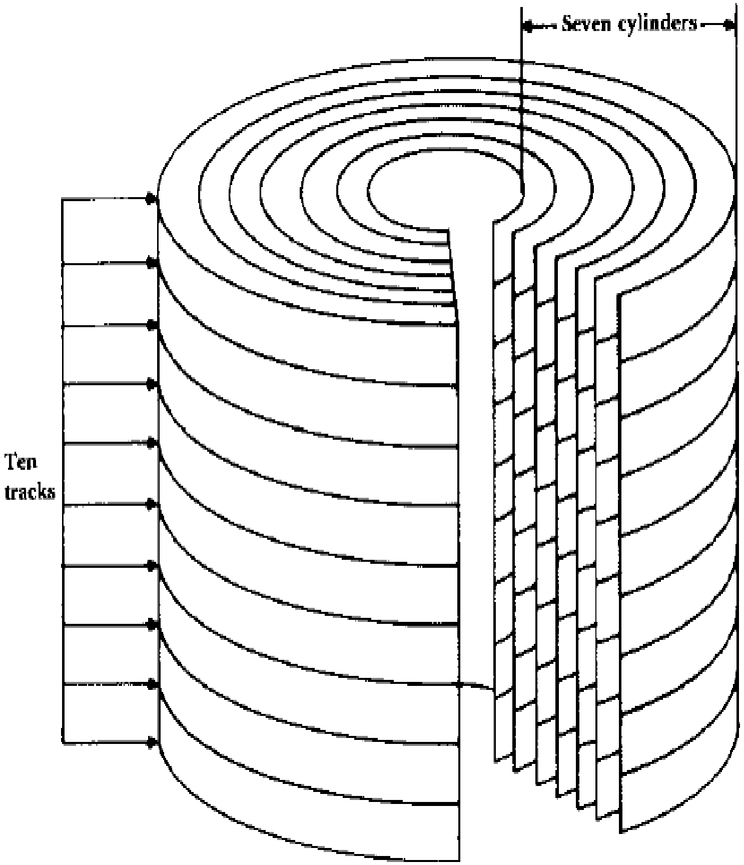
Partition Table

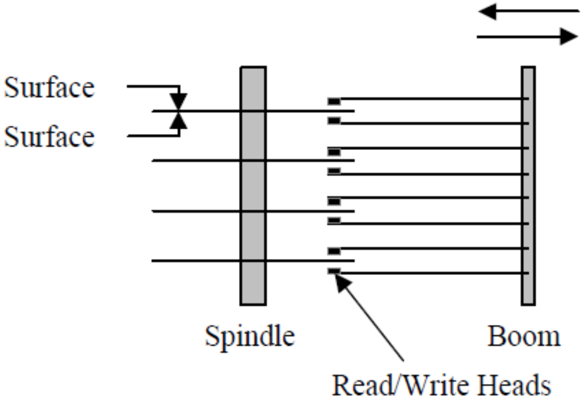


```
root@ubuntu:/dev# fdisk sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): p

Disk sda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000a1bb6

Device Boot      Start         End      Blocks   Id  System
   sda1   *           1        2497    20051968   83  Linux
   sda2            2497        2611      916481    5  Extended
   sda5            2497        2611      916480   82  Linux swap / Solaris

Command (m for help): 
```

Sector/Track(head)/Cylinder







Sectors/Track * Head(Track)/Cylinder = Sector/Cylinder
→ 63 * 255 = 16065
Bytes/Sector * Sector/Cylinder * Cylinder/Disk = Byte/Disk
→ 512 * 16065 * 2610 = 21.5GB

mkfs command

- “mkfs” initializes a filesystem on a new partition
  - Warning!! : any old data on the partition will be lost
  - e.g.) $ mkfs –t ext4 –c /dev/sda2
    - Make an ext4 filesystem on /dev/sda2

Types of filesystems

- FAT(File Allocation Table) : FAT12/16/32
  ➔ DOS, Windows 9x
- NTFS (Net Technology File System)
  ➔ Windows NT, windows 2000, windows XP/7
- HFS (Hierarchical File System), HFS+ ➔ Mac
- HPFS (High Performance File system) ➔ OS/2 operation system
- UFS (Unix File System) ➔ BSD unix file system
- ext2, ext3, ext4 (extended filesystem)
  ➔ Linux, Android 2.3
- VMFS (Virtual Machine File System) ➔ VMWare
- UDF (Universal Disk Format) ➔ Optical disks

Mounting Filesystems

- A partition contains entirely arbitrary data
- One filesystem is made the root filesystem
  - the root directory on that filesystem becomes the directory named “/”
- Other filesystems can be mounted
  - The root directory of that filesystem is grafted onto a directory of the root filesystems
  - This arranges for every file in every mounted filesystem to be accessible from a single unified name space
- The directory grafted onto is called the mount point

mount

- Important filesystems are mounted at boot-up
  - Other filesystems can be mounted or unmounted at any time
- “mount” mounts a filesystem
  - You usually need to have root permission to mount a filesystem
- “mount” makes it easy to mount filesystems configured by the system administrator

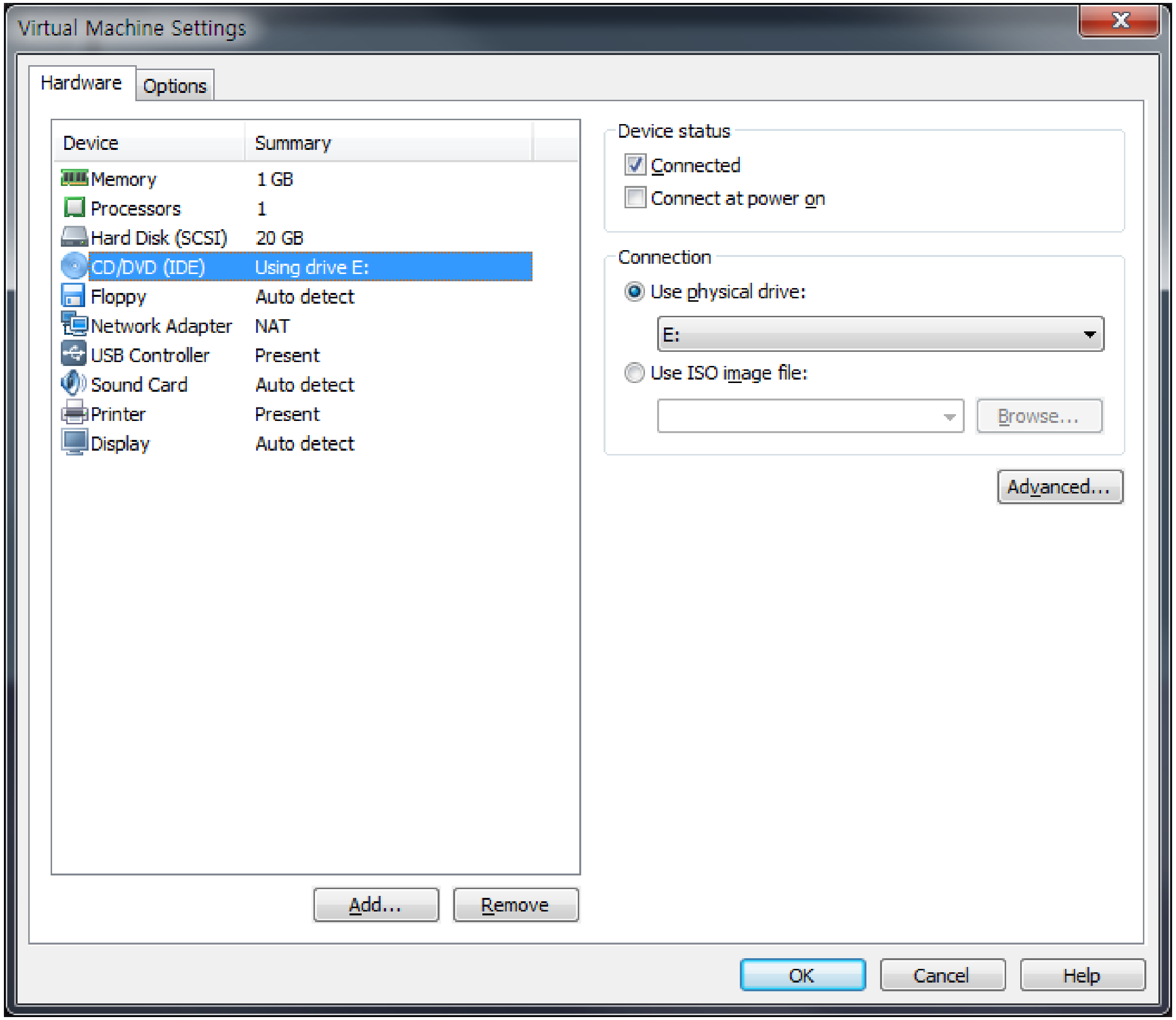
Mounting other filesystems

- “mount /dev/sdb3 /mnt/extra”
  - mounts the filesystem stored in the “/dev/sdb3” device on the mount point “/mnt/extra”
- You may occasionally need to specify the filesystem type explicitly
  - e.g.) $ mount –t vfat /dev/hdd1 /mnt/windows

umount

- “umount” unmounts a filesystem
  - Note the spelling.
- “umount /mnt/extra”
  - Unmounts whatever is on the /mnt/extra mount point
- “umount /dev/sdb3”
  - Unmounts any filesystem in the /dev/sdb3 device
- Need root permission
- It is also impossible to unmount a “busy” filesystem
  - If a process has file on it open
  - If a process has a directory with it as its current directory

Example of mount/umount : setting for VMWare



Device	Summary
Memory	1 GB
Processors	1
Hard Disk (SCSI)	20 GB
CD/DVD (IDE)	Using drive E:
Floppy	Auto detect
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Device status

☑ Connected

□ Connect at power on

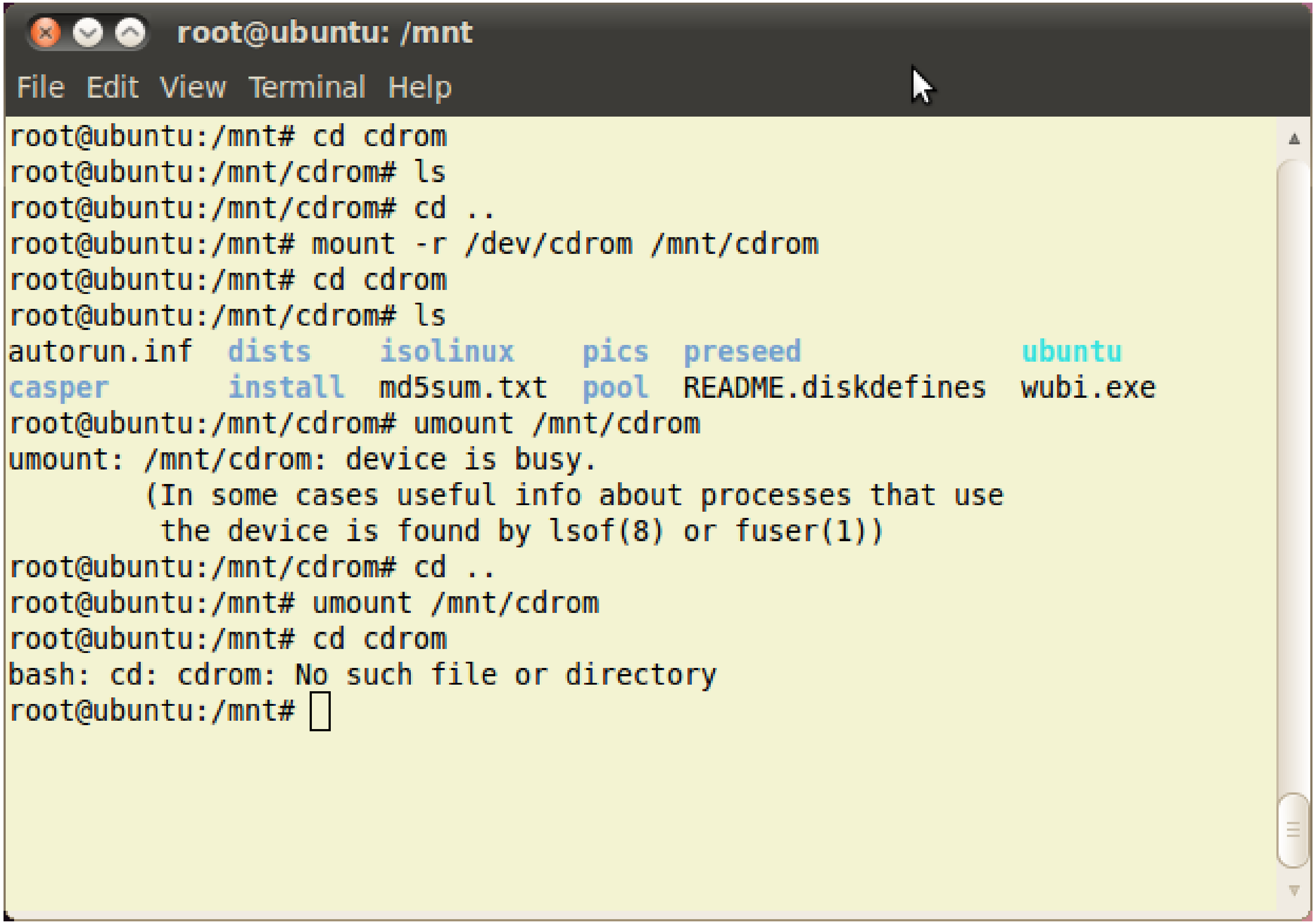
Connection

● Use physical drive:

