information technology & management

viabiliit

INTRO TO OPEN SOURCE

ILLINOIS INSTITUTE OF TECHNOLOGY

Administering Filesystems

Sean Hughes-Durkin

ITMO/IT-O 456 Fall 2017

Information Technology & Management Programs

School of Applied Technology

Slides based on Eckert, Jason W., *Linux+ Guide to Linux+ Certification 3ed*; Thomson Course Technology 2012

Objectives

At the end of this lesson students should be able to:

- Identify the structure and types of device files in the /dev directory
- Understand common filesystem types and their features
- Mount and unmount filesystems to and from the Linux directory tree

Objectives

At the end of this lesson students should be able to:

- Create and manage filesystems on floppy disks, CDs, DVDs, USB storage devices, FireWire storage devices, and hard disk partitions
- Create and use ISO images
- Use the LVM to create and manage logical volumes

Objectives

At the end of this lesson students should be able to:

- Monitor free space on mounted filesystems
- Check filesystems for errors
- Use hard disk quotas to limit user space usage

The /dev Directory

- Device file
  - Represents a device
  - Typically found in the /dev directory
  - Specifies how to transfer data to and from the device
  - Character or Block Devices
- Character devices
  - Transfer data to and from the system one character at a time

The /dev Directory

- Block devices
  - Transfer chunks or blocks of data using physical memory as a buffer
  - Fast data transfer
  - Represented by block device files
  - Floppy disks, CD-ROMs, DVDs, USB flash drives, hard disk drives, SSDs

The /dev Directory

Device File	Description	Block or Character
/dev/fd0	First floppy disk on the system	Block
/dev/fd1	Second floppy disk on the system	Block
/dev/hda1	First primary partition on the first IDE hard disk drive (primary master)	Block
/dev/hdb1	First primary partition on the second IDE hard disk drive (primary slave)	Block
/dev/hdc1	First primary partition on the third IDE hard disk drive (secondary master)	Block
/dev/hdd1	First primary partition on the fourth IDE hard disk drive (secondary slave)	Block
/dev/sda1	First primary partition on the first SCSI/SAS/SATA hard disk drive	Block
/dev/sdb1	First primary partition on the second SCSI/SAS/SATA hard disk drive	Block

Table 5-1: Common device files

The /dev Directory

Device File	Description	Block or Character
/dev/cdrom	Default CD-ROM/DVD-ROM drive	Block
/dev/tty0	First local terminal on the system (Ctrl-Alt-F1)	Character
/dev/tty1	Second local terminal on the system (Ctrl-Alt-F2)	Character
/dev/ttyS0	First serial port on the system (COM1)	Character
/dev/ttyS1	Second serial port on the system (COM2)	Character
/dev/psaux	PS/2 mouse port	Character
/dev/lp0	First parallel port on the system (LPT1)	Character
/dev/null	A device that represents nothing; any data sent to this device is discarded (the proverbial “bit bucket”)	Character
/dev/st0	The first SCSI tape device on the system	Character
/dev/usb	USB device files	Character

Table 5-1: Common device files

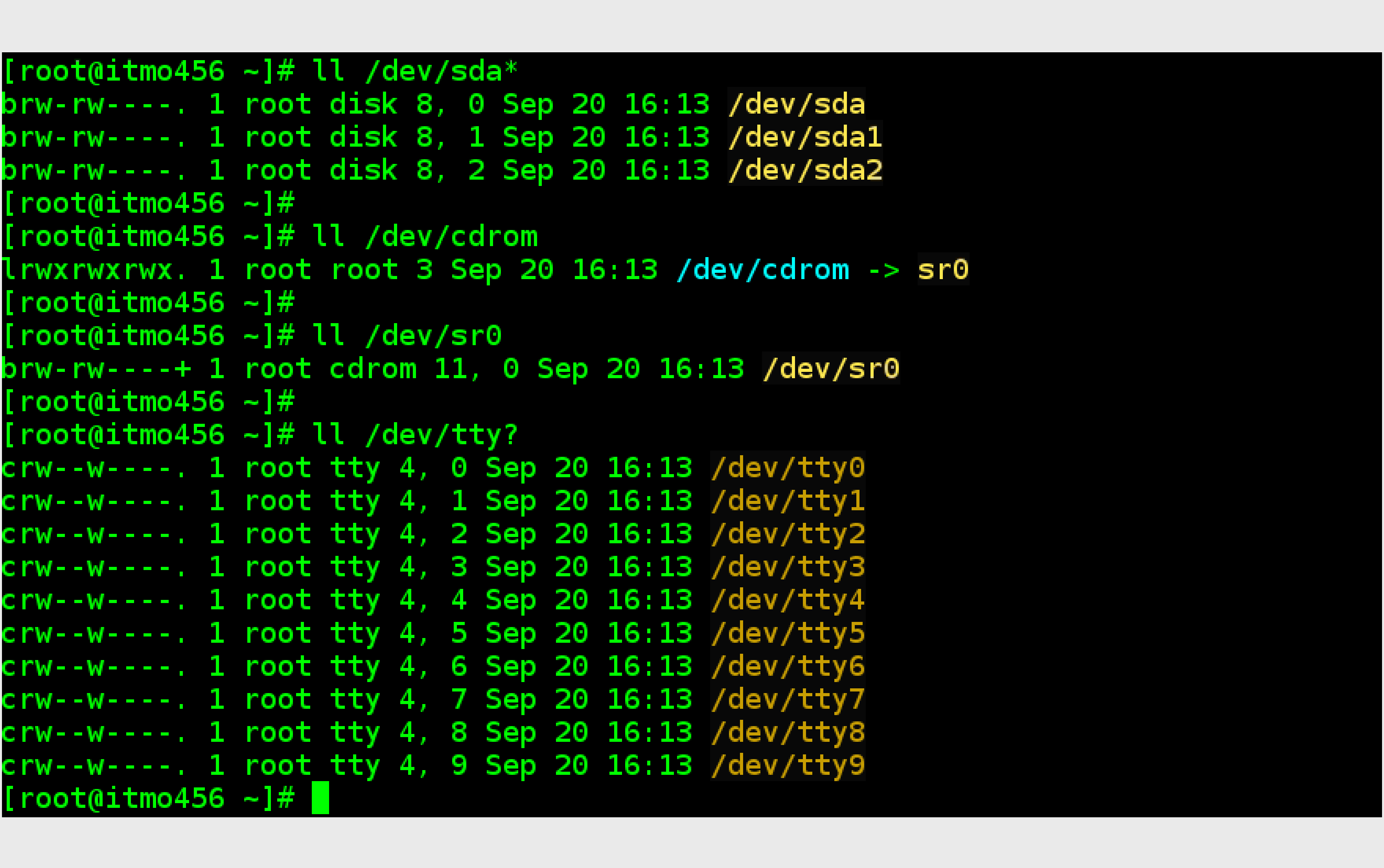
The /dev Directory

- Major number
    - Used by the kernel to identify what device driver to call to interact properly with a given category of hardware
    - Number in the device file that points to the device’s driver in the Linux kernel
    - Several different devices can share the same major number if they are of the same general type
    - https://www.kernel.org/doc/html/v4.11/admin-guide/devices.html

The /dev Directory

- Minor number
  - Used by the kernel to identify which specific hardware device, within a given category, to use a driver to communicate with
  - Indicates the particular device
- Device file type (block or character), major number, and minor number make up a device file’s unique characteristics

The /dev Directory

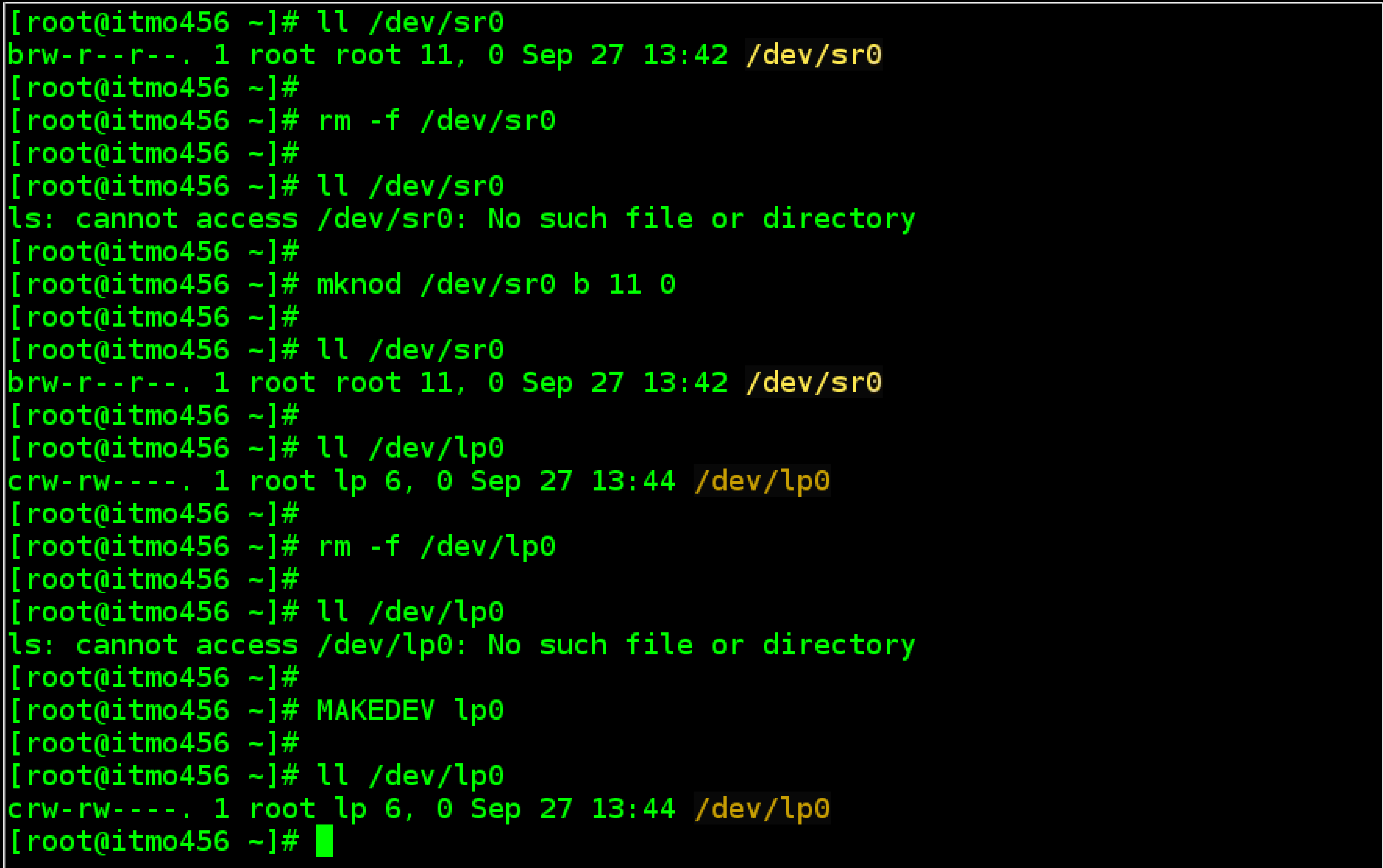


```
[root@itmo456 ~]# ll /dev/sda*
brw-rw----. 1 root disk 8, 0 Sep 20 16:13 /dev/sda
brw-rw----. 1 root disk 8, 1 Sep 20 16:13 /dev/sda1
brw-rw----. 1 root disk 8, 2 Sep 20 16:13 /dev/sda2
[root@itmo456 ~]#
[root@itmo456 ~]# ll /dev/cdrom
lrwxrwxrwx. 1 root root 3 Sep 20 16:13 /dev/cdrom -> sr0
[root@itmo456 ~]#
[root@itmo456 ~]# ll /dev/sr0
brw-rw----+ 1 root cdrom 11, 0 Sep 20 16:13 /dev/sr0
[root@itmo456 ~]#
[root@itmo456 ~]# ll /dev/tty?
crw--w----. 1 root tty 4, 0 Sep 20 16:13 /dev/tty0
crw--w----. 1 root tty 4, 1 Sep 20 16:13 /dev/tty1
crw--w----. 1 root tty 4, 2 Sep 20 16:13 /dev/tty2
crw--w----. 1 root tty 4, 3 Sep 20 16:13 /dev/tty3
crw--w----. 1 root tty 4, 4 Sep 20 16:13 /dev/tty4
crw--w----. 1 root tty 4, 5 Sep 20 16:13 /dev/tty5
crw--w----. 1 root tty 4, 6 Sep 20 16:13 /dev/tty6
crw--w----. 1 root tty 4, 7 Sep 20 16:13 /dev/tty7
crw--w----. 1 root tty 4, 8 Sep 20 16:13 /dev/tty8
crw--w----. 1 root tty 4, 9 Sep 20 16:13 /dev/tty9
[root@itmo456 ~]#
```

The /dev Directory

- **`mknod`** command
  - Can be used to re-create a corrupted device
  - Must know file type, major, and minor numbers
  - List of devices loaded in kernel **`/proc/devices`**
- **`/dev/MAKEDEV`** command
  - Can be used to re-create a device file based on its common name
  - Useful if you don’t know some of the information required for the **`mknod`** command

The /dev Directory



```
[root@itmo456 ~]# ll /dev/sr0
brw-r--r--. 1 root root 11, 0 Sep 27 13:42 /dev/sr0
[root@itmo456 ~]#
[root@itmo456 ~]# rm -f /dev/sr0
[root@itmo456 ~]#
[root@itmo456 ~]# ll /dev/sr0
ls: cannot access /dev/sr0: No such file or directory
[root@itmo456 ~]#
[root@itmo456 ~]# mknod /dev/sr0 b 11 0
[root@itmo456 ~]#
[root@itmo456 ~]# ll /dev/sr0
brw-r--r--. 1 root root 11, 0 Sep 27 13:42 /dev/sr0
[root@itmo456 ~]#
[root@itmo456 ~]# ll /dev/lp0
crw-rw----. 1 root lp 6, 0 Sep 27 13:44 /dev/lp0
[root@itmo456 ~]#
[root@itmo456 ~]# rm -f /dev/lp0
[root@itmo456 ~]#
[root@itmo456 ~]# ll /dev/lp0
ls: cannot access /dev/lp0: No such file or directory
[root@itmo456 ~]#
[root@itmo456 ~]# MAKEDEV lp0
[root@itmo456 ~]#
[root@itmo456 ~]# ll /dev/lp0
crw-rw----. 1 root lp 6, 0 Sep 27 13:44 /dev/lp0
[root@itmo456 ~]#
```

Filesystems

- Filesystem
  - Organization imposed on a physical storage medium used to manage the storage and retrieval of data
  - All storage media need to contain filesystem to be used
- Formatting
  - Creating a filesystem on a device
- All storage media need to contain a filesystem before they can be used

Special Filesystems

- Swap filesystems
  - Allows RAM to “swap out” and “swap in” data between memory and disk, as needed.
  - Downside: Performance hit occurs
  - Upside: Better than running out of memory
- Logical Volume Manager (LVM)
  - Creates pools of storage space, called Volume Groups
  - Provides flexibility in growing/shrinking storage

Partition Connections

- The root Partition
  - At least one disk partition is required
  - Assigned to root (/)
- The Other Partitions
  - Common to have more than one disk partition
  - Typically assigned to directories, such as
    - /home
    - /var
    - /tmp

Partition Connection to the filesystem

- Connected by mounting the partition
  - Manually using the mount command
  - Automatically at boot via the **`/etc/fstab`** file
- Mount point in filesystem determines where partition will be used
- Any file stored in that mount point is stored on the physical disk partition

Filesystem Differences

- Windows
  - Storage uses drive letters, example: **`C:`**, **`D:`**
  - Understands VFAT
  - Uses primarily NTFS
- Linux
  - Storage fits into directory structure, Example: **`/`**
  - Understands VFAT
  - Can support NTFS, but needs additional kernel drivers (increasingly included by default)

Filesystem Types

Filesystem	Description
bfs	Boot File System—a small bootable filesystem used to hold the files necessary for system startup; it is commonly used on UNIX systems
cdfs	Compact disc filesystem—used to view all tracks and data on a CD-ROM as normal files
cramfs	Compressed ROM filesystem—A read-only filesystem typically used on embedded Linux systems to host system files in a small amount of space
ext2	Second extended filesystem; tradiational filesystem used on lInux, supports Access Control Lists (individual user permissions). Retains name from being a new version of the original extended filesystem, based on the Minix filesystem

Table 5-2: Common Linux filesystems

Filesystem Types

Filesystem	Description
ext3	Third extended filesystem—a variation on ext2 that allows for journaling and has faster startup and recovery time
ext4	Fourth extended filesystem; a variation on ext3 that has larger filesystem support and speed enhancements
hfs, hfsplus	Hierarchical File System—a filesystem native to Apple Macintosh computers
hpfs	High Performance File System—an IBM-proprietary OS/2 filesystem that provides long file name support and is optimized to manipulate data on large disk volumes
iso9660	The CD-ROM filesystem—originated from the International Standards Organization recommendation 9660 and used to access data stored on CD-ROMs

Table 5-2: Common Linux filesystems

LINUX & OPEN SOURCE

Filesystem Types

Filesystem	Description
minix	The MINIX filesystem—filesystem used by Linus Torvalds in the early days of Linux development
msdos	FAT filesystem─filesystem used by DOS & early Windows
ntfs	New Technology File System—Microsoft-proprietary filesystem developed for its Windows operating systems
reiserfs	The REISERFS filesystem—a journaling filesystem similar to ext3 and more suited for use with databases
udf	The Universal Disk Format filesystem—filesystem used by software systems that write to a CD-RW or DVD-RW
vfat	DOS FAT filesystem with long file name support
vxfs	Veritas filesystem—journaling filesystem offering large file support & ACLs; commonly used in major UNIX versions

Table 5-2: Common Linux filesystems

Filesystem Types

Filesystem	Description
xfs	X filesystem—high performance filesystem created by Silicon Graphics for use on IRIX UNIX systems. Many Linux sysadmins prefer to use xfs on systems that quickly need to write large numbers of files to the hard disk
zfs	Zettabyte File System—Very high performance 128-bit FS developed for Sun Solaris; has integrated volume management and filesystem; transaction-based with no block ever changed until a transaction is complete. Instantaneous snapshots and clones; fast native backup & restore; highly scalable with built-in compression and a simplified administration model. Widely used in industry for very large data stores. Volumes can span multiple drives. May be in future release of Mac OS/X.

Table 5-2: Common Linux filesystems

Filesystem Types

Filesystem	Description
nilfs	New Implementation of a Log-structured File System: writes all data in a continuous log-like format that is appended to & never overwritten; reduces seek times & minimizes data losses that occur with Linux filesystems. Included in the Linux kernel as of version 2.6.30
tux3	Write-anywhere, atomic commit file system using B-Trees instead of H-trees (used on ext3); attempts to avoid traditional journaling using a recovery logic which allows it to recover upon remounting. Open Source, and aims to be better than ZFS. Not production ready but but actively under development.

Table 5-2: Common Linux filesystems

Mounting

- Mounting
  - Making a device accessible to users via the logical directory tree
  - Term originated in the 1960s, when information was stored on large tape reels that had to be mounted on computers to make the data available
- Mount point
  - Directory to which a device is attached
  - Any existing directory can be a mount point

Mounting

◆Mount point

- Current Linux distributions mount removable media at **`/run/media`**
- Example: CD-ROMs mount at **`/media/cdrom`**
- If a mount point directory has contents, contents cannot be accessed while a device is mounted there
  - In order to prevent making files inaccessible, create empty directories used specifically for mounting devices