E:

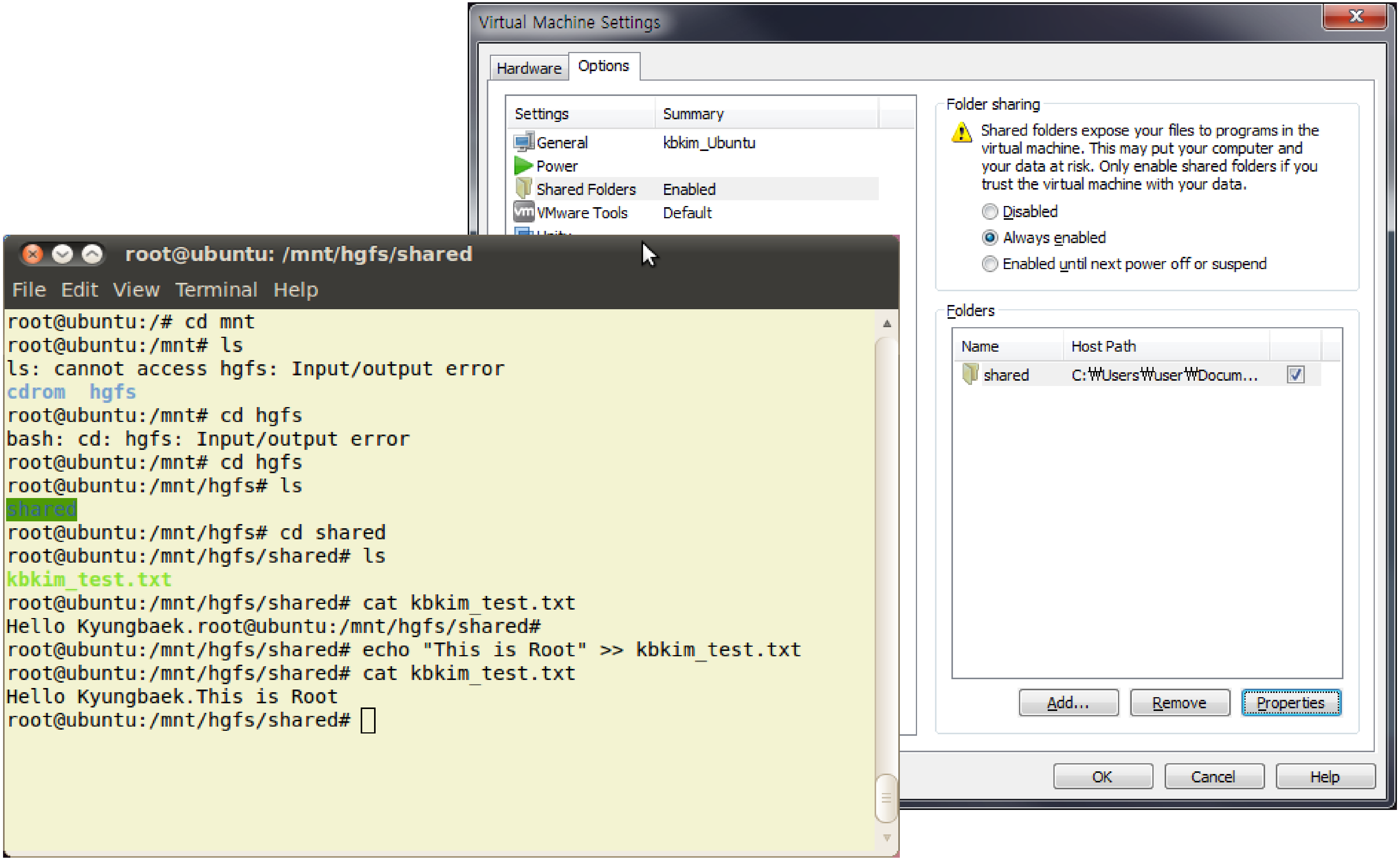
○ Use ISO image file:

Example of mount/umount



```
root@ubuntu:/mnt# cd cdrom
root@ubuntu:/mnt/cdrom# ls
root@ubuntu:/mnt/cdrom# cd ..
root@ubuntu:/mnt# mount -r /dev/cdrom /mnt/cdrom
root@ubuntu:/mnt# cd cdrom
root@ubuntu:/mnt/cdrom# ls
autorun.inf  dists    isolinux    pics  preseed             ubuntu
casper       install  md5sum.txt  pool  README.diskdefines  wubi.exe
root@ubuntu:/mnt/cdrom# umount /mnt/cdrom
umount: /mnt/cdrom: device is busy.
        (In some cases useful info about processes that use
         the device is found by lsof(8) or fuser(1))
root@ubuntu:/mnt/cdrom# cd ..
root@ubuntu:/mnt# umount /mnt/cdrom
root@ubuntu:/mnt# cd cdrom
bash: cd: cdrom: No such file or directory
root@ubuntu:/mnt# 
```

HGFS (Host Guest File System)



Virtual Machine Settings

Settings	Summary
General	kbkim_Ubuntu
Power	
Shared Folders	Enabled
VMware Tools	Default
Unity	

Folder sharing

Shared folders expose your files to programs in the virtual machine. This may put your computer and your data at risk. Only enable shared folders if you trust the virtual machine with your data.

○ Disabled
● Always enabled
○ Enabled until next power off or suspend

Folders

Name	Host Path	
shared	C:₩Users₩user₩Docum...	☑

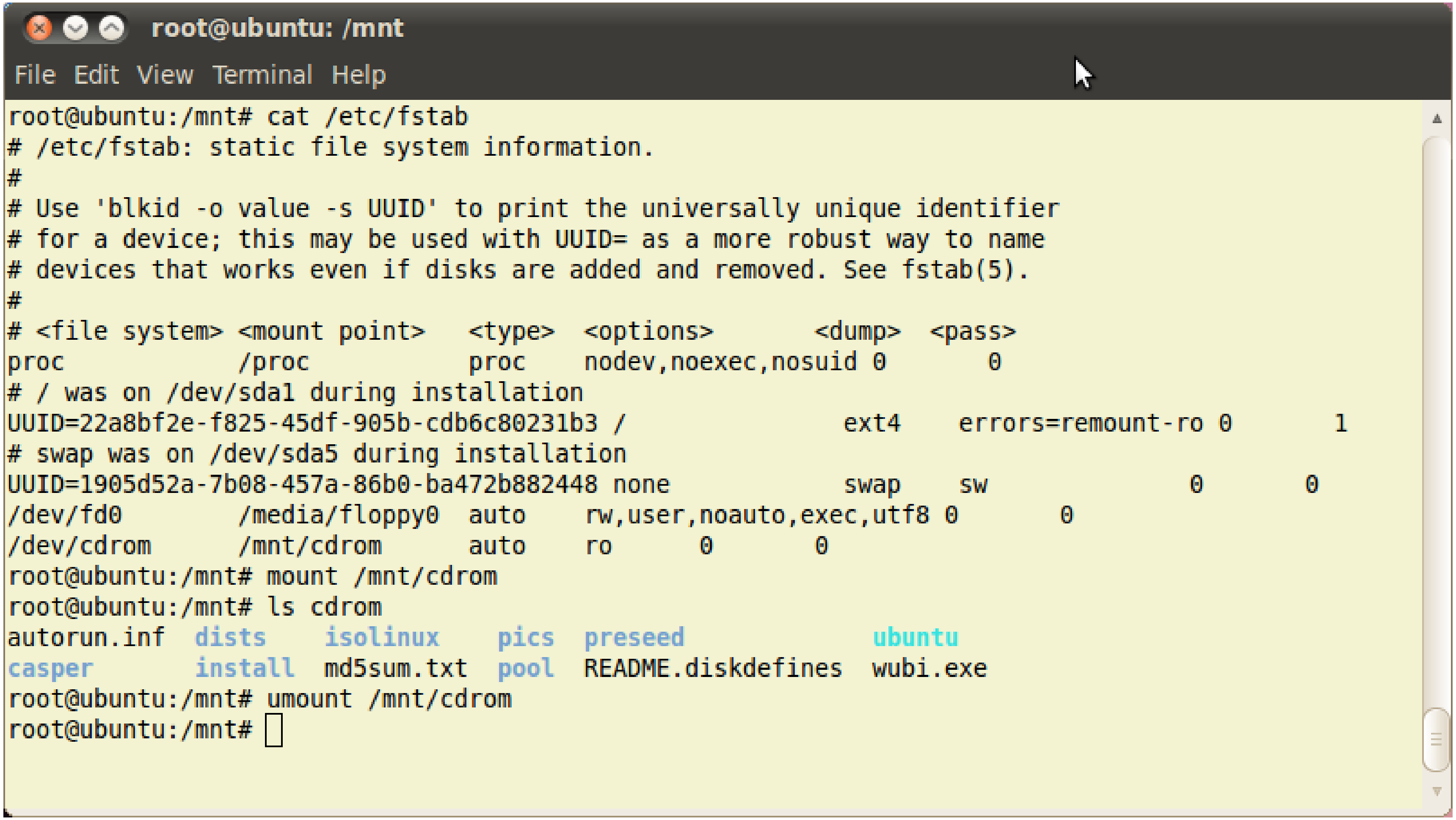
root@ubuntu: /mnt/hgfs/shared

```
root@ubuntu:/# cd mnt
root@ubuntu:/mnt# ls
ls: cannot access hgfs: Input/output error
cdrom  hgfs
root@ubuntu:/mnt# cd hgfs
bash: cd: hgfs: Input/output error
root@ubuntu:/mnt# cd hgfs
root@ubuntu:/mnt/hgfs# ls
shared
root@ubuntu:/mnt/hgfs# cd shared
root@ubuntu:/mnt/hgfs/shared# ls
kbkim_test.txt
root@ubuntu:/mnt/hgfs/shared# cat kbkim_test.txt
Hello Kyungbaek.root@ubuntu:/mnt/hgfs/shared#
root@ubuntu:/mnt/hgfs/shared# echo "This is Root" >> kbkim_test.txt
root@ubuntu:/mnt/hgfs/shared# cat kbkim_test.txt
Hello Kyungbaek.This is Root
root@ubuntu:/mnt/hgfs/shared# 
```

Configuring mount : /etc/fstab

- “/etc/fstab” contains information about filesystems that are known the system administrator
  - Specifying a filesystem in /etc/fstab makes it possible to use its mount point as the only argument to mount
- Configure Boot-up mount
- Each line describes one filesystem
- Six columns on each line

Example of fstab file



```
root@ubuntu:/mnt# cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally unique identifier
# for a device; this may be used with UUID= as a more robust way to name
# devices that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    nodev,noexec,nosuid 0       0
# / was on /dev/sda1 during installation
UUID=22a8bf2e-f825-45df-905b-cdb6c80231b3 /               ext4    errors=remount-ro 0       1
# swap was on /dev/sda5 during installation
UUID=1905d52a-7b08-457a-86b0-ba472b882448 none            swap    sw              0       0
/dev/fd0        /media/floppy0  auto    rw,user,noauto,exec,utf8 0       0
/dev/cdrom      /mnt/cdrom      auto    ro      0       0
root@ubuntu:/mnt# mount /mnt/cdrom
root@ubuntu:/mnt# ls cdrom
autorun.inf  dists    isolinux    pics  preseed             ubuntu
casper       install  md5sum.txt  pool  README.diskdefines  wubi.exe
root@ubuntu:/mnt# umount /mnt/cdrom
root@ubuntu:/mnt# 
```

Mount options

- Comma-separated options in /etc/fstab
- Alternatively, use comma-separated options with –o on the mount command line
- Options
  - "auto"➔ mount a filesystem on boot-up
  - "noauto" ➔ prevent mounting a filesystem on boot-up, useful for removable media
  - "ro" ➔ read only
  - "rw" ➔ read/write
  - "users" ➔ let non-root users mount/umount

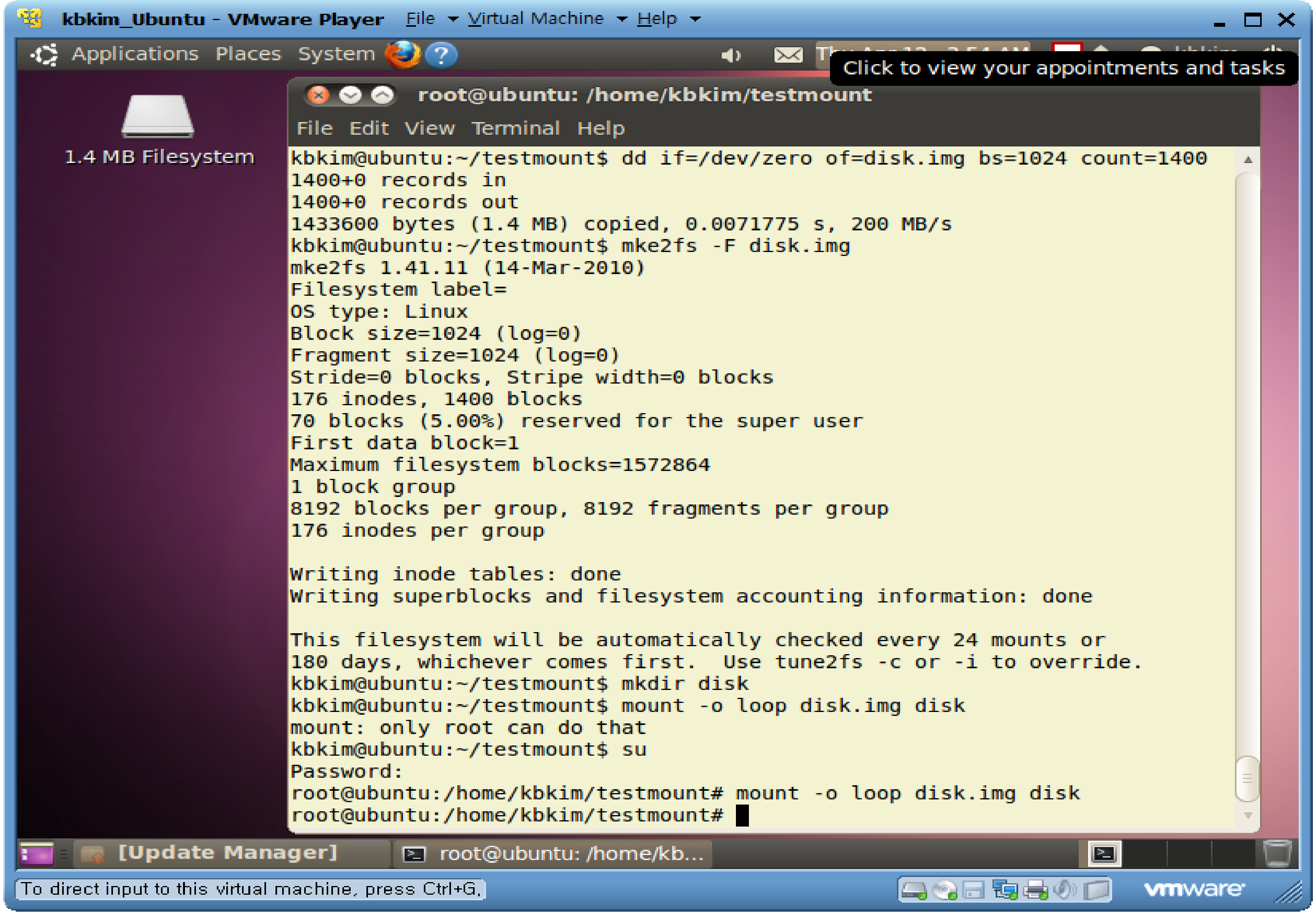
Other colums in /etc/fstab

- Dump
  - Used by the dump and restore backup utilities
  - Just use 1 for normal filesystems and for 0 for removable filesystems
- Pass-no
  - Controls the order in which automatically-mounted filesystems are checked by fsck
  - Use 1 for the root filesystem
  - Use 0 for filesystems that are not mounted at boot-up
  - Use 2 for other filesystems

Mounting a file

- Using loop devices
  - Linux can mount a filesystem stored in a normal file, instead of a disk
  - Use “-o loop” option
  - e.g.) image files
- Useful for testing images of CD-ROMs before burning them to disk

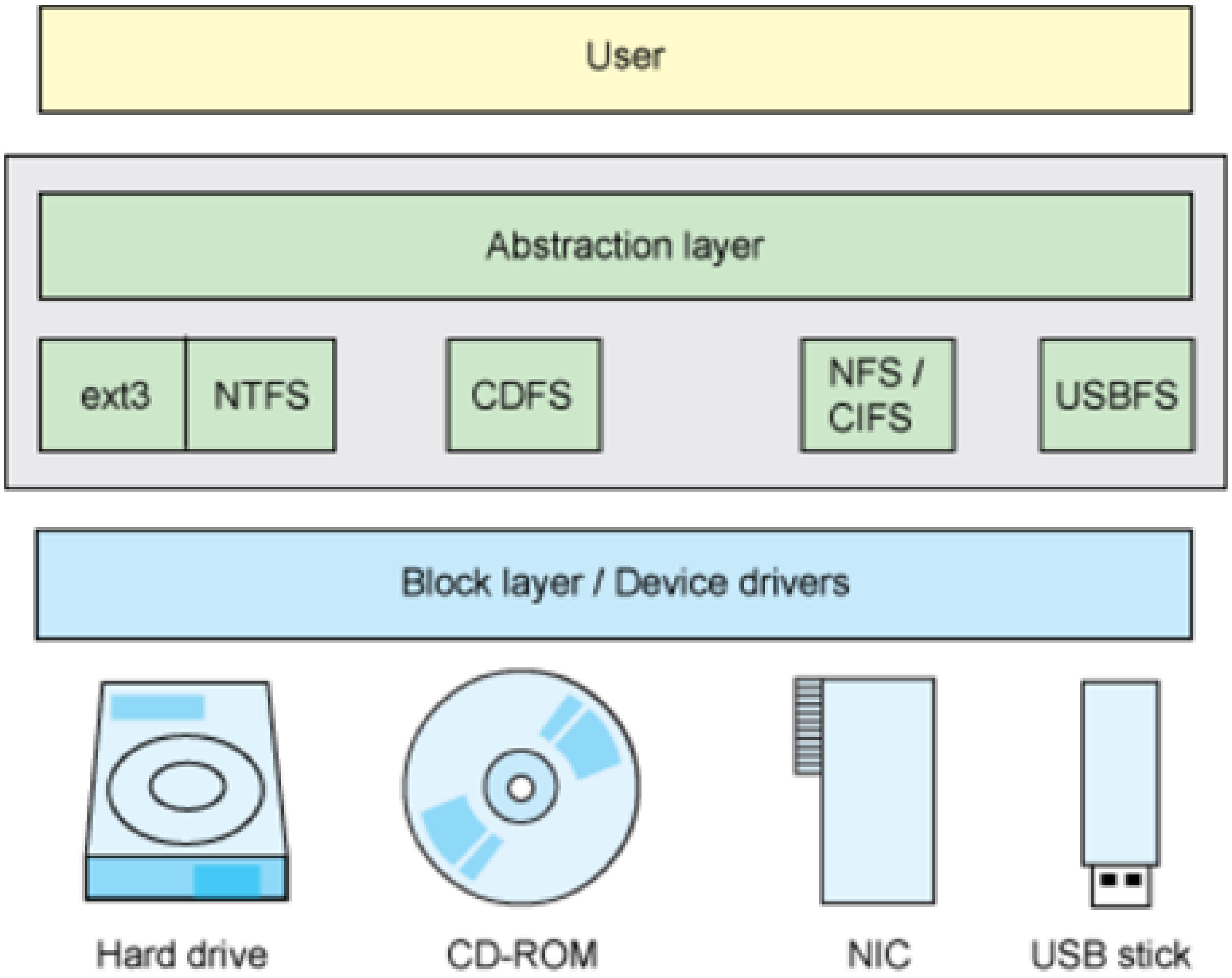
Example of mounting a file



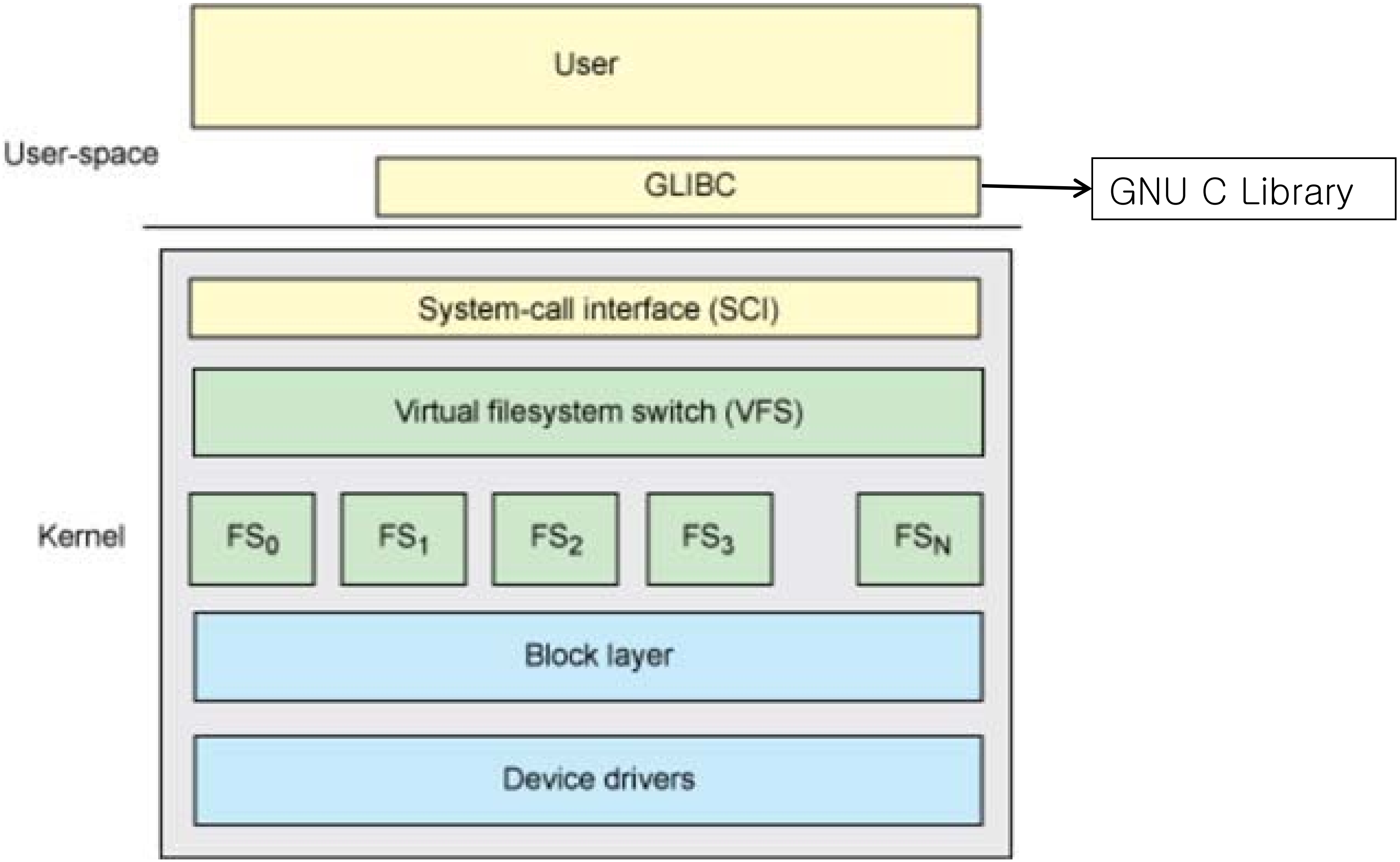
kbkim_Ubuntu - VMware Player
File
Virtual Machine
Help
Applications
Places
System
Click to view your appointments and tasks
1.4 MB Filesystem
root@ubuntu: /home/kbkim/testmount
File
Edit
View
Terminal
Help
[Update Manager]
root@ubuntu: /home/kb...
To direct input to this virtual machine, press Ctrl+G.
vmware

```
kbkim@ubuntu:~/testmount$ dd if=/dev/zero of=disk.img bs=1024 count=1400
1400+0 records in
1400+0 records out
1433600 bytes (1.4 MB) copied, 0.0071775 s, 200 MB/s
kbkim@ubuntu:~/testmount$ mke2fs -F disk.img
mke2fs 1.41.11 (14-Mar-2010)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
176 inodes, 1400 blocks
70 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=1572864
1 block group
8192 blocks per group, 8192 fragments per group
176 inodes per group

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

This filesystem will be automatically checked every 24 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
kbkim@ubuntu:~/testmount$ mkdir disk
kbkim@ubuntu:~/testmount$ mount -o loop disk.img disk
mount: only root can do that
kbkim@ubuntu:~/testmount$ su
Password:
root@ubuntu:/home/kbkim/testmount# mount -o loop disk.img disk
root@ubuntu:/home/kbkim/testmount# 
```