Mounting

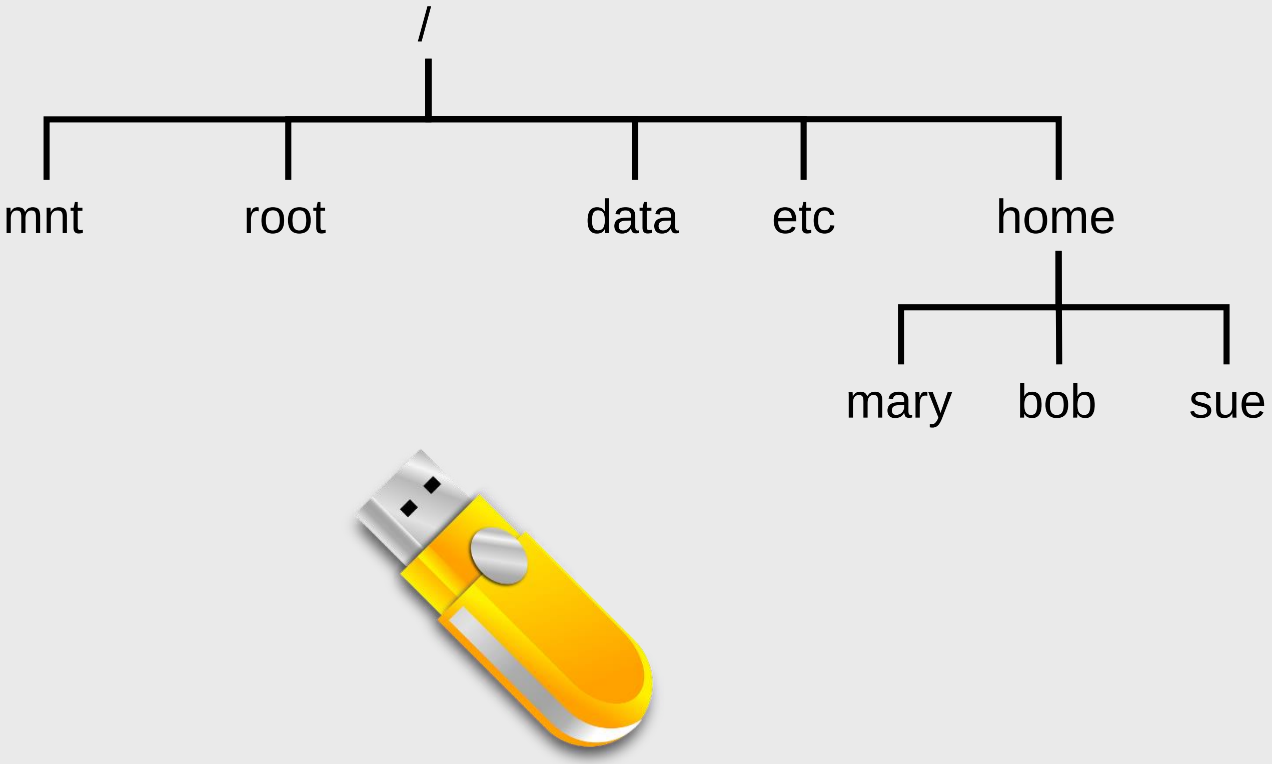


Figure 5-1: The filesystem prior to mounting

Mounting

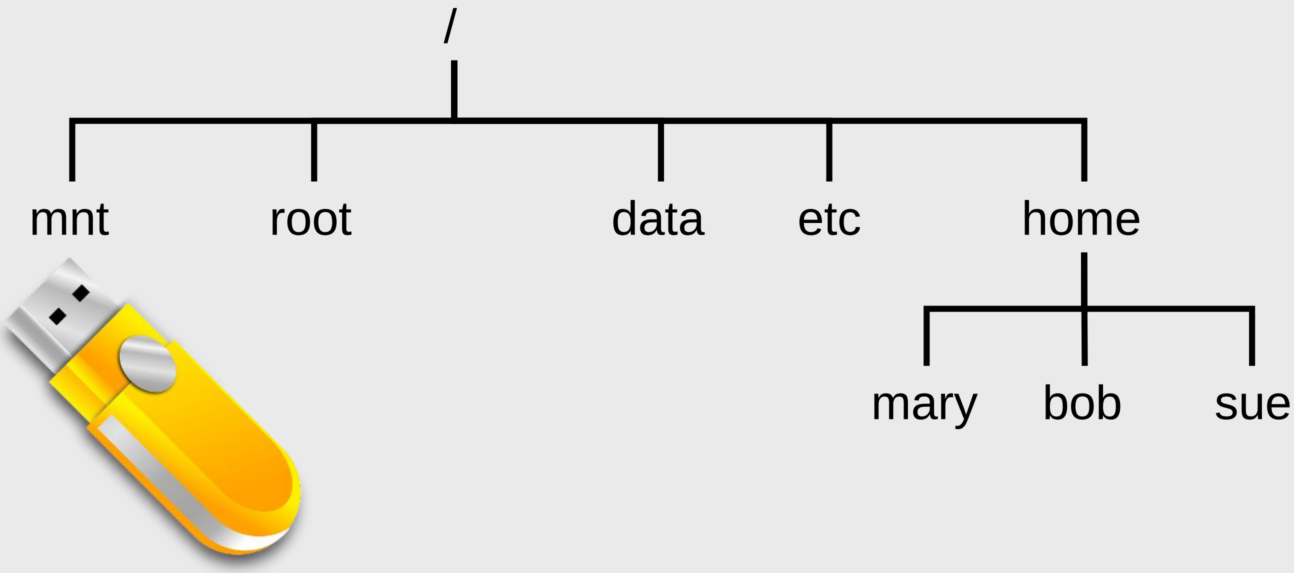
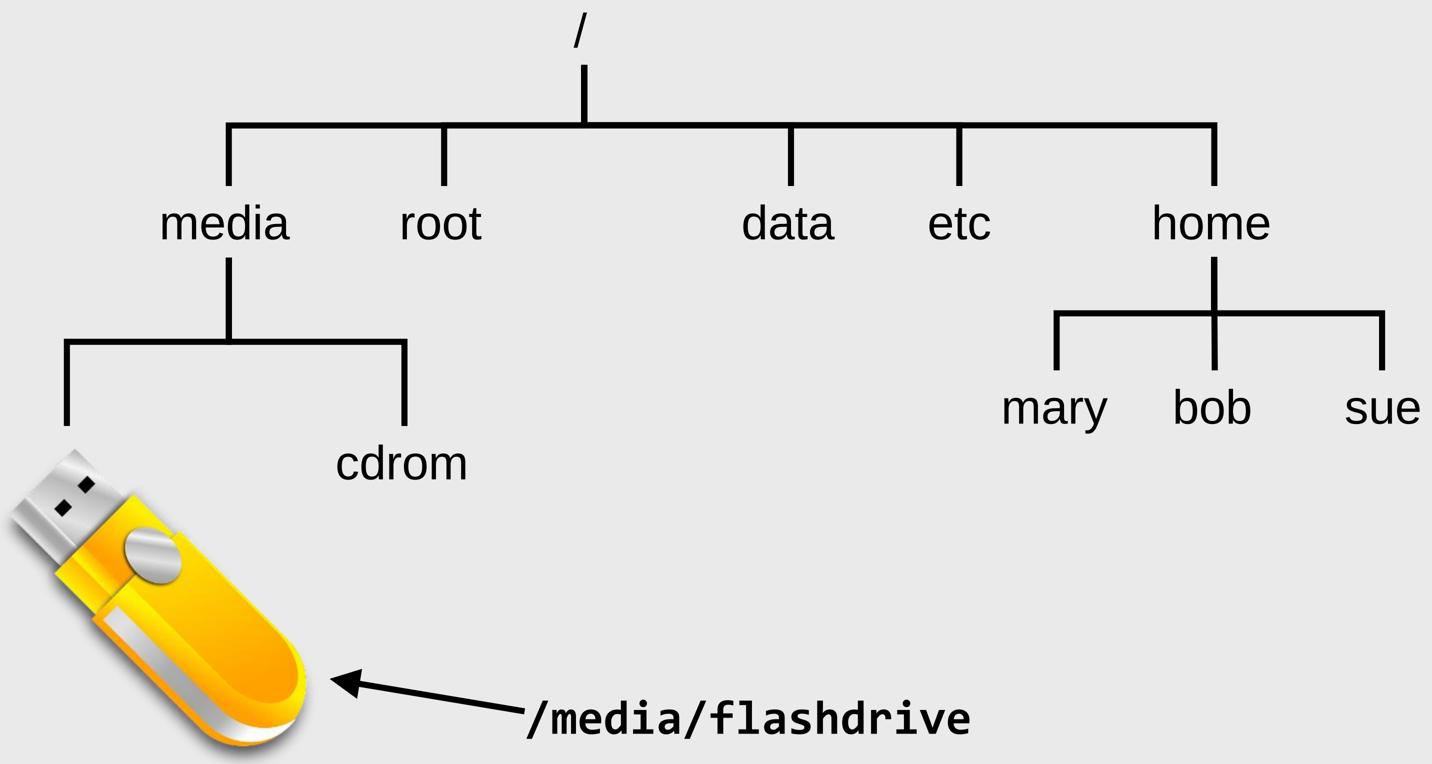


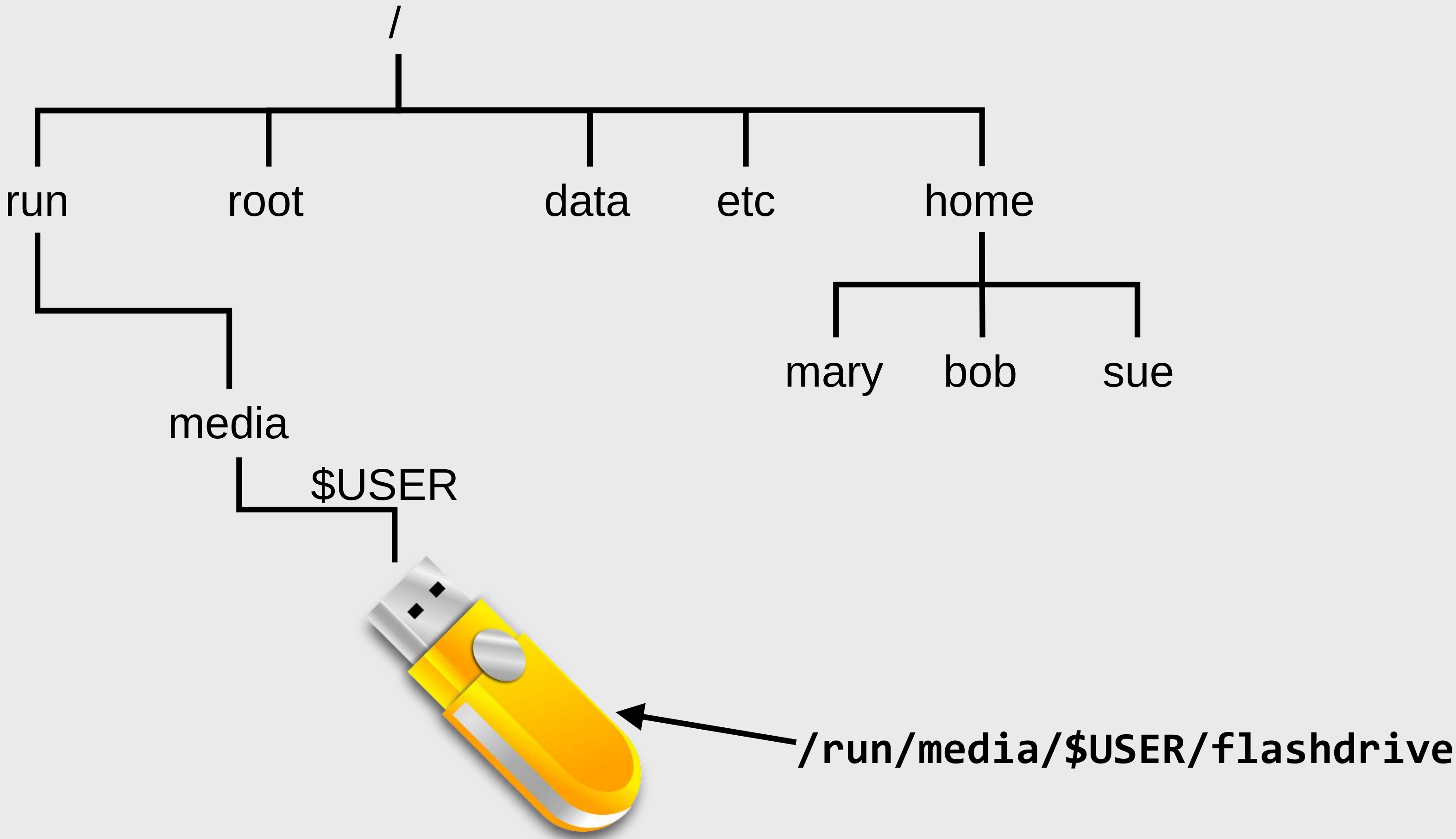
Figure 5-2: The directory structure after mounting a flash drive *(old method)*

Mounting



The directory structure after mounting a floppy device *(slightly less old method)*

Mounting



The directory structure after mounting a floppy device *(current method)*

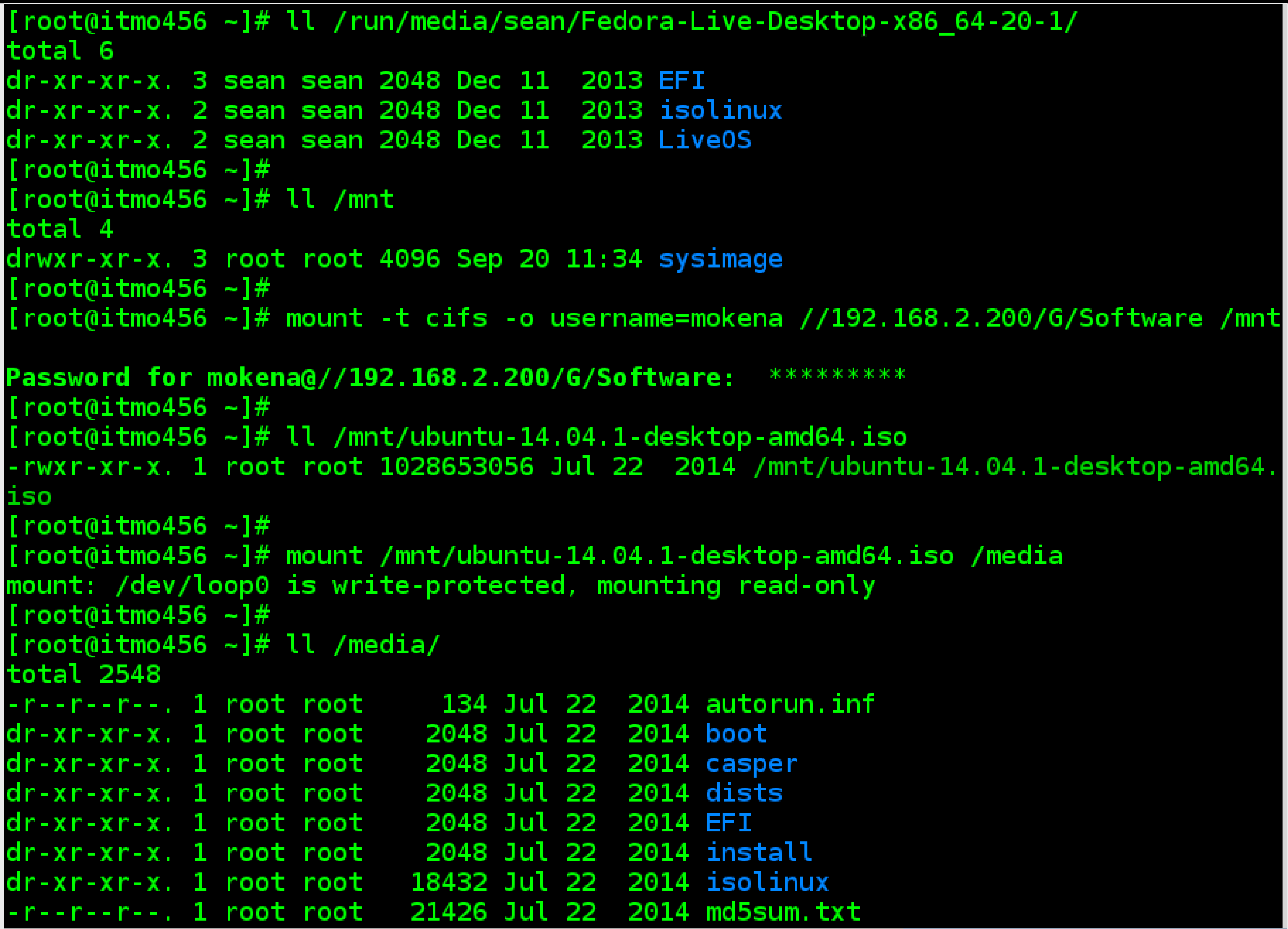
Mounting

- Root filesystem
  - When Linux filesystem is first turned on, a filesystem on the hard drive is mounted to the / directory
  - Contains most OS files