Abstraction of Multiple File System



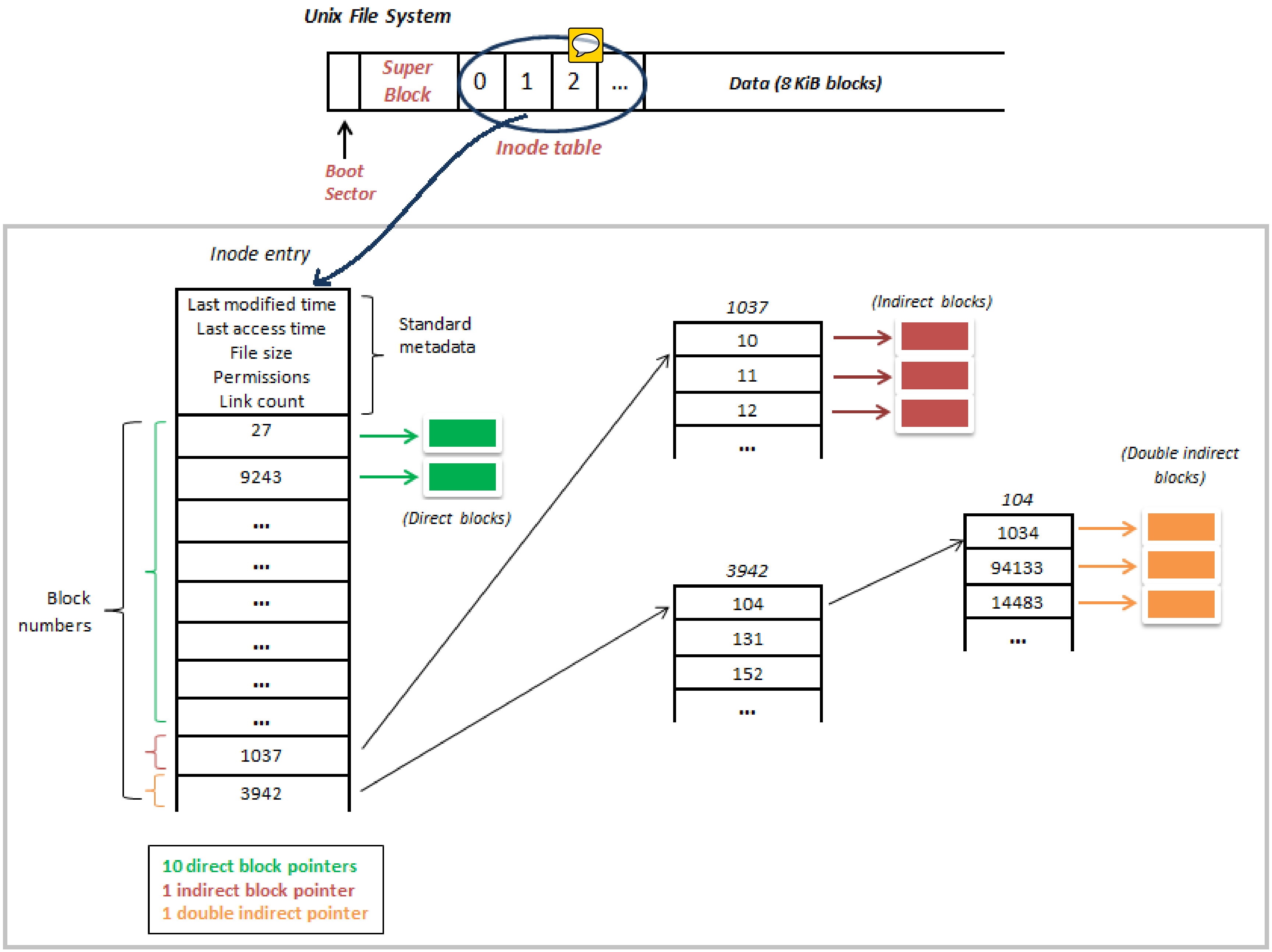
Virtual Filesystem Switch (VFS)



GNU C Library

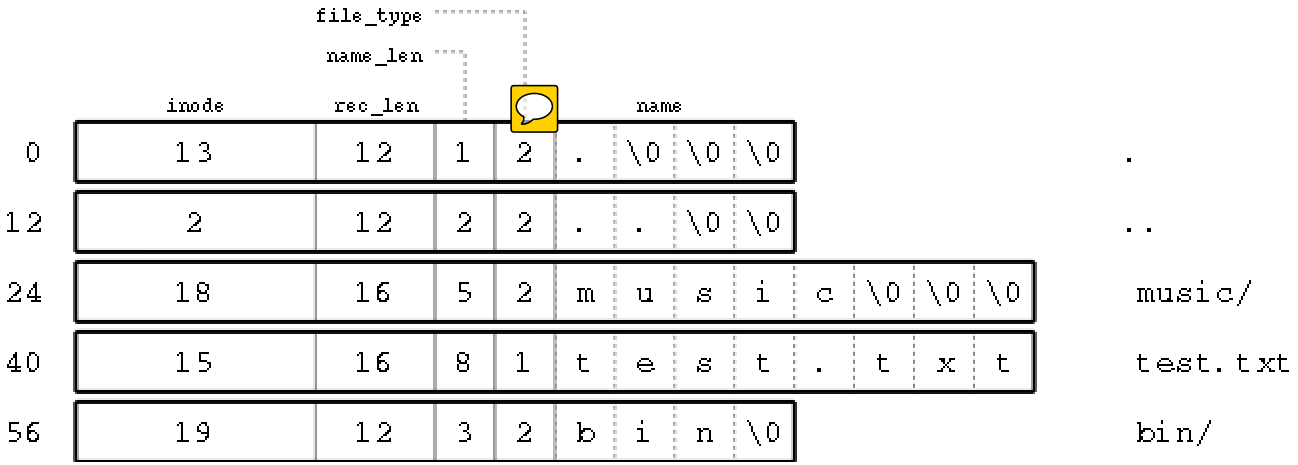
Inodes (Index nodes)

- Data structure that describes a file on an individual filesystem
- It contains information about the file, including its type (file/directory/device), size, modification, time, permissions, etc
- You can regard an inode as being the file itself
- The Inodes within an individual filesystem are numbered ➔ inum


Unix File System
Super Block
0
1
2
...
Data (8 KiB blocks)
Boot Sector
Inode table
Inode entry
Last modified time
Last access time
File size
Permissions
Link count
Standard metadata
27
9243
...
(Direct blocks)
1037
3942
Block numbers
1037
10
11
12
...
(Indirect blocks)
3942
104
131
152
...
104
1034
94133
14483
...
(Double indirect blocks)
10 direct block pointers
1 indirect block pointer
1 double indirect pointer

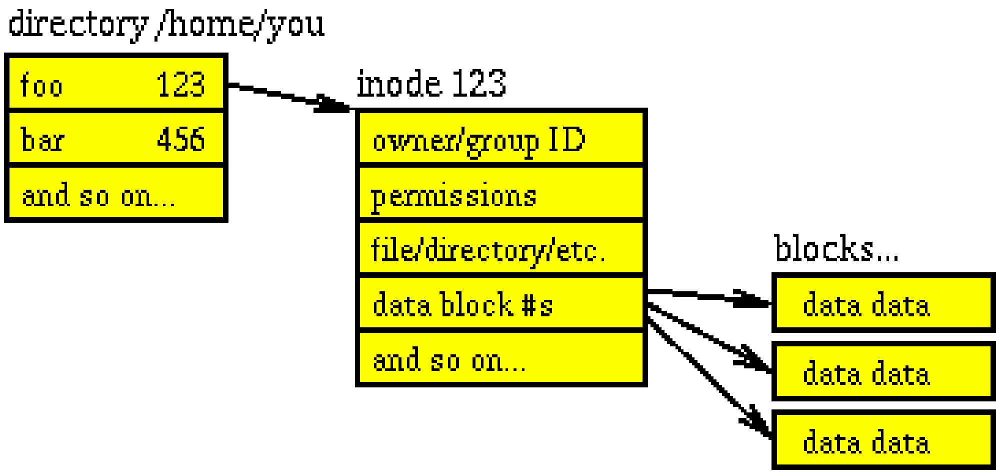
Directory

- File's name is stored not in its inode, but in directory
- Stored on disk as a list of file and directory names
- Each name has an inode number associated with it
- Separating names from Inodes
  → multiple directory entries referring to the same file



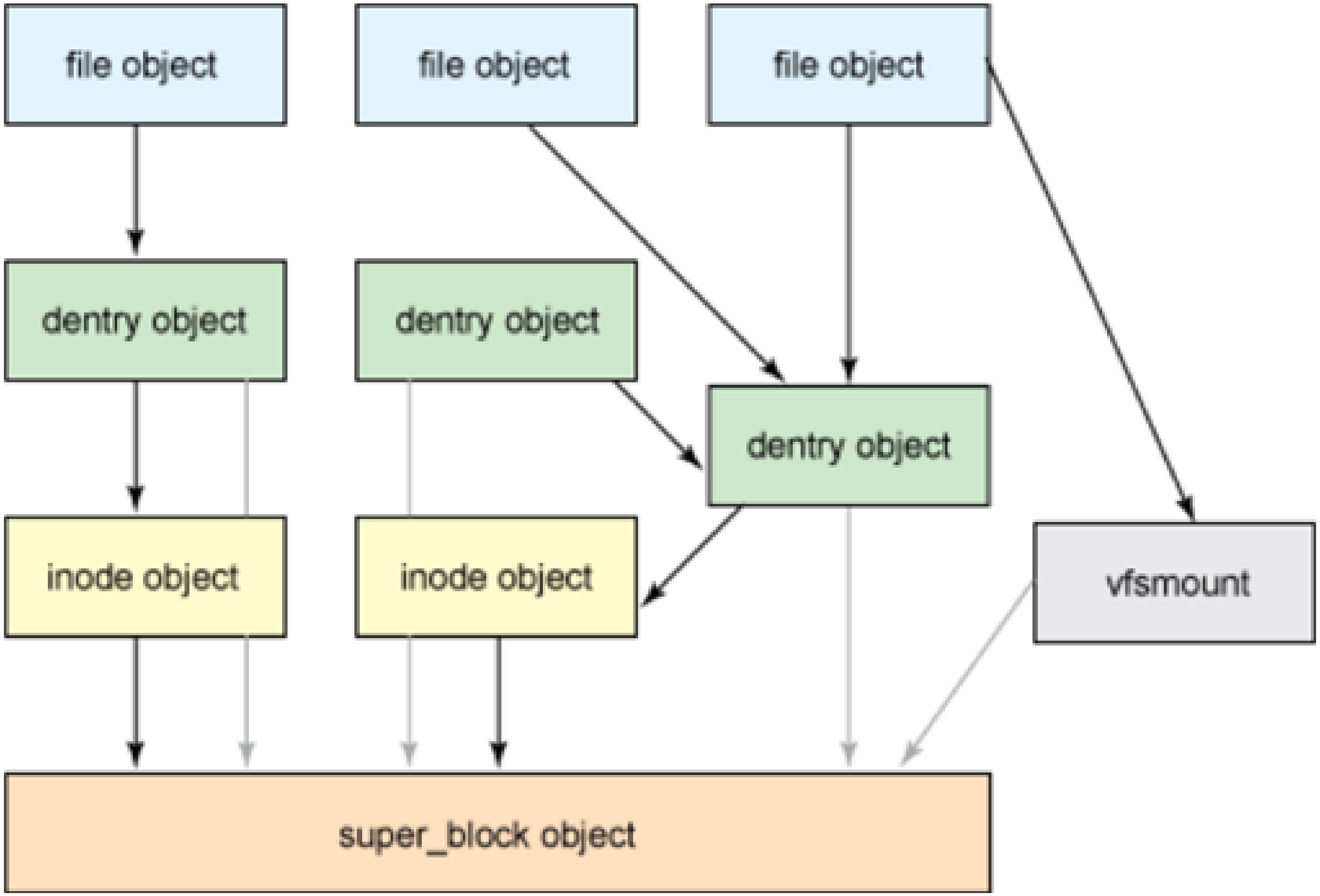
	inode	rec_len	name_len	file_type	name								
0	13	12	1	2	.	\0	\0	\0					.
12	2	12	2	2	.	.	\0	\0					..
24	18	16	5	2	m	u	s	i	c	\0	\0	\0	music/
40	15	16	8	1	t	e	s	t	.	t	x	t	test.txt
56	19	12	3	2	b	i	n	\0					bin/

The structure of a directory entry



Resolution of file path

In-memory VFS objects



struct file

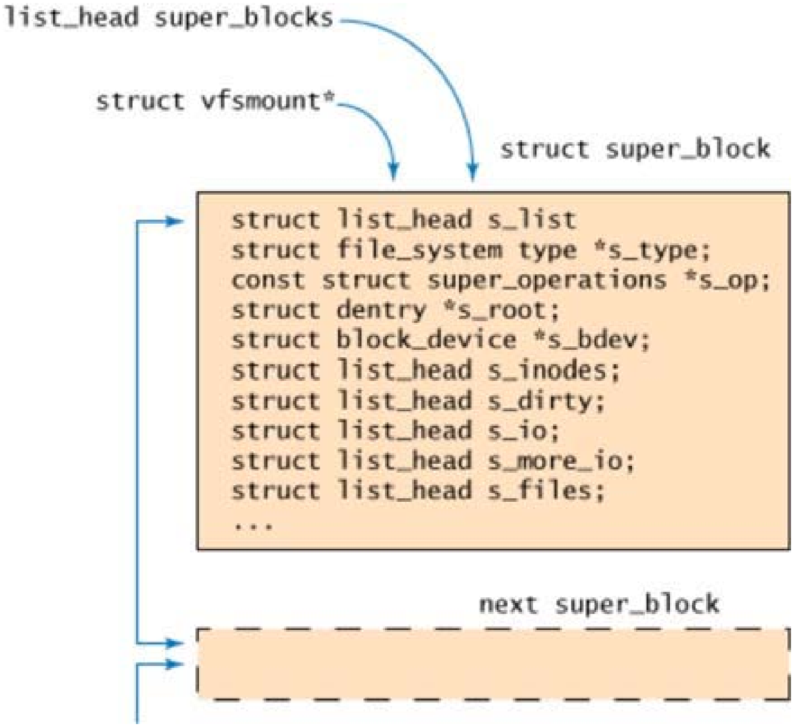
```
struct path f_path;
struct dentry (f_path.dentry);
const struct file_operations *f_op;
unsigned int f_flags;
fmode_t f_mode;
lodd_t f_pos;
...
```

struct dentry

```
struct super_block *d_sb;
struct dentry *d_parent;
struct list_head d_subdirs;
struct dentry_operations *d_op;
unsigned char d_iname[];
struct inod *d_inode;
...
```

struct inode

```
struct list_head i_dentry;
struct timespec i_atime;
struct timespec i_mtime;
struct timespec i_ctime;
gid_t i_gid;
uid_t i_uid;
loff_t i_size;
const struct file_operations *i_fop;
const struct inode_operations *i_op;
struct address_space *i_mapping;
struct address_space *i_data;
...
```



```
struct list_head s_list
struct file_system type *s_type;
const struct super_operations *s_op;
struct dentry *s_root;
struct block_device *s_bdev;
struct list_head s_inodes;
struct list_head s_dirty;
struct list_head s_io;
struct list_head s_more_io;
struct list_head s_files;
...
```

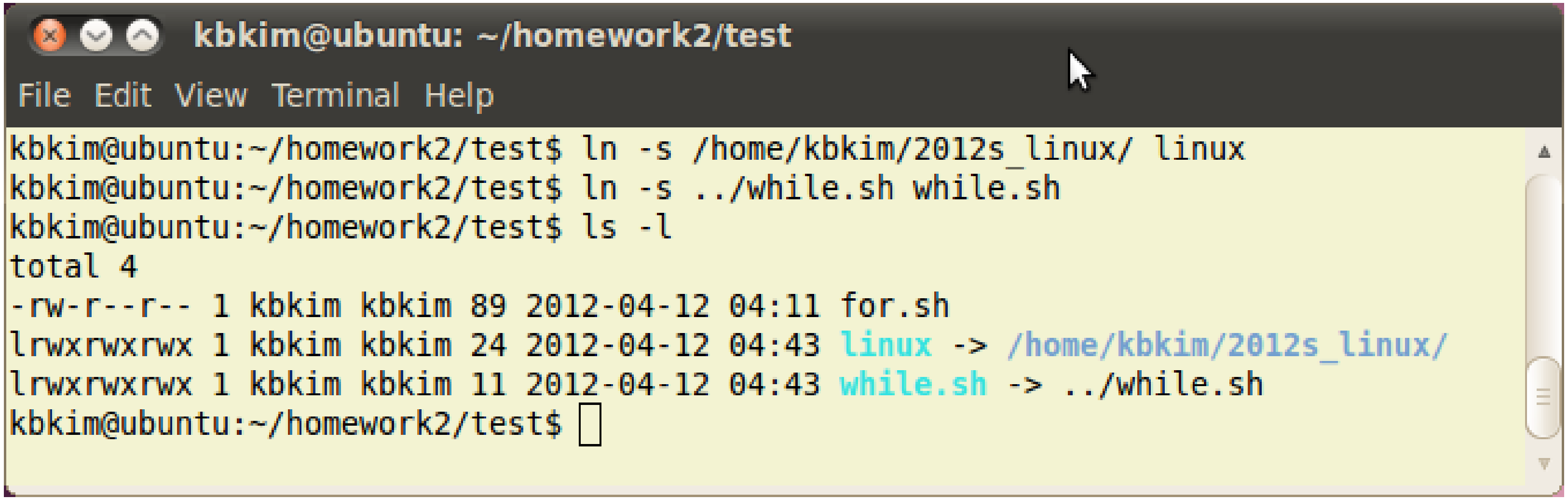
Superblock
- Store the information of a filesystem
- Essential to mount a filesystem
- List of super_block means that multiple filesystems are in a device

Symbolic Links

- A symbolic link (or symlink) is a pseudo-file which behaves as an alternative name for some other file or directory
- The contents of the symlink are the real name pointed to
  - Kernel replace the symlink contents automatically
- Keep a file in one place, but pretend it lives in another
  - To ensure that an obsolete name continues to work for older software
  - To spread data from a single filesystem hierarchy over multiple disk partitions

Examining and creating symbloic links

- “ls –l” shows where a symbolic link points to
  - Different color or suffix with ‘@’
- “ln –s” create a symbolic link



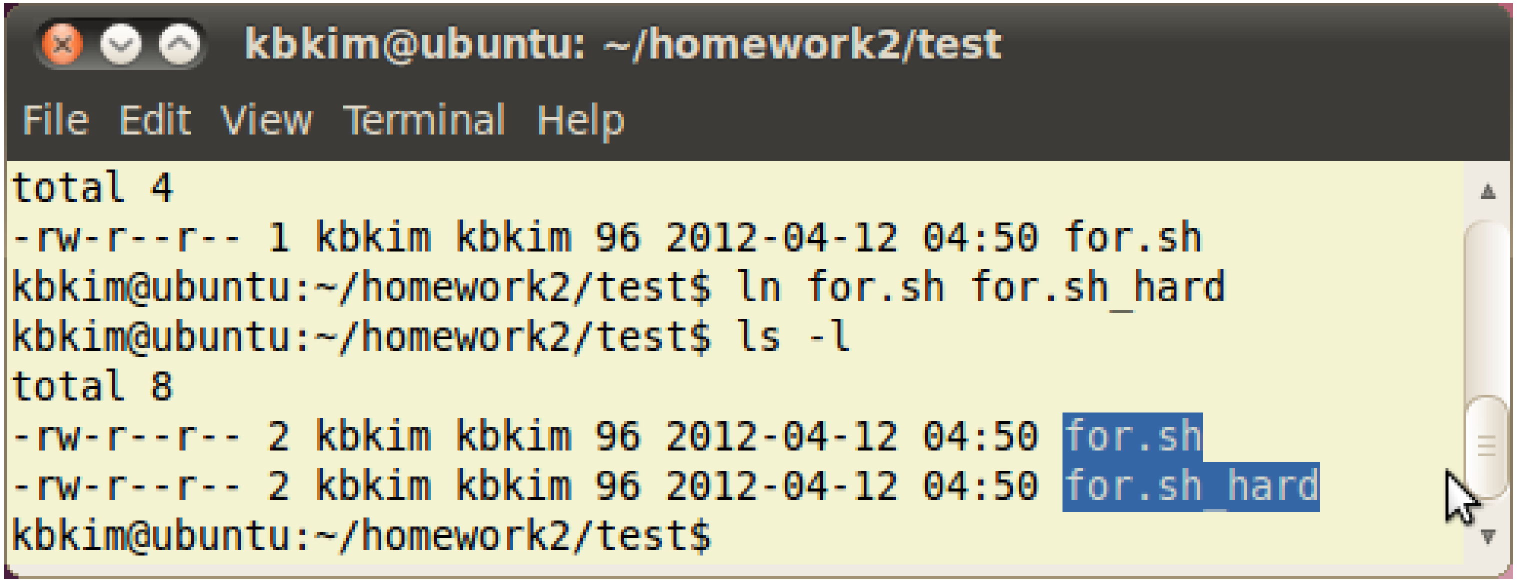
```
kbkim@ubuntu:~/homework2/test$ ln -s /home/kbkim/2012s_linux/ linux
kbkim@ubuntu:~/homework2/test$ ln -s ../while.sh while.sh
kbkim@ubuntu:~/homework2/test$ ls -l
total 4
-rw-r--r-- 1 kbkim kbkim 89 2012-04-12 04:11 for.sh
lrwxrwxrwx 1 kbkim kbkim 24 2012-04-12 04:43 linux -> /home/kbkim/2012s_linux/
lrwxrwxrwx 1 kbkim kbkim 11 2012-04-12 04:43 while.sh -> ../while.sh
kbkim@ubuntu:~/homework2/test$ 
```

Hard Links

- Hard link refers to another file by inode number
- A directory entry contains a name and an inode number
- The modification of file affected to both of the original file and the hard link file
  - But deleting or renaming one does not affect the other

Examining and creating hard links

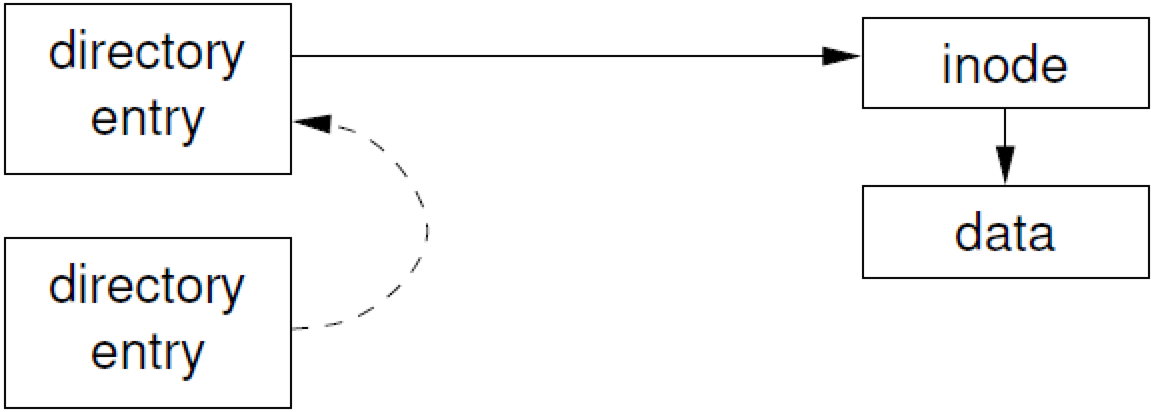
- “ls” shows a hard link file has multiple link count
- “ln” create a hard link



```
total 4
-rw-r--r-- 1 kbkim kbkim 96 2012-04-12 04:50 for.sh
kbkim@ubuntu:~/homework2/test$ ln for.sh for.sh_hard
kbkim@ubuntu:~/homework2/test$ ls -l
total 8
-rw-r--r-- 2 kbkim kbkim 96 2012-04-12 04:50 for.sh
-rw-r--r-- 2 kbkim kbkim 96 2012-04-12 04:50 for.sh_hard
kbkim@ubuntu:~/homework2/test$
```