Mounting

- `mount` command
    - Used to mount devices to mount point directories
    - With no options or arguments, lists currently mounted filesystems
- `umount` command
    - Used to unmount devices from mount point directories
- On most distributions, mount & umount can only be executed by root

Mounting



```
[root@itmo456 ~]# ll /run/media/sean/Fedora-Live-Desktop-x86_64-20-1/
total 6
dr-xr-xr-x. 3 sean sean 2048 Dec 11  2013 EFI
dr-xr-xr-x. 2 sean sean 2048 Dec 11  2013 isolinux
dr-xr-xr-x. 2 sean sean 2048 Dec 11  2013 LiveOS
[root@itmo456 ~]#
[root@itmo456 ~]# ll /mnt
total 4
drwxr-xr-x. 3 root root 4096 Sep 20 11:34 sysimage
[root@itmo456 ~]#
[root@itmo456 ~]# mount -t cifs -o username=mokena //192.168.2.200/G/Software /mnt

Password for mokena@//192.168.2.200/G/Software:  *********
[root@itmo456 ~]#
[root@itmo456 ~]# ll /mnt/ubuntu-14.04.1-desktop-amd64.iso
-rwxr-xr-x. 1 root root 1028653056 Jul 22  2014 /mnt/ubuntu-14.04.1-desktop-amd64.
iso
[root@itmo456 ~]#
[root@itmo456 ~]# mount /mnt/ubuntu-14.04.1-desktop-amd64.iso /media
mount: /dev/loop0 is write-protected, mounting read-only
[root@itmo456 ~]#
[root@itmo456 ~]# ll /media/
total 2548
-r--r--r--. 1 root root   134 Jul 22  2014 autorun.inf
dr-xr-xr-x. 1 root root  2048 Jul 22  2014 boot
dr-xr-xr-x. 1 root root  2048 Jul 22  2014 casper
dr-xr-xr-x. 1 root root  2048 Jul 22  2014 dists
dr-xr-xr-x. 1 root root  2048 Jul 22  2014 EFI
dr-xr-xr-x. 1 root root  2048 Jul 22  2014 install
dr-xr-xr-x. 1 root root 18432 Jul 22  2014 isolinux
-r--r--r--. 1 root root 21426 Jul 22  2014 md5sum.txt
```

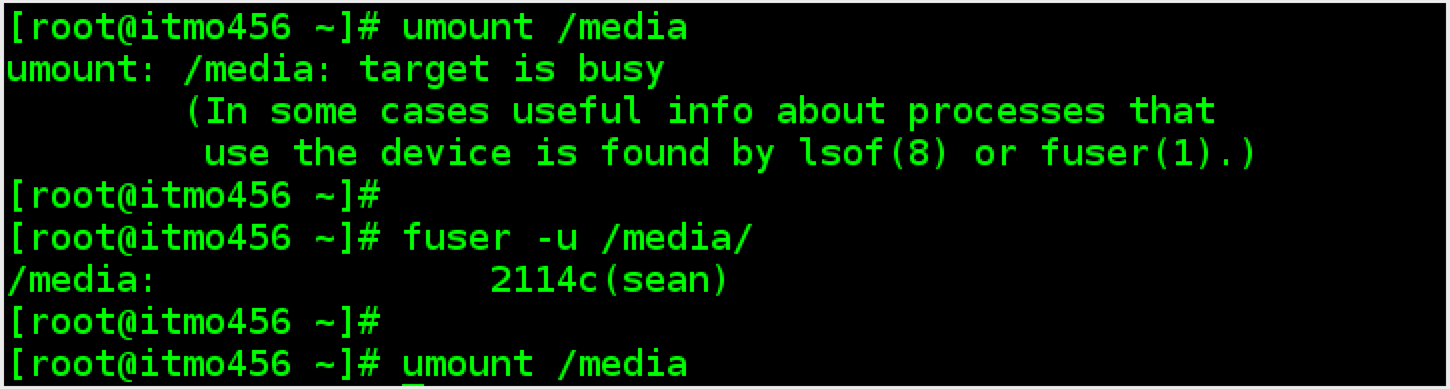
Working with Storage Devices

- When transferring small amounts of information from computer to computer, floppy disk removable media formerly used to store files
  - However, floppy disks had to be prepared before they are used in Linux
  - Each disk device must be formatted with a filesystem prior to being used to store files

Working with Storage Devices

- To mount or unmount floppies, ensure that no user is currently using the mount point directory
- fuser command
  - With the –u option, lists users using a directory

Working with Storage Devices



```
[root@itmo456 ~]# umount /media
umount: /media: target is busy
        (In some cases useful info about processes that
         use the device is found by lsof(8) or fuser(1).)
[root@itmo456 ~]#
[root@itmo456 ~]# fuser -u /media/
/media:               2114c(sean)
[root@itmo456 ~]#
[root@itmo456 ~]# umount /media
```

Working with Storage Devices

- **`mkfs`** (make filesystem) command
  - Used to format a disk with a filesystem
  - **`–t`** option: Specifies filesystem type
    - ext4 default filesystem in Ubuntu & Fedora
    - Also used to create filesystems on hard drive partitions

Working with Storage Devices

- Most (almost all) floppy disks today come preformatted with the DOS FAT filesystem
  - Will work just fine for for Linux when mounted correctly
  - (Work on Macs as well…)
- Flash drives have nearly entirely displaced floppies today
  - Universally preformatted with FAT32

Working with Storage Devices

Command	Description
mkfs	Used to create filesystems of most types
mkfs.msdos **mkdosfs** **mkfs.vfat**	Used to create a DOS FAT filesystem
mkfs.ext2 **mke2fs**	Used to create an ext2 filesystem
mkfs.ext3 **mke2fs -j**	Used to create an ext3 filesystem (j = journaling)
mkfs.ext4 **mke2fs -t ext4**	Used to create an ext4 filesystem
mkisofs	Used to create a CD-ROM filesystem
mkfs.reiserfs **mkreiserfs**	Used to create a REISERFS filesystem

Table 5-3: Commands used to create filesystems

Working with Storage Devices

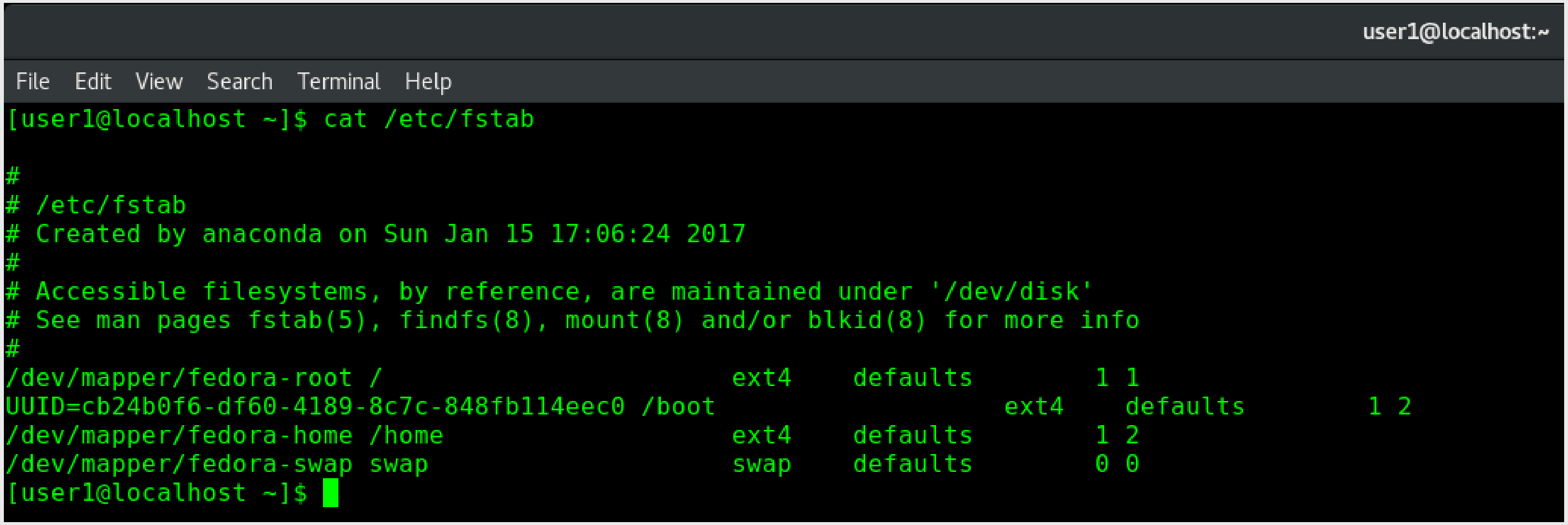
Command	Description
mkfs.xfs	Used to create XFS filesystems
mkntfs **Mkfs.ntfs**	Used to create a Windows NTFS filesystem

Table 5-3: Commands used to create filesystems

Working with Storage Devices

- **`/etc/fstab`** file
  - Used to mount devices at boot time
  - Also consulted when users do not specify enough mount command arguments
  - Six fields: Device to mount, mount point, type, mount options, dump#, fsck#

Working with Storage Devices



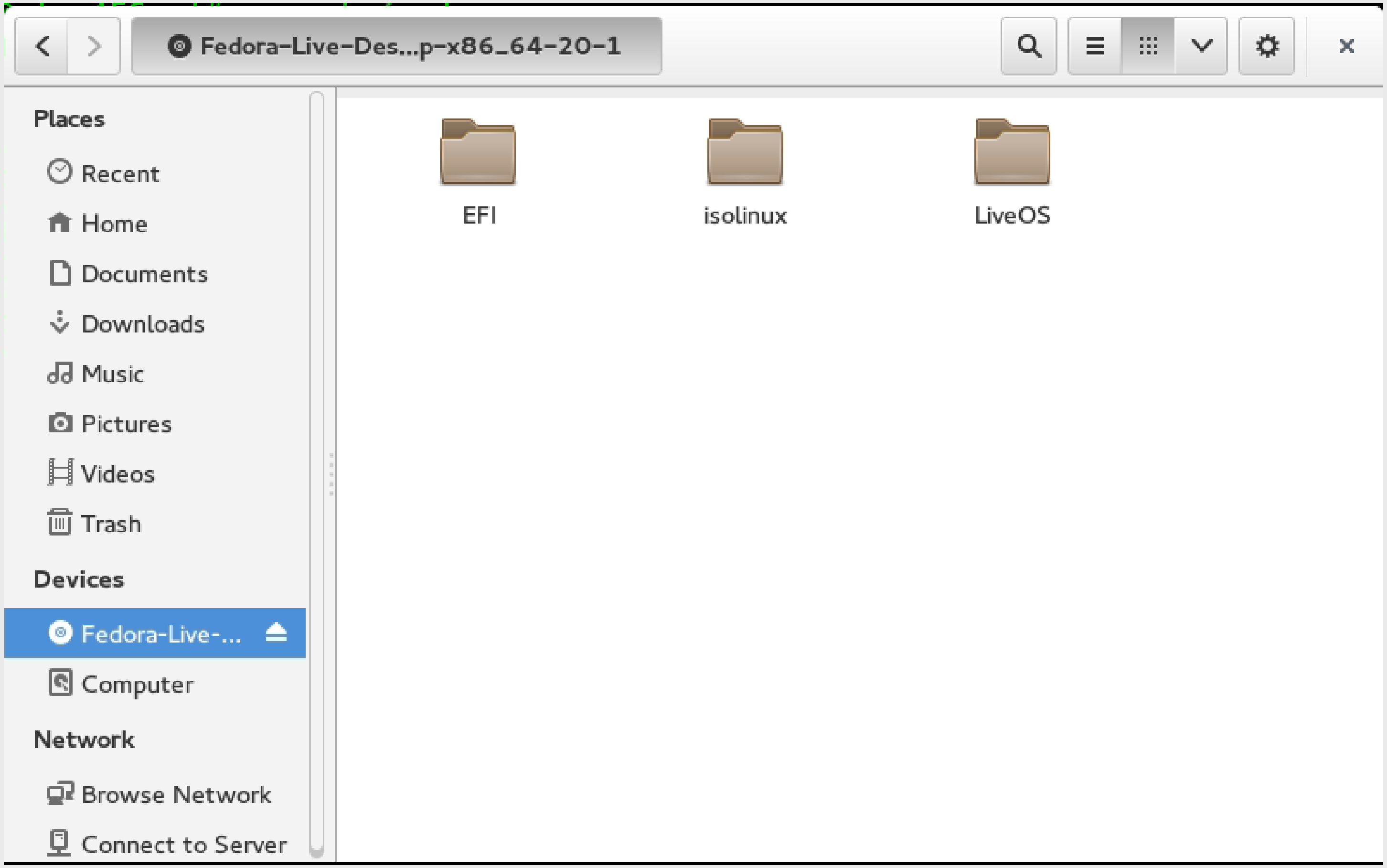
```
[user1@localhost ~]$ cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Sun Jan 15 17:06:24 2017
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/fedora-root /                       ext4    defaults        1 1
UUID=cb24b0f6-df60-4189-8c7c-848fb114eec0 /boot                   ext4    defaults        1 2
/dev/mapper/fedora-home /home                   ext4    defaults        1 2
/dev/mapper/fedora-swap swap                    swap    defaults        0 0
[user1@localhost ~]$ 
```