Symbolic Link vs Hard Link

Symbolic Link



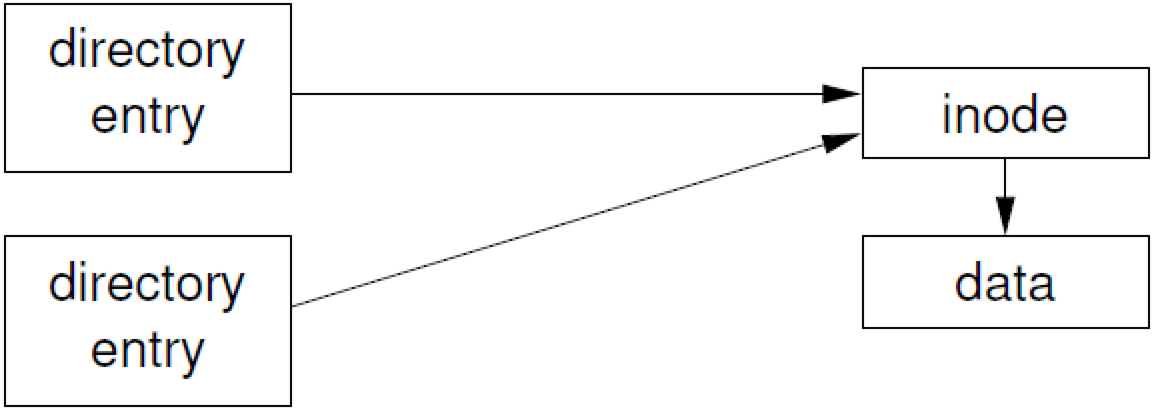
directory entry

directory entry

inode

data

Hard Link



directory entry

directory entry

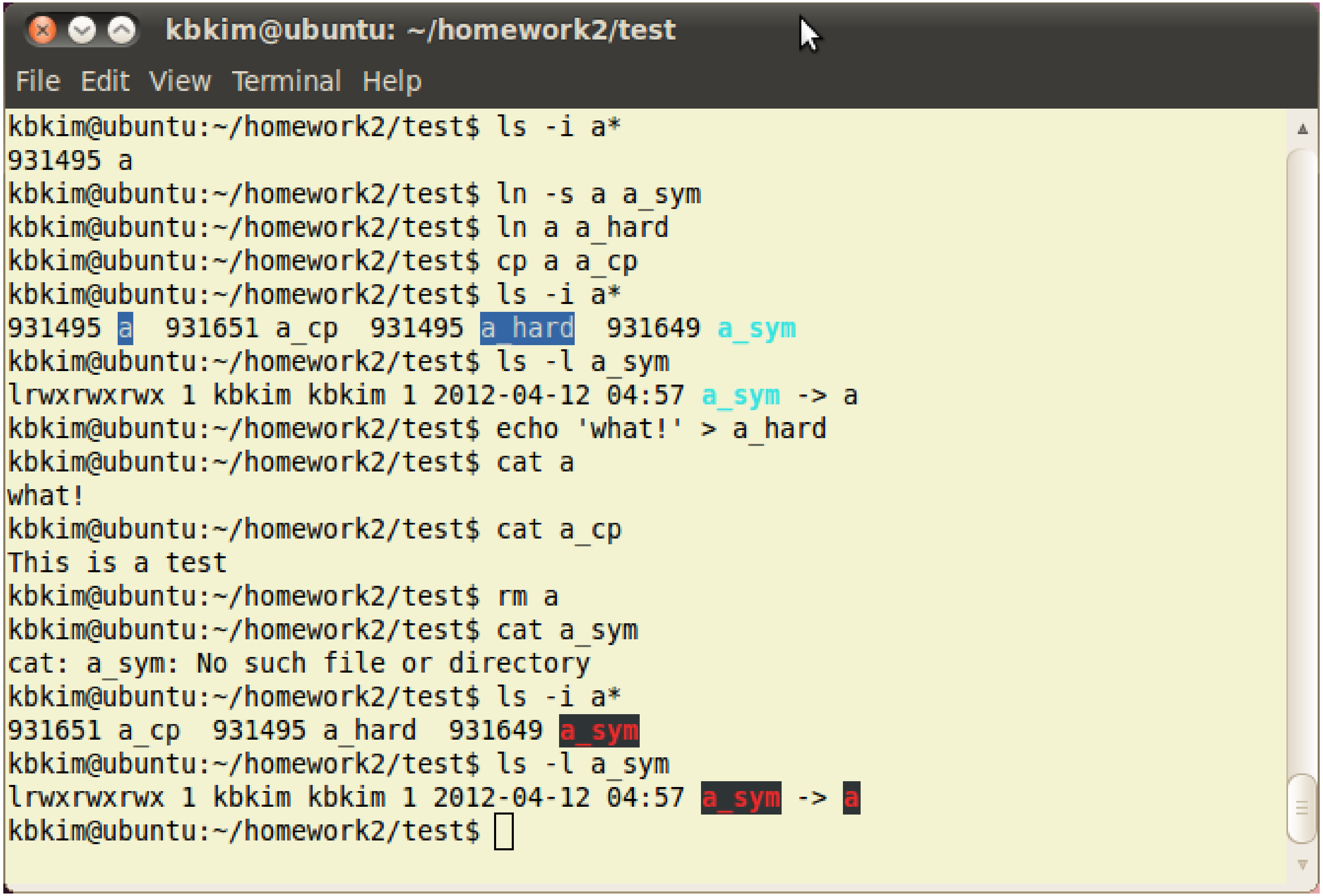
inode

data

Symbolic link vs Hard link

Symbolic links	Hard Links
Distinguishable (pointer)	indistinguishable
Point to any type of file	may not point to a directory
Refer to names	Works by inode numbers
Point on other filesystems	Work within a single filesystem
Renaming or deleting a original file breaks the symbolic link	Renaming or deleting a original file has no effect on the hard link
Take up additional disk space (to store the name pointed to)	Need as much disk space as a directory entry

Inode and symbolic/hard link



```
kbkim@ubuntu:~/homework2/test$ ls -i a*
931495 a
kbkim@ubuntu:~/homework2/test$ ln -s a a_sym
kbkim@ubuntu:~/homework2/test$ ln a a_hard
kbkim@ubuntu:~/homework2/test$ cp a a_cp
kbkim@ubuntu:~/homework2/test$ ls -i a*
931495 a  931651 a_cp  931495 a_hard  931649 a_sym
kbkim@ubuntu:~/homework2/test$ ls -l a_sym
lrwxrwxrwx 1 kbkim kbkim 1 2012-04-12 04:57 a_sym -> a
kbkim@ubuntu:~/homework2/test$ echo 'what!' > a_hard
kbkim@ubuntu:~/homework2/test$ cat a
what!
kbkim@ubuntu:~/homework2/test$ cat a_cp
This is a test
kbkim@ubuntu:~/homework2/test$ rm a
kbkim@ubuntu:~/homework2/test$ cat a_sym
cat: a_sym: No such file or directory
kbkim@ubuntu:~/homework2/test$ ls -i a*
931651 a_cp  931495 a_hard  931649 a_sym
kbkim@ubuntu:~/homework2/test$ ls -l a_sym
lrwxrwxrwx 1 kbkim kbkim 1 2012-04-12 04:57 a_sym -> a
kbkim@ubuntu:~/homework2/test$ 
```

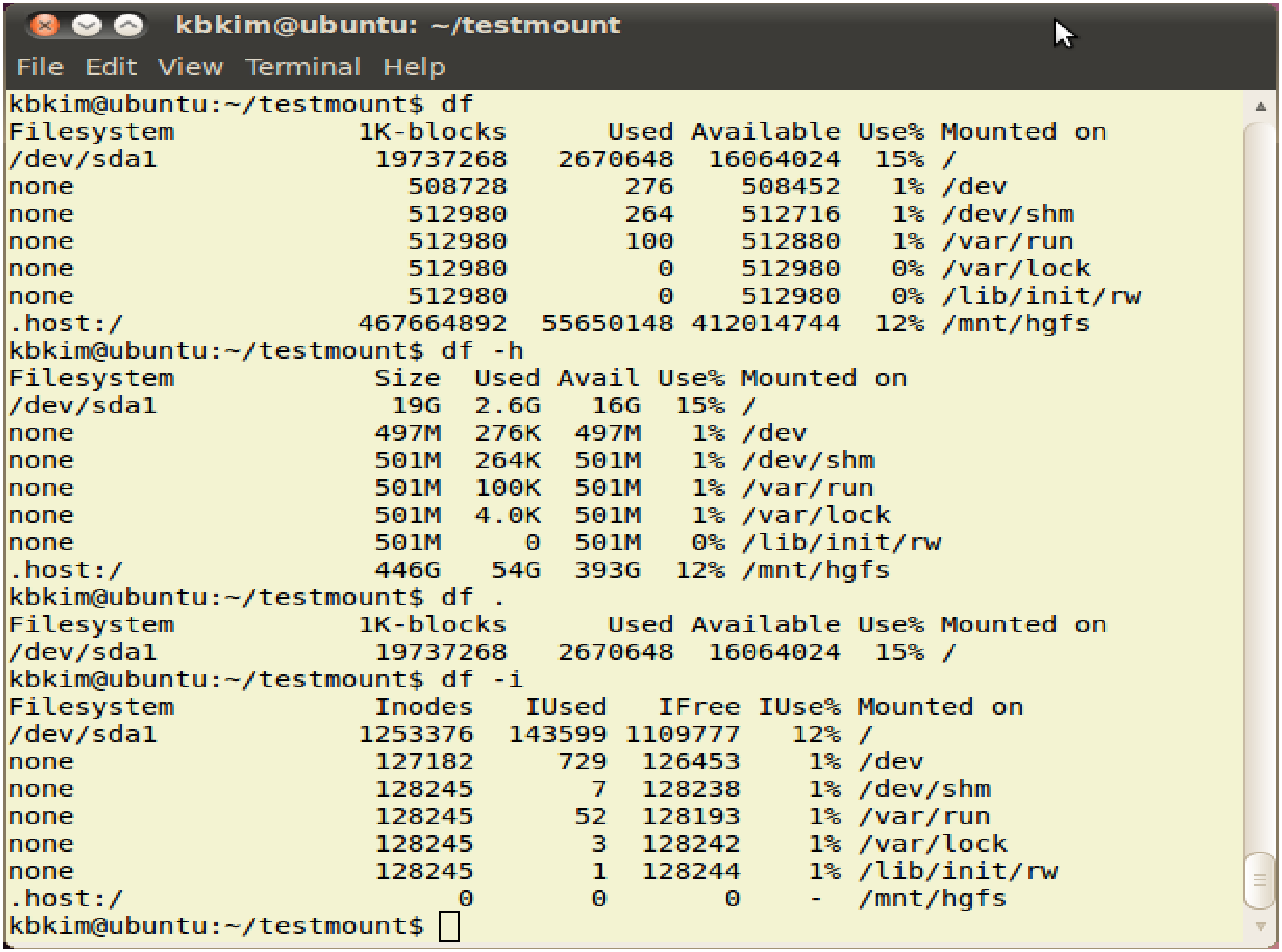
Problems of Filesystem

- Over time, an active filesystem can develop problems
  - It can fill up, causing individual programs or even the entire system to fail
  - It can become corrupted, perhaps due to a power failure or a system crash
  - It can run out of space for inodes, so no new files or directories can be created
- Monitoring and checking filesystems regularly can help prevent and correct problems

Monitoring space : df

- "df" get a list of **free space** on all mounted filesystems
- "-h" option → human-readable option
- "Use%" column shows what percentage of the filesystem is in use
- Directory argument → make it show space on the filesystems those directories are mounted on
- "-i" option → inode usage

Example of DF

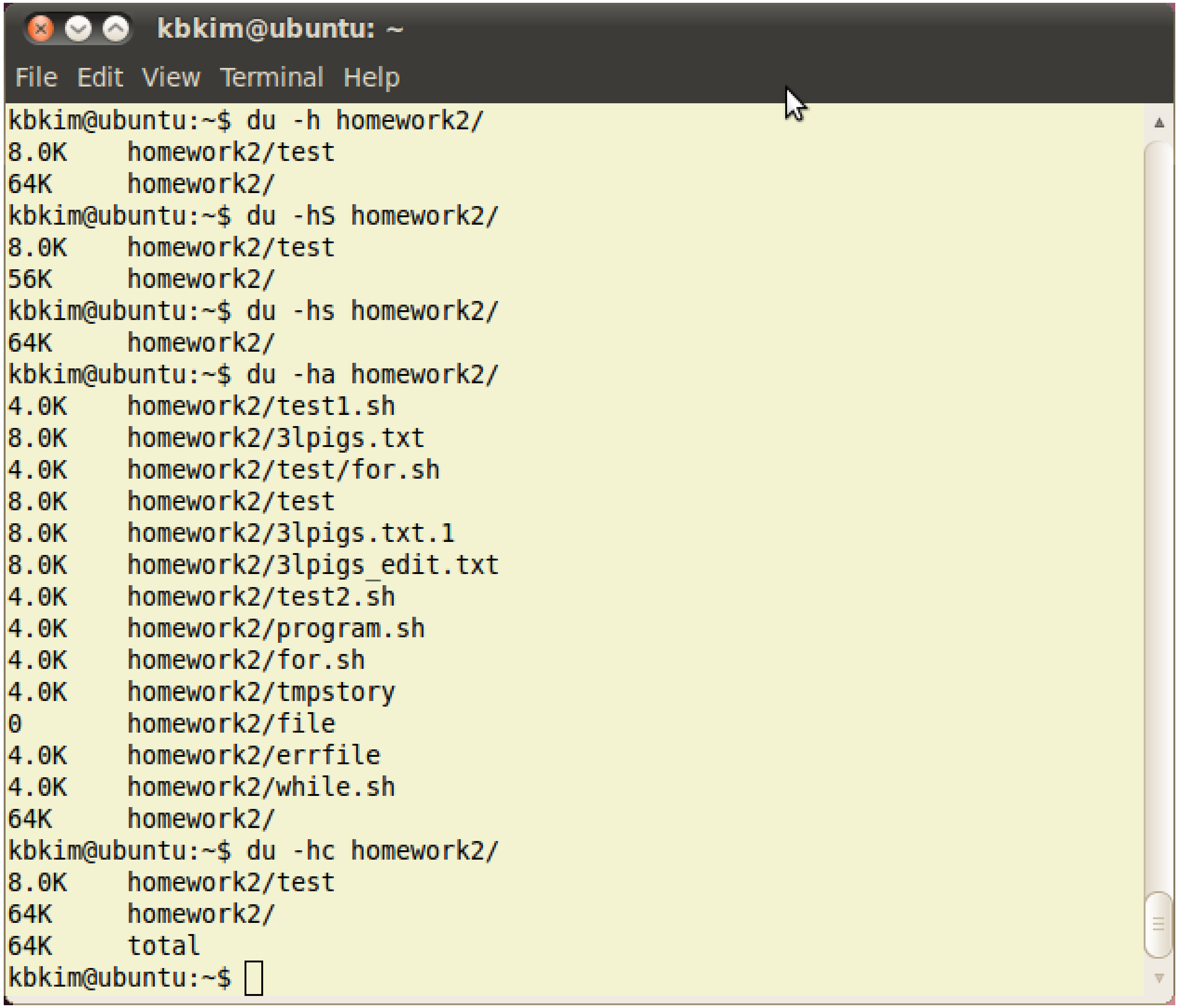


```
kbkim@ubuntu:~/testmount$ df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda1             19737268   2670648  16064024  15% /
none                    508728       276    508452   1% /dev
none                    512980       264    512716   1% /dev/shm
none                    512980       100    512880   1% /var/run
none                    512980         0    512980   0% /var/lock
none                    512980         0    512980   0% /lib/init/rw
.host:/              467664892  55650148 412014744  12% /mnt/hgfs
kbkim@ubuntu:~/testmount$ df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda1              19G  2.6G   16G  15% /
none                  497M  276K  497M   1% /dev
none                  501M  264K  501M   1% /dev/shm
none                  501M  100K  501M   1% /var/run
none                  501M  4.0K  501M   1% /var/lock
none                  501M     0  501M   0% /lib/init/rw
.host:/               446G   54G  393G  12% /mnt/hgfs
kbkim@ubuntu:~/testmount$ df .
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda1             19737268   2670648  16064024  15% /
kbkim@ubuntu:~/testmount$ df -i
Filesystem            Inodes   IUsed   IFree IUse% Mounted on
/dev/sda1            1253376  143599 1109777   12% /
none                  127182     729  126453    1% /dev
none                  128245       7  128238    1% /dev/shm
none                  128245      52  128193    1% /var/run
none                  128245       3  128242    1% /var/lock
none                  128245       1  128244    1% /lib/init/rw
.host:/                    0       0       0    -  /mnt/hgfs
kbkim@ubuntu:~/testmount$ 
```

Monitoring disk usage : du

- “du” shows a summary of the disk usage in a directory tree
- Options
  - “-a” → show all files
  - “-c” → cumulated total for all directories
  - “-h” → human-readable
  - “-S” → exclude subdirectories
  - “-s” → summary for each directory

Example of du



```
kbkim@ubuntu:~$ du -h homework2/
8.0K    homework2/test
64K     homework2/
kbkim@ubuntu:~$ du -hS homework2/
8.0K    homework2/test
56K     homework2/
kbkim@ubuntu:~$ du -hs homework2/
64K     homework2/
kbkim@ubuntu:~$ du -ha homework2/
4.0K    homework2/test1.sh
8.0K    homework2/3lpigs.txt
4.0K    homework2/test/for.sh
8.0K    homework2/test
8.0K    homework2/3lpigs.txt.1
8.0K    homework2/3lpigs_edit.txt
4.0K    homework2/test2.sh
4.0K    homework2/program.sh
4.0K    homework2/for.sh
4.0K    homework2/tmpstory
0       homework2/file
4.0K    homework2/errfile
4.0K    homework2/while.sh
64K     homework2/
kbkim@ubuntu:~$ du -hc homework2/
8.0K    homework2/test
64K     homework2/
64K     total
kbkim@ubuntu:~$ 
```

Finding and repairing filesystem : fsck

- Checks the integrity of a filesystem
  - Can make repairs if necessary
- Tow main parts
  - Driver program, fsck
    - Handles any filesystem type
  - Backend program
    - For each specific filesystem type
    - Ext2 → e3fsck
      - Invoked through fsck
- "–f" option → force checking the filesystem
- Usually it is a bad idea to run fsck on a mounted filesystem

What are Quotas?

- Quotas are a way of limiting the amount of disk space that users may take up
- Some organizations absolutely need to ensure that
  - No user can prevent other users from using a reasonable amount of disk space
  - No user can impede the correct functioning of the system

Hard and soft limits

- Quotas have **hard limits** and **soft limits**
- A user can exceed the soft limit without retribution
  - But only for a certain period of time
    - The grace period
    - Set per-filesystem
  - The user is also warned that the soft limit has been exceeded
- A hard limit may never be exceeded
- If a user tries to exceed a hard limit, the attempt fails

Block and Inode Limits

- Quotas can be set for blocks
  - Limits the amount of data space that may be used
- Quotas can be set for inodes
  - Limits the number of files that may be created

Per-user and Per-group Quotas

- Most quotas are set per-user
  - Each user has his or her own soft limit and hard limit
- Quotas can also be set per-group
  - A group can be given a soft limit and hard limit
- Group quotas apply to all users in a group
- If a group hard limit has been reached, no user in the group may use more space
  - Including users who have not yet reached their individual quota

Options in /etc/fstab

- The options in /etc/fstab specify which filesystems should have quota enabled
  - Add the option *usrquota* to enable user quotas
  - Use *grpquota* to enable group quotas
  - Either or both can be used for each filesystem
- The filesystems with quota enabled should have files called **aquota.user** and **aquota.group** in their root directories
- Mount option
  - “usrquota” and “grpquota”

quotacheck command

- Check the users and groups related to quota regulation
- Option
  - “-c” → creating aquota db
  - “-u” → for user
  - “-g” → for group
  - “-f” → force checking
- “aquota.group” → db for group quota
- “aquota.user” → db for user quota

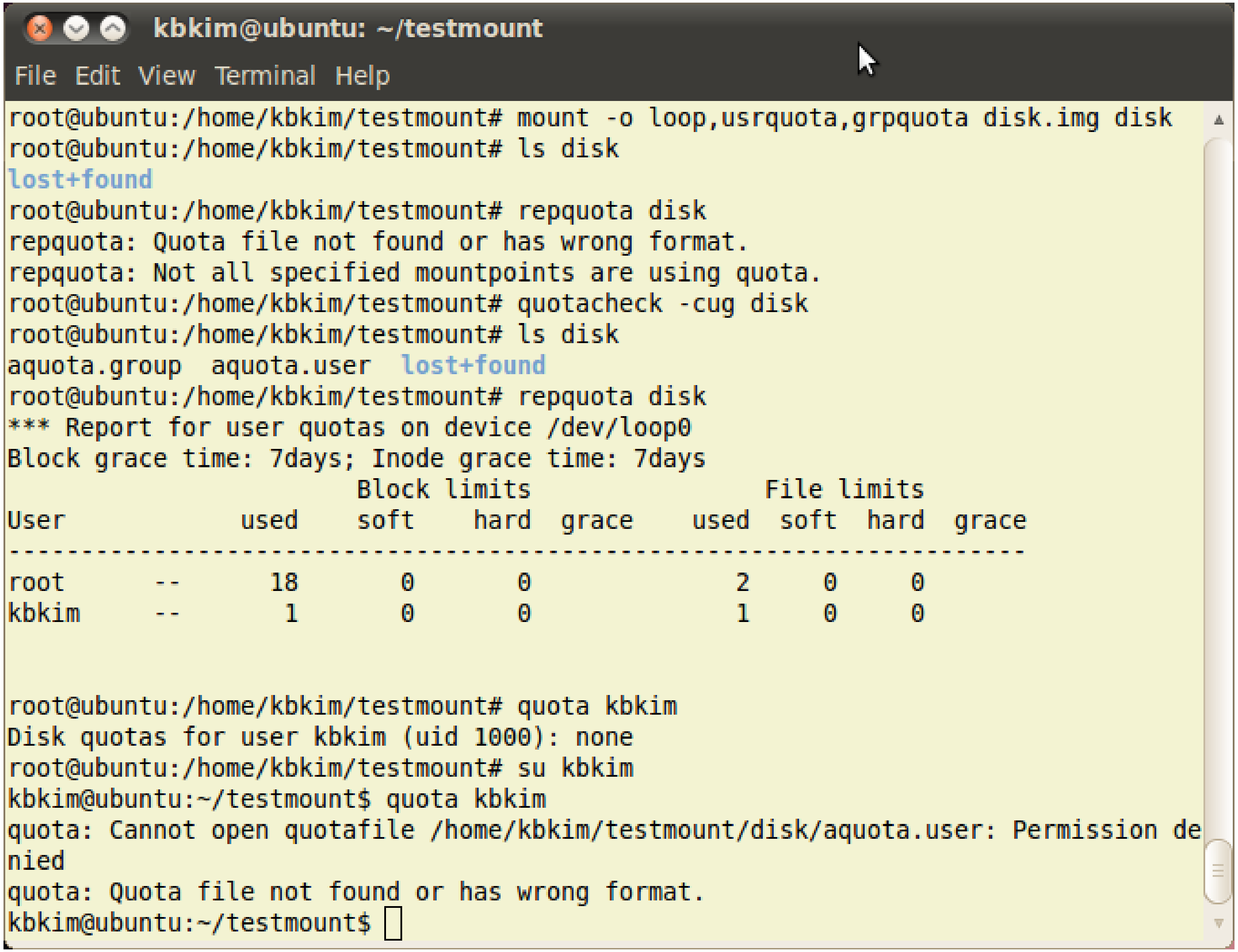
setquota command

- Command line program to alter quota limits for a user or group
- Specify the name of a user or group with –u username or –g groupname
- Specify the filesystem
- Specify the limit to set must be specified in the following order
  - Soft limit for blocks
  - Hard limit for blocks
  - Soft limit for inodes
  - Hard limit for inodes
- Setting any limit to 0 will remove that limit
- –t option : set grace period of block and inode softlimit in secounds
- Examples
  - setquota –u kbkim 10 20 10 20 /dev/hda1
  - setquota –t 3600 3600 /dev/hda1