Working with Storage Devices

Command	Description
mount	Displays mounted filesystems
mount -t <type> <device> <mount point>	Mounts a <device> of a certain <type> to a <mount point> directory
fuser -u <directory>	Displays the users using a particular directory
umount <mount point> ***or*** **umount <device>**	Unmounts a <device> from its <mount point> directory

Useful commands when mounting/unmounting filesystems

Working with Storage Devices



Working with CDs, DVDs & ISOs

- Most software packaged on CDs and DVDs
- Can be mounted using the **`mount`** command and unmounted using **`umount`** command
  - Different device file - depend on the technology used by the drive itself

Working with CDs, DVDs & ISOs

- Older systems may have an ATAPI compliant PATA CD-ROM or DVD-ROM drive
  - Must be configured as one of the following:
    - Primary master (/dev/hda)
    - Primary slave (/dev/hdb)
    - Secondary master (/dev/hdc)
    - Secondary slave (/dev/hdd)

Working with CDs, DVDs & ISOs

- Newer PCs will have a SATA CD-ROM or DVD-ROM drive
  - SATA & SCSI drives will be configured with the next sequential SCSI device letter:
    - Primary HDD (/dev/sda)
    - First partition on primary hard drive (/dev/sda1)
    - Second partition on primary hard drive (/dev/sda2)
    - DVD-ROM (/dev/sr0)

Working with CDs, DVDs & ISOs

- For SATA or SCSI drives, Linux may use many different names, depending on the actual CD or DVD drive
- To make identification of CD/DVD drive easier, Fedora Linux creates a file called /dev/cdrom
  - Is a symbolic link to the correct device file for your first CD or DVD drive

Working with CDs, DVDs & ISOs

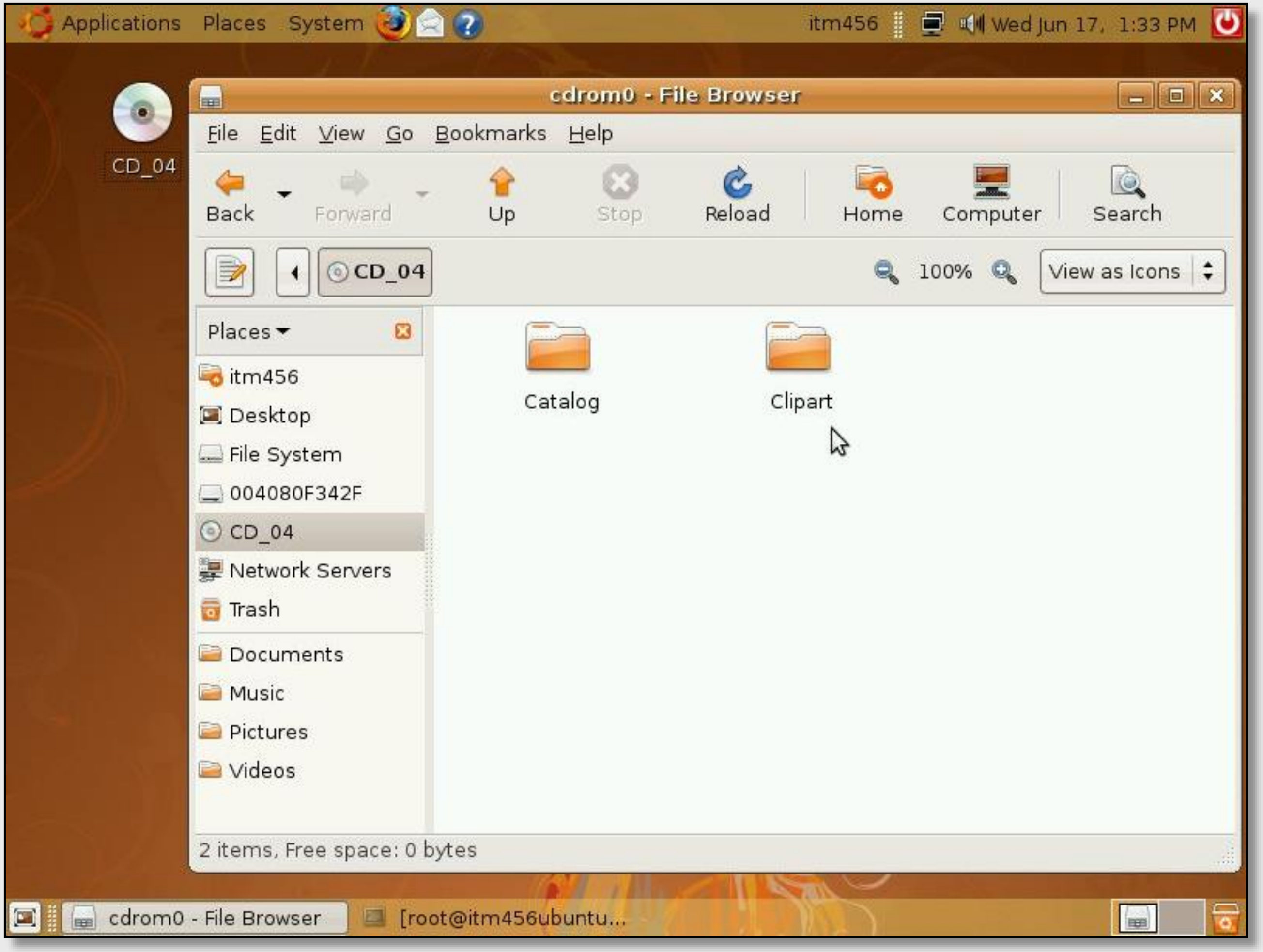
- CDs and DVDs typically use iso9660 or UDF filesystem type and are read only
  - Mount with `–r` (read-only) option
  - Cannot be ejected until properly unmounted
- In GUI environment, CD/DVD is often automatically mounted to a directory underneath the **`/run/media/`*`username`*** directory
  - Named for the label on the CD or DVD
  - System will place shortcut on desktop

Working with CDs, DVDs & ISOs

- Current Linux distributions will automatically unmount CD or DVD disc when “eject” is selected from the GUI menu