repquota/quotaon/quotaoff commands

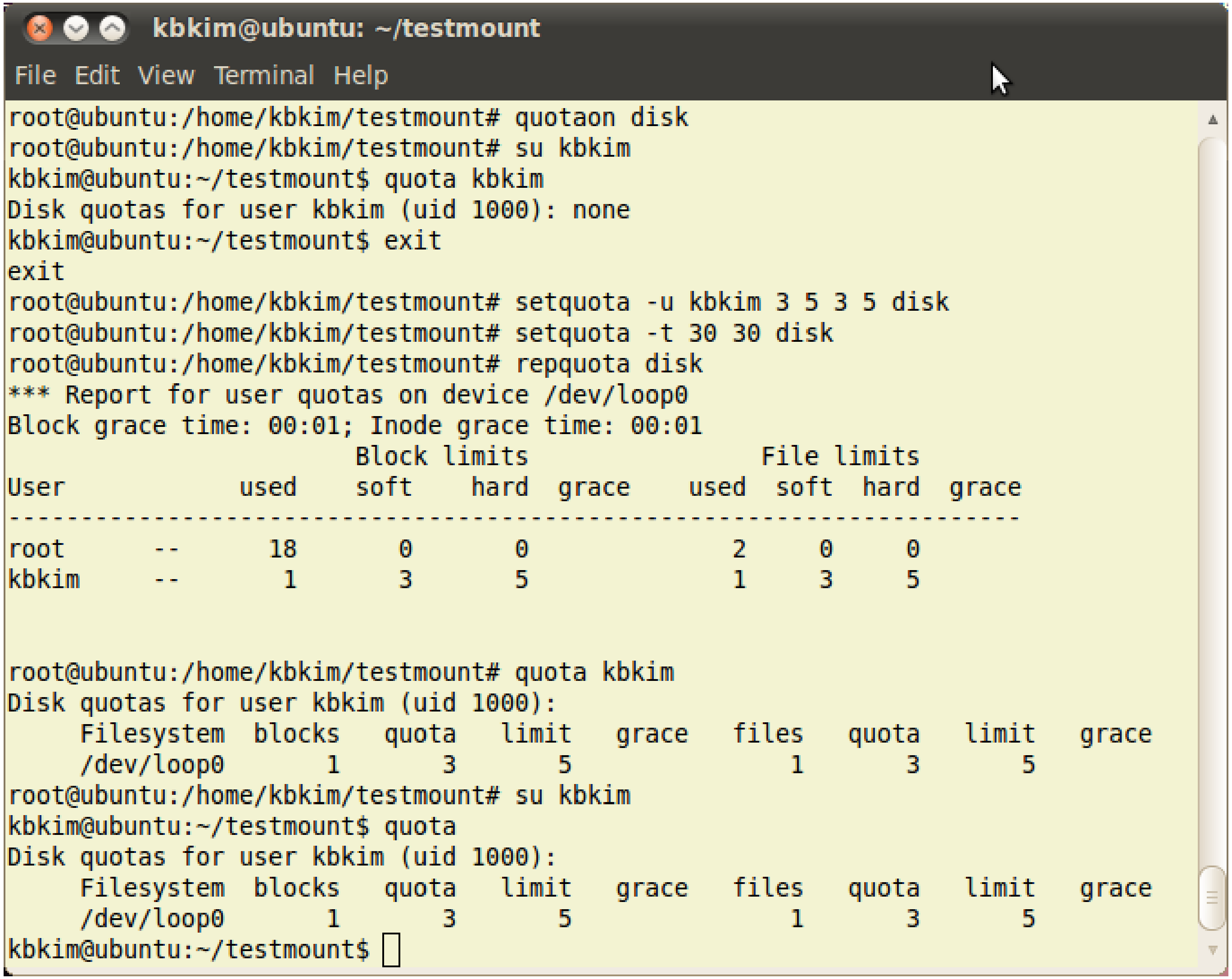
- “quotaon”/”quotaoff”
  - Turns on/off quota support
  - Can only be done by root
- “Repquota”
  - Prints information about the quota limits assigned to each user
  - Root privilege
  - “-a” → all filesystems
  - “-g” → group quotas
  - “-v” → more complete information
- “quota”
  - For each users to check their quota

Example of quota



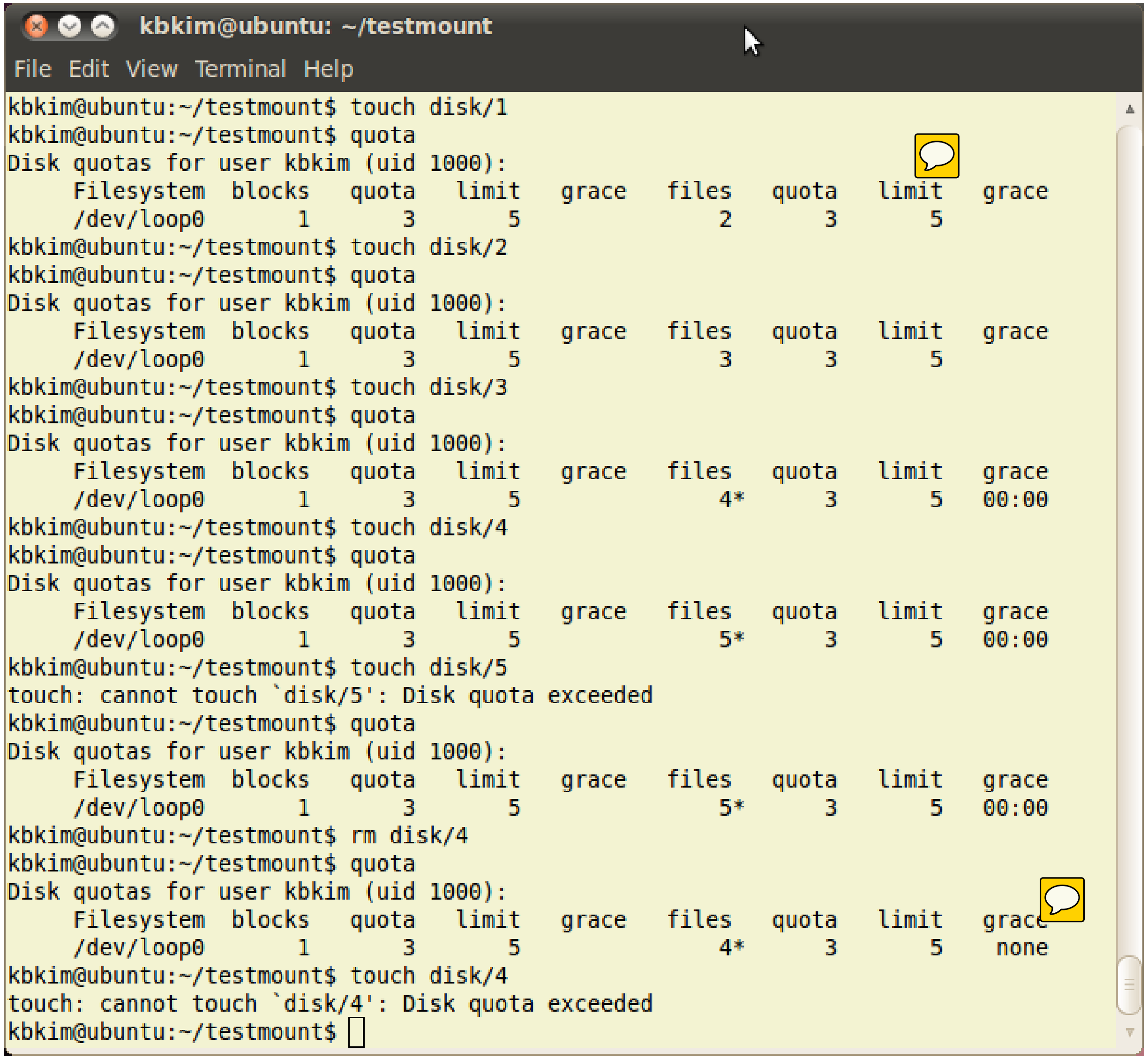
```
root@ubuntu:/home/kbkim/testmount# mount -o loop,usrquota,grpquota disk.img disk
root@ubuntu:/home/kbkim/testmount# ls disk
lost+found
root@ubuntu:/home/kbkim/testmount# repquota disk
repquota: Quota file not found or has wrong format.
repquota: Not all specified mountpoints are using quota.
root@ubuntu:/home/kbkim/testmount# quotacheck -cug disk
root@ubuntu:/home/kbkim/testmount# ls disk
aquota.group  aquota.user  lost+found
root@ubuntu:/home/kbkim/testmount# repquota disk
*** Report for user quotas on device /dev/loop0
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --      18       0       0              2     0     0
kbkim     --       1       0       0              1     0     0


root@ubuntu:/home/kbkim/testmount# quota kbkim
Disk quotas for user kbkim (uid 1000): none
root@ubuntu:/home/kbkim/testmount# su kbkim
kbkim@ubuntu:~/testmount$ quota kbkim
quota: Cannot open quotafile /home/kbkim/testmount/disk/aquota.user: Permission de
nied
quota: Quota file not found or has wrong format.
kbkim@ubuntu:~/testmount$ 
```

Example of setquota



```
root@ubuntu:/home/kbkim/testmount# quotaon disk
root@ubuntu:/home/kbkim/testmount# su kbkim
kbkim@ubuntu:~/testmount$ quota kbkim
Disk quotas for user kbkim (uid 1000): none
kbkim@ubuntu:~/testmount$ exit
exit
root@ubuntu:/home/kbkim/testmount# setquota -u kbkim 3 5 3 5 disk
root@ubuntu:/home/kbkim/testmount# setquota -t 30 30 disk
root@ubuntu:/home/kbkim/testmount# repquota disk
*** Report for user quotas on device /dev/loop0
Block grace time: 00:01; Inode grace time: 00:01
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --      18       0       0              2     0     0
kbkim     --       1       3       5              1     3     5


root@ubuntu:/home/kbkim/testmount# quota kbkim
Disk quotas for user kbkim (uid 1000):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
     /dev/loop0       1       3       5               1       3       5
root@ubuntu:/home/kbkim/testmount# su kbkim
kbkim@ubuntu:~/testmount$ quota
Disk quotas for user kbkim (uid 1000):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
     /dev/loop0       1       3       5               1       3       5
kbkim@ubuntu:~/testmount$ 
```

Example of quota



```
kbkim@ubuntu:~/testmount$ touch disk/1
kbkim@ubuntu:~/testmount$ quota
Disk quotas for user kbkim (uid 1000):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
     /dev/loop0       1       3       5               2       3       5
kbkim@ubuntu:~/testmount$ touch disk/2
kbkim@ubuntu:~/testmount$ quota
Disk quotas for user kbkim (uid 1000):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
     /dev/loop0       1       3       5               3       3       5
kbkim@ubuntu:~/testmount$ touch disk/3
kbkim@ubuntu:~/testmount$ quota
Disk quotas for user kbkim (uid 1000):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
     /dev/loop0       1       3       5              4*       3       5   00:00
kbkim@ubuntu:~/testmount$ touch disk/4
kbkim@ubuntu:~/testmount$ quota
Disk quotas for user kbkim (uid 1000):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
     /dev/loop0       1       3       5              5*       3       5   00:00
kbkim@ubuntu:~/testmount$ touch disk/5
touch: cannot touch `disk/5': Disk quota exceeded
kbkim@ubuntu:~/testmount$ quota
Disk quotas for user kbkim (uid 1000):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
     /dev/loop0       1       3       5              5*       3       5   00:00
kbkim@ubuntu:~/testmount$ rm disk/4
kbkim@ubuntu:~/testmount$ quota
Disk quotas for user kbkim (uid 1000):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
     /dev/loop0       1       3       5              4*       3       5    none
kbkim@ubuntu:~/testmount$ touch disk/4
touch: cannot touch `disk/4': Disk quota exceeded
kbkim@ubuntu:~/testmount$ 
```