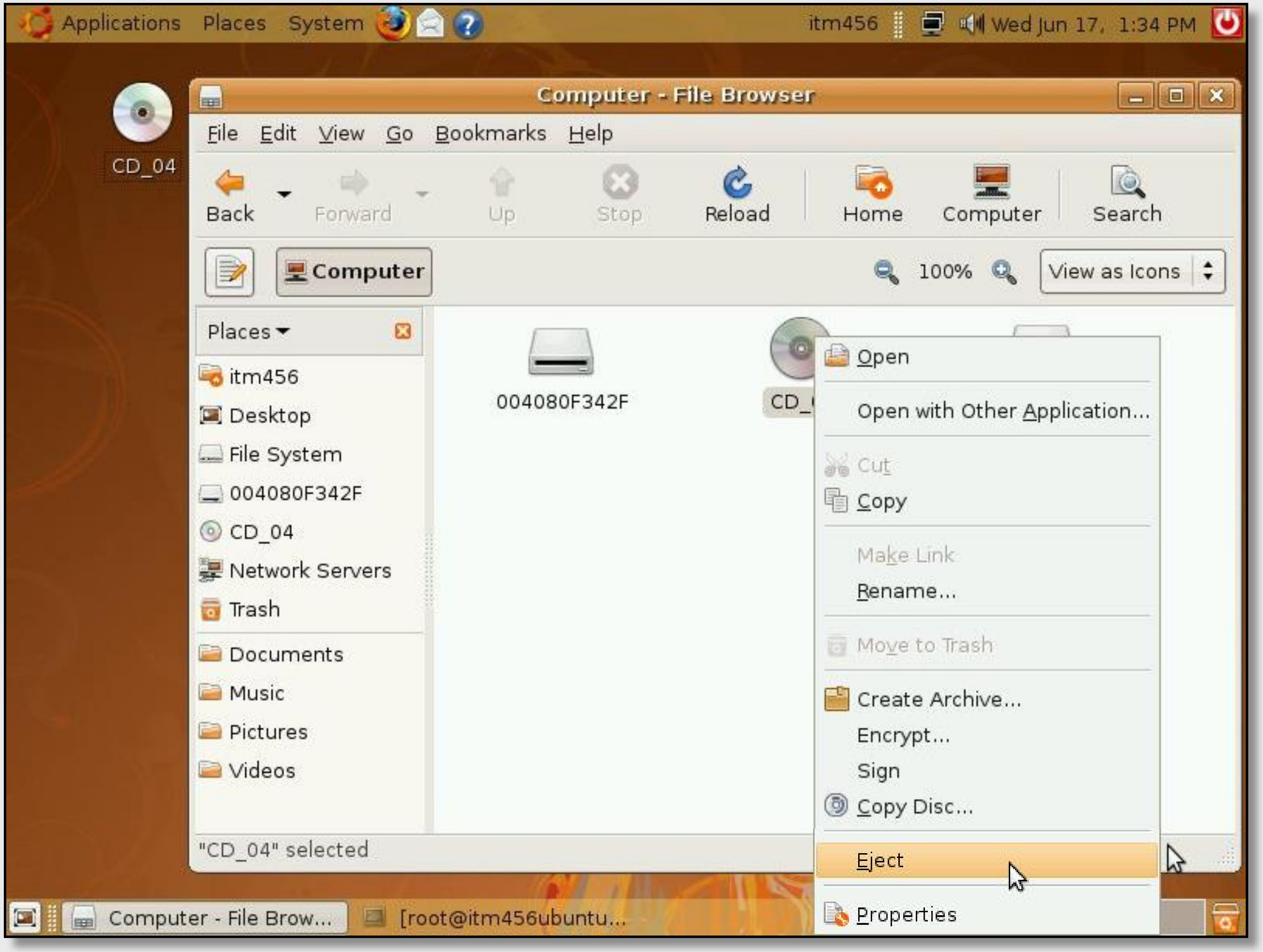
Working with CDs, DVDs & ISOs

Figure 5-3: Accessing a DVD within the GNOME desktop environment



Working with CDs, DVDs & ISOs

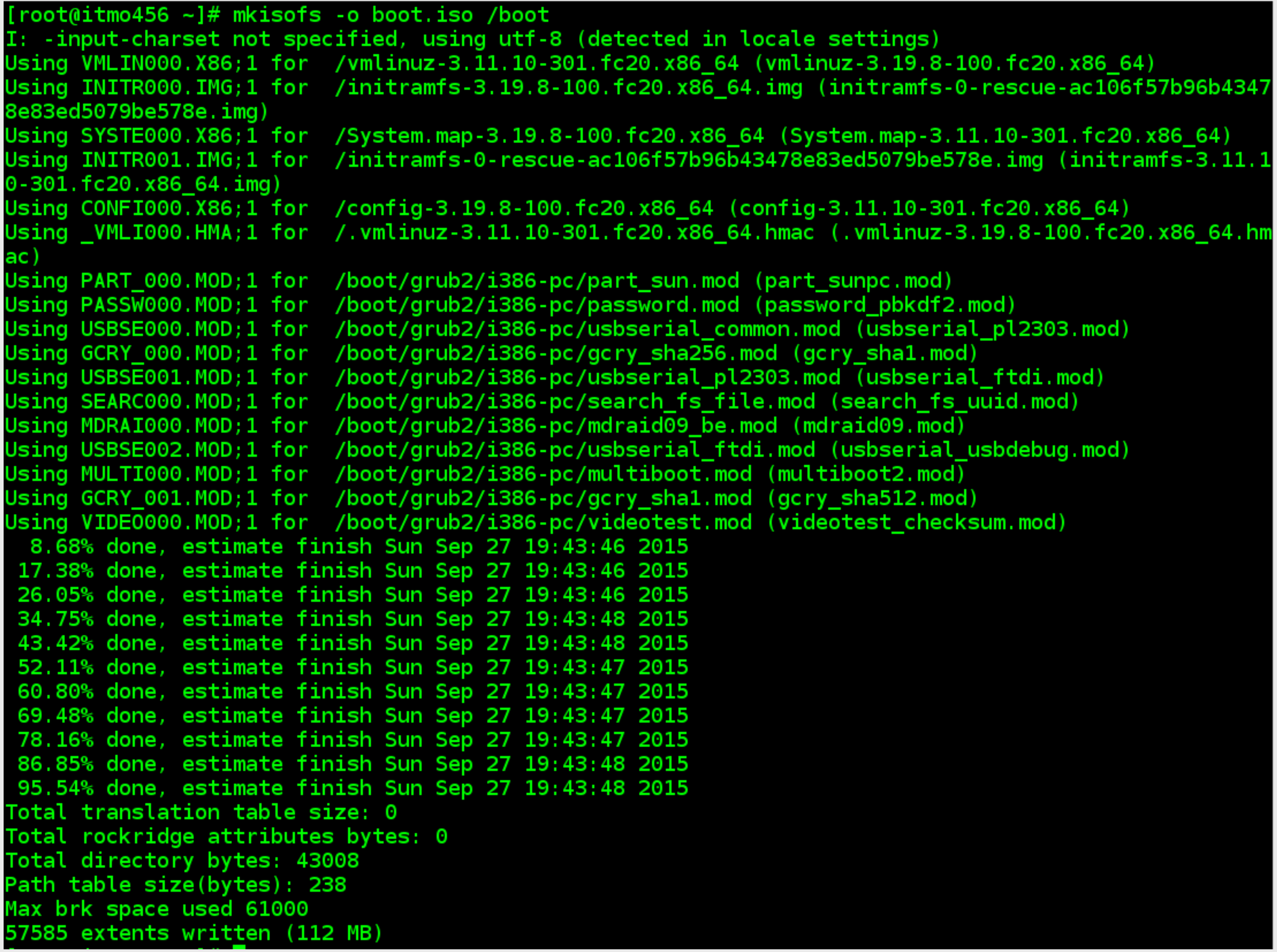
Unmounting a CD-ROM device in a GUI environment



Working with CDs, DVDs & ISOs

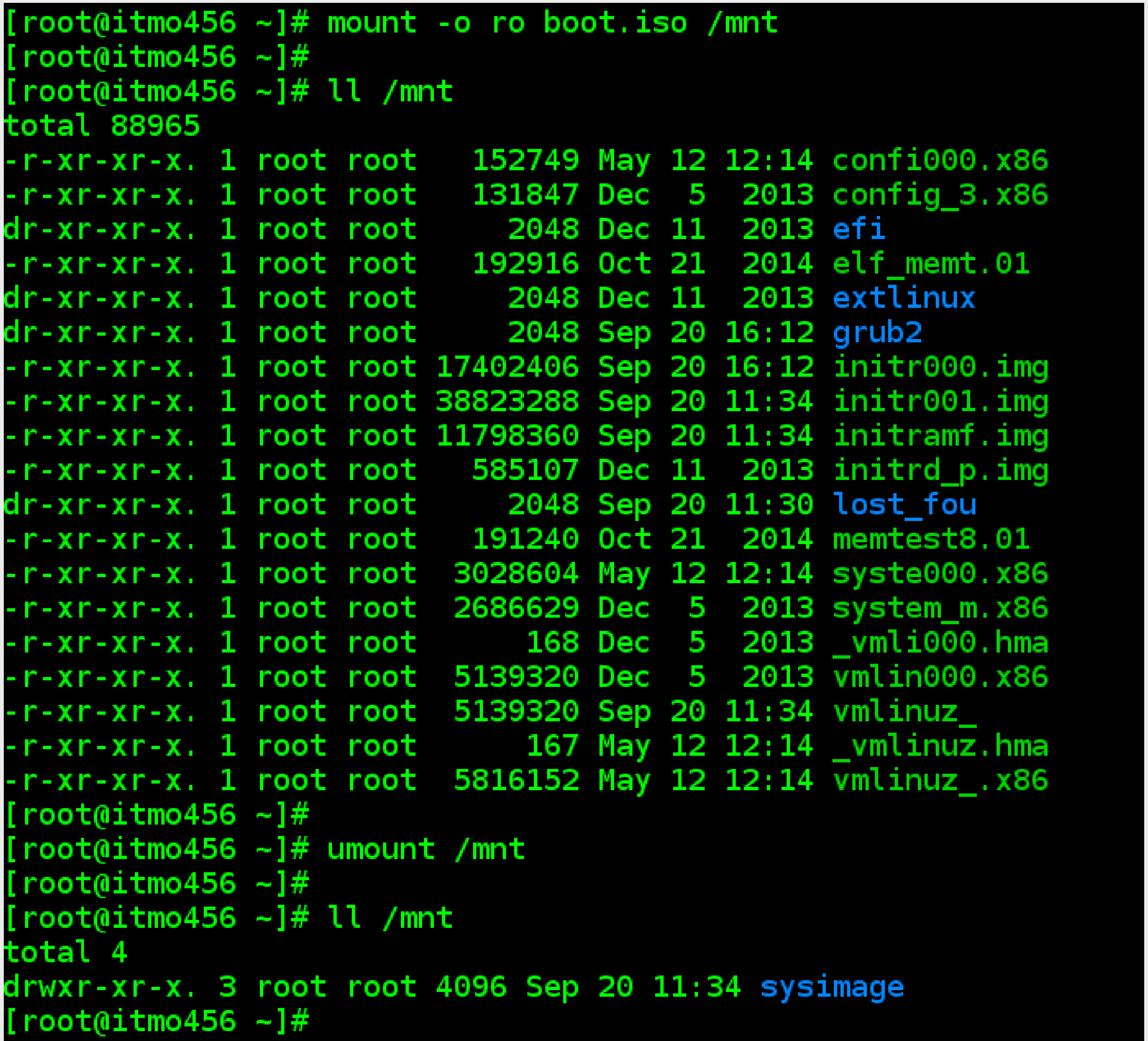
- The iso9660 filesystem can be used to create ISO images that contain other files
  - Can be mounted as a loopback device using the **`mount`** command
- **`mkisofs`** command
  - Creates an ISO image from a directory
  - Receives at least two arguments:
    - Filename to be created
    - Directory used to create the ISO image

Working with CDs, DVDs & ISOs



```
[root@itmo456 ~]# mkisofs -o boot.iso /boot
I: -input-charset not specified, using utf-8 (detected in locale settings)
Using VMLIN000.X86;1 for  /vmlinuz-3.11.10-301.fc20.x86_64 (vmlinuz-3.19.8-100.fc20.x86_64)
Using INITR000.IMG;1 for  /initramfs-3.19.8-100.fc20.x86_64.img (initramfs-0-rescue-ac106f57b96b4347
8e83ed5079be578e.img)
Using SYSTE000.X86;1 for  /System.map-3.19.8-100.fc20.x86_64 (System.map-3.11.10-301.fc20.x86_64)
Using INITR001.IMG;1 for  /initramfs-0-rescue-ac106f57b96b43478e83ed5079be578e.img (initramfs-3.11.1
0-301.fc20.x86_64.img)
Using CONFI000.X86;1 for  /config-3.19.8-100.fc20.x86_64 (config-3.11.10-301.fc20.x86_64)
Using _VMLI000.HMA;1 for  /.vmlinuz-3.11.10-301.fc20.x86_64.hmac (.vmlinuz-3.19.8-100.fc20.x86_64.hm
ac)
Using PART_000.MOD;1 for  /boot/grub2/i386-pc/part_sun.mod (part_sunpc.mod)
Using PASSW000.MOD;1 for  /boot/grub2/i386-pc/password.mod (password_pbkdf2.mod)
Using USBSE000.MOD;1 for  /boot/grub2/i386-pc/usbserial_common.mod (usbserial_pl2303.mod)
Using GCRY_000.MOD;1 for  /boot/grub2/i386-pc/gcry_sha256.mod (gcry_sha1.mod)
Using USBSE001.MOD;1 for  /boot/grub2/i386-pc/usbserial_pl2303.mod (usbserial_ftdi.mod)
Using SEARC000.MOD;1 for  /boot/grub2/i386-pc/search_fs_file.mod (search_fs_uuid.mod)
Using MDRAI000.MOD;1 for  /boot/grub2/i386-pc/mdraid09_be.mod (mdraid09.mod)
Using USBSE002.MOD;1 for  /boot/grub2/i386-pc/usbserial_ftdi.mod (usbserial_usbdebug.mod)
Using MULTI000.MOD;1 for  /boot/grub2/i386-pc/multiboot.mod (multiboot2.mod)
Using GCRY_001.MOD;1 for  /boot/grub2/i386-pc/gcry_sha1.mod (gcry_sha512.mod)
Using VIDEO000.MOD;1 for  /boot/grub2/i386-pc/videotest.mod (videotest_checksum.mod)
  8.68% done, estimate finish Sun Sep 27 19:43:46 2015
 17.38% done, estimate finish Sun Sep 27 19:43:46 2015
 26.05% done, estimate finish Sun Sep 27 19:43:46 2015
 34.75% done, estimate finish Sun Sep 27 19:43:48 2015
 43.42% done, estimate finish Sun Sep 27 19:43:48 2015
 52.11% done, estimate finish Sun Sep 27 19:43:47 2015
 60.80% done, estimate finish Sun Sep 27 19:43:47 2015
 69.48% done, estimate finish Sun Sep 27 19:43:47 2015
 78.16% done, estimate finish Sun Sep 27 19:43:47 2015
 86.85% done, estimate finish Sun Sep 27 19:43:48 2015
 95.54% done, estimate finish Sun Sep 27 19:43:48 2015
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 43008
Path table size(bytes): 238
Max brk space used 61000
57585 extents written (112 MB)
```

Working with CDs, DVDs & ISOs



```
[root@itmo456 ~]# mount -o ro boot.iso /mnt
[root@itmo456 ~]#
[root@itmo456 ~]# ll /mnt
total 88965
-r-xr-xr-x. 1 root root   152749 May 12 12:14 confi000.x86
-r-xr-xr-x. 1 root root   131847 Dec  5  2013 config_3.x86
dr-xr-xr-x. 1 root root     2048 Dec 11  2013 efi
-r-xr-xr-x. 1 root root   192916 Oct 21  2014 elf_memt.01
dr-xr-xr-x. 1 root root     2048 Dec 11  2013 extlinux
dr-xr-xr-x. 1 root root     2048 Sep 20 16:12 grub2
-r-xr-xr-x. 1 root root 17402406 Sep 20 16:12 initr000.img
-r-xr-xr-x. 1 root root 38823288 Sep 20 11:34 initr001.img
-r-xr-xr-x. 1 root root 11798360 Sep 20 11:34 initramf.img
-r-xr-xr-x. 1 root root   585107 Dec 11  2013 initrd_p.img
dr-xr-xr-x. 1 root root     2048 Sep 20 11:30 lost_fou
-r-xr-xr-x. 1 root root   191240 Oct 21  2014 memtest8.01
-r-xr-xr-x. 1 root root  3028604 May 12 12:14 syste000.x86
-r-xr-xr-x. 1 root root  2686629 Dec  5  2013 system_m.x86
-r-xr-xr-x. 1 root root      168 Dec  5  2013 _vmli000.hma
-r-xr-xr-x. 1 root root  5139320 Dec  5  2013 vmlin000.x86
-r-xr-xr-x. 1 root root  5139320 Sep 20 11:34 vmlinuz_
-r-xr-xr-x. 1 root root      167 May 12 12:14 _vmlinuz.hma
-r-xr-xr-x. 1 root root  5816152 May 12 12:14 vmlinuz_.x86
[root@itmo456 ~]#
[root@itmo456 ~]# umount /mnt
[root@itmo456 ~]#
[root@itmo456 ~]# ll /mnt
total 4
drwxr-xr-x. 3 root root 4096 Sep 20 11:34 sysimage
[root@itmo456 ~]#
```

Working with Hard Disks

- Three types of hard disks: PATA, SATA, and SCSI
- PATA HDDs must be configured in one of the following:
  - Primary master (/dev/hda)
  - Primary slave (/dev/hdb)
  - Secondary master (/dev/hdc)
  - Secondary slave (/dev/hdd)
- Different device file for each

Working with Hard Disks

- SATA and SCSI hard disks are well-suited to Linux servers
  - Faster access speed
  - Multiple hard drives can be attached to a controller
- Associated with different device files
  - First SCSI/SATA HDD (/dev/sda)
  - Second SCSI/SATA HDD (/dev/sdb)
  - Third SCSI/SATA HDD (/dev/sdc)
  - And so on

Working with Hard Disks

- SSDs (Solid State Drives) are large-capacity flash drives built to mount and appear to the OS as SATA or SCSI hard disks
  - Treated by the OS as a regular SATA or SCSI drive

Standard Hard Disk Partitioning

- As disk size increases, organization becomes more difficult and chance of error increases
- Partition
  - Physical division of an HDD
  - Can have its own filesystem
- Linux requires at least two partitions; root and swap

Standard Hard Disk Partitioning

- Good practice to use more than just two partitions
  - Segregates different types of data
  - Allows for use of multiple filesystem types on one HDD
  - Reduces chance that filesystem corruption will render a system unusable
  - Speeds up access to stored data by keeping filesystems as small as possible

Standard Hard Disk Partitioning

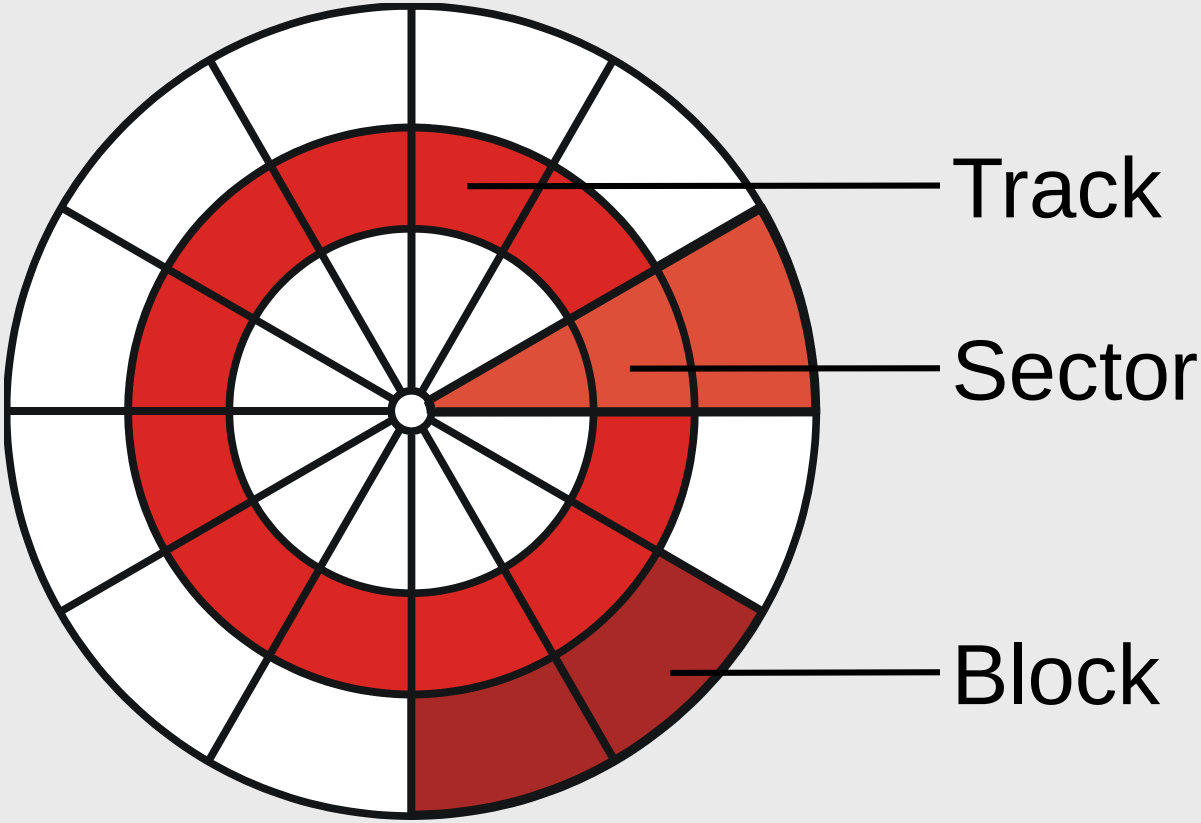
- Tracks
  - Area on a hard disk that form a concentric circle of sectors
- Sector
  - Smallest unit of data storage on a hard disk
- Block
  - Sectors of information that are combined

Standard Hard Disk Partitioning

- Cylinder
  - Series of tracks on a hard disk that are written to simultaneously by the magnetic heads in a hard disk drive
- Partition definitions stored in first readable sector of the hard disk
  - Master Boot Record (MBR) or master boot block (MBB)

Standard Hard Disk Partitioning

Figure 5-4: The physical areas of a hard drive



Standard Hard Disk Partitioning

Partition	PATA Device Name (assuming /dev/hda)	SCSI/SAS/SATA Device Name (assuming /dev/sda)
1st primary partition	/dev/hda1	/dev/sda1
2nd primary partition	/dev/hda2	/dev/sda2
3rd primary partition	/dev/hda3	/dev/sda3
4th primary partition	/dev/hda4	/dev/sda4
1st logical drive in the extended partition	/dev/hda5	/dev/sda5
2nd logical drive in the extended partition	/dev/hda6	/dev/sda6
3rd logical drive in the extended partition	/dev/hda7	/dev/sda7
4th logical drive in the extended partition	/dev/hda8	/dev/sda8
5th logical drive in the extended partition	/dev/hda9	/dev/sda9
nth logical drive in the extended partition	/dev/hdan	/dev/sdan

Table 5-5: Common hard disk partition device files for /dev/hda and /dev/sda

Standard Hard Disk Partitioning

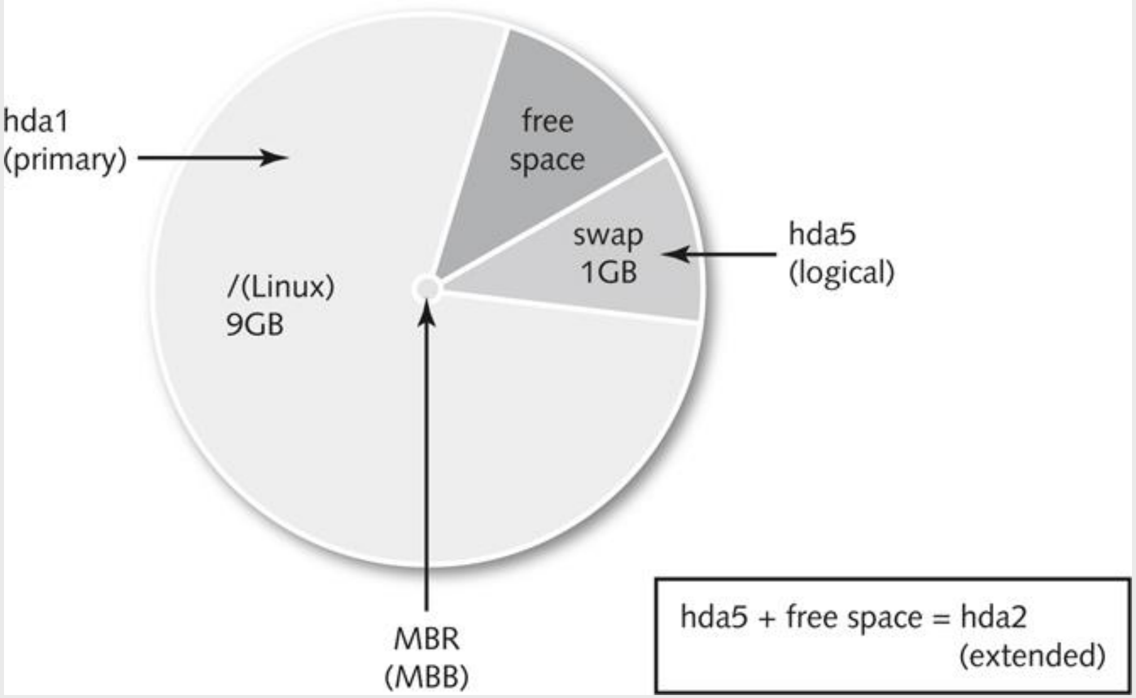


Figure 5-5: A sample Linux partitioning strategy

Standard Hard Disk Partitioning

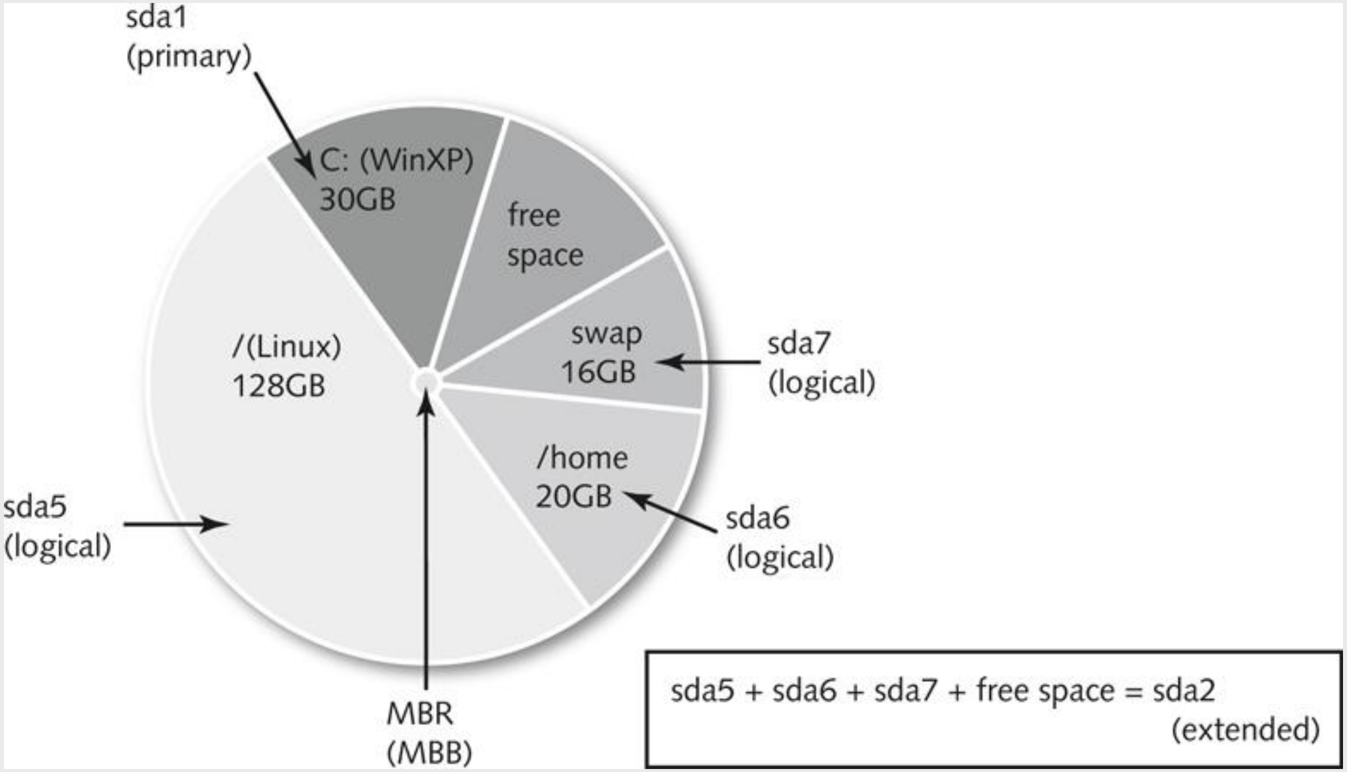


Figure 5-6: A sample dual-boot Linux partitioning strategy

Working with Standard Hard Disk Partitions

- **`fdisk`** command
  - Create partitions after installation
  - Specify hard disk partition as an argument
  - Variety of options for **`fdisk`** prompt to achieve different tasks
- **`cfdisk`** command
  - Interactive graphical utility for creating, manipulating and deleting partitions

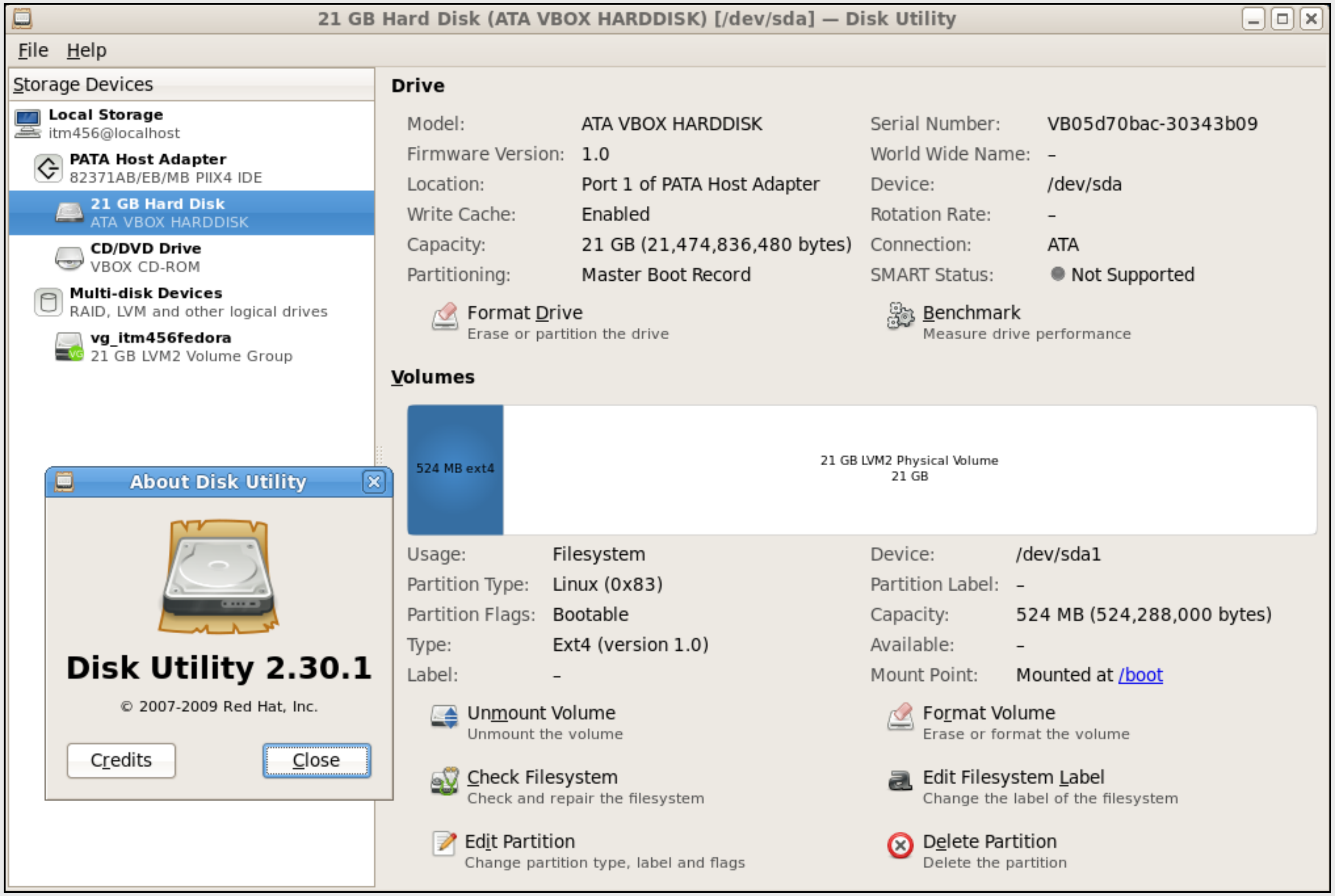
Working with Standard Hard Disk Partitions

- Reboot computer after using the **`fdisk`** and **`cfdisk`** commands to ensure proper reloading into memory
- Red Hat’s Disk Utility provides a GUI for partition management
- Edit **`/etc/fstab`** file to allow system to mount new filesystems automatically at boot time

Working with Standard Hard Disk Partitions

- `mkswap` command
  - Prepare the swap partition
- `swapon` command
  - Activate the swap partition
- `swapoff` command
  - Deactivate the swap partition
- Edit `/etc/fstab` file to ensure new swap partition is activated as virtual memory

Working with Hard Disk Partitions



21 GB Hard Disk (ATA VBOX HARDDISK) [/dev/sda] — Disk Utility

Drive

Model:	ATA VBOX HARDDISK	Serial Number:	VB05d70bac-30343b09
Firmware Version:	1.0	World Wide Name:	–
Location:	Port 1 of PATA Host Adapter	Device:	/dev/sda
Write Cache:	Enabled	Rotation Rate:	–
Capacity:	21 GB (21,474,836,480 bytes)	Connection:	ATA
Partitioning:	Master Boot Record	SMART Status:	● Not Supported

Volumes

Usage:	Filesystem	Device:	/dev/sda1
Partition Type:	Linux (0x83)	Partition Label:	–
Partition Flags:	Bootable	Capacity:	524 MB (524,288,000 bytes)
Type:	Ext4 (version 1.0)	Available:	–
Label:	–	Mount Point:	Mounted at /boot

About Disk Utility

Disk Utility 2.30.1

© 2007-2009 Red Hat, Inc.

Using the Red Hat Disk Utility for hard drive partitioning

(In Ubuntu as well as Fedora; also supports some LVM management)

Working with Hard Disk Partitions

- If your hard disk uses a GPT instead of a MBR
  - You can’t use the **`fdisk`** or **`cfdisk`** commands to create and modify partitions before you format them with a filesystem or prepare them for use as swap memory
- Use the **`gdisk`** (GPT fdisk) command
  - To create and work with partitions on GPT hard disk
- **`parted`** (GNU Parted) command
  - Can be used to create and modify partitions on both MBR and GPT hard disks

Working with Hard Disk Partitions

- After creating GPT partitions that should contain a filesystem:
  - Format those partitions with a filesystem using **`mkfs`**
  - Mount them to the directory tree with the `mount` command
  - Update the **`/etc/fstab`** file to mount them automatically

Logical Volumes

- Current operating systems place filesystems on logical volumes (LV) instead of partitions
  - A logical volume is an abstraction of a disk partition provided by a Volume Manager
  - Logical volume normally contained in a partition
  - Conceptually similar to logical partitions in an extended partition

Logical Volumes

- LVs appear as regular block devices which can hold filesystems or swap
- Volume group breaks up storage such as a disk partition into a pool of *physical extents*
  - Extents are then re-assembled by the Volume Manager into logical volumes
- LVs can be managed from the Fedora command line using the logical volume manager **`lvm`**

Working with the LVM

- Logical Volume Manager (LVM)
  - Used to create volumes
  - Volumes can contain filesystems and can be mounted to directories
  - More flexible than standard partitions – allows use of free space across multiple hard disks
  - Has error correction abilities
- LVM components
  - Physical volumes (PVs), volume group (VG), and logical volumes (LVs)

Logical Volumes

- ◆ **Physical volume (PV)** - unused partitions on hard disks that the LVM can use to store information
- ◆ **Volume Group (VG)** – Contains one or more PVs. Represents pool of hard disk storage space to the LVM
- ◆ **Logical volume (LV) –** usable volumes that are created by the LVM from available storage in a VG

Working with the LVM

PV	Size
/dev/hda1	200GB
/dev/sda1	300GB
/dev/sda4	100GB
/dev/sdb2	400GB

VG Total	1000GB

LV Name	LV Size
/var/nsm/	350GB
/home/	250GB
/storage	300GB
Free space	100GB

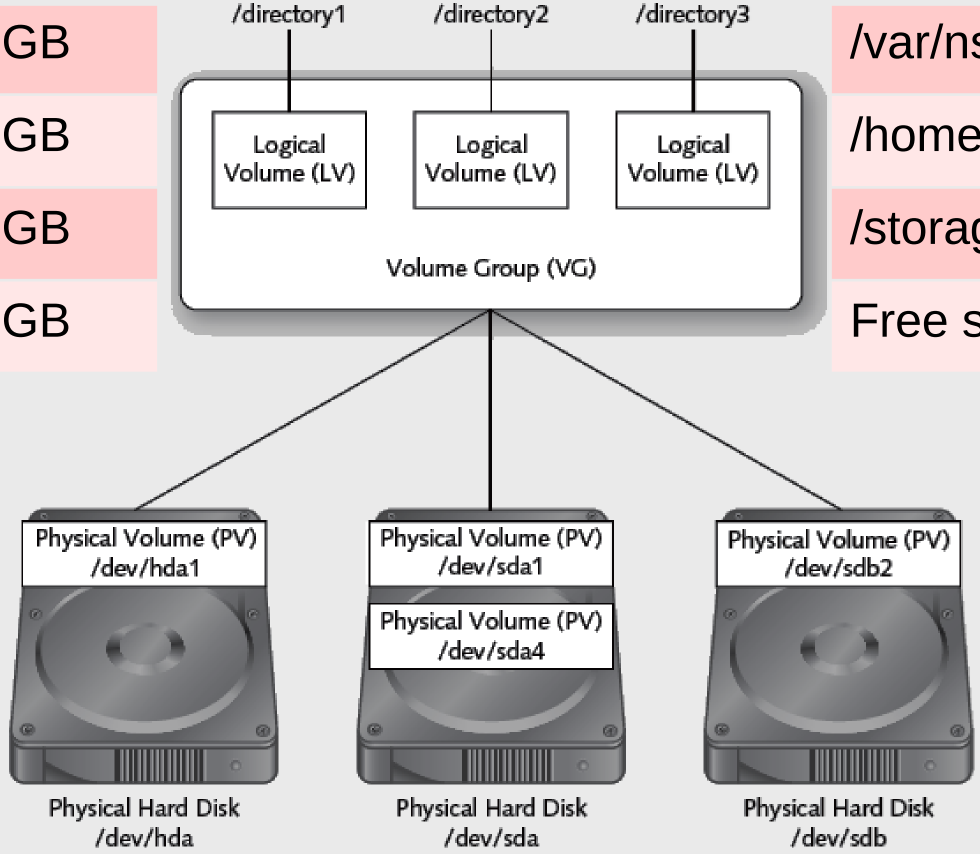

/directory1
/directory2
/directory3
Logical Volume (LV)
Logical Volume (LV)
Logical Volume (LV)
Volume Group (VG)
Physical Volume (PV) /dev/hda1
Physical Volume (PV) /dev/sda1
Physical Volume (PV) /dev/sda4
Physical Volume (PV) /dev/sdb2
Physical Hard Disk /dev/hda
Physical Hard Disk /dev/sda
Physical Hard Disk /dev/sdb

Figure 5-8: A sample LVM configuration

Working with the LVM

- **`pvcreate`** command
  - Create PVs
- **`pvdisplay`** command
  - Display detailed information about each PV
- **`vgcreate`** command
  - Create a VG that uses the space in PVs
  - Arguments are name of the VG and PVs to be used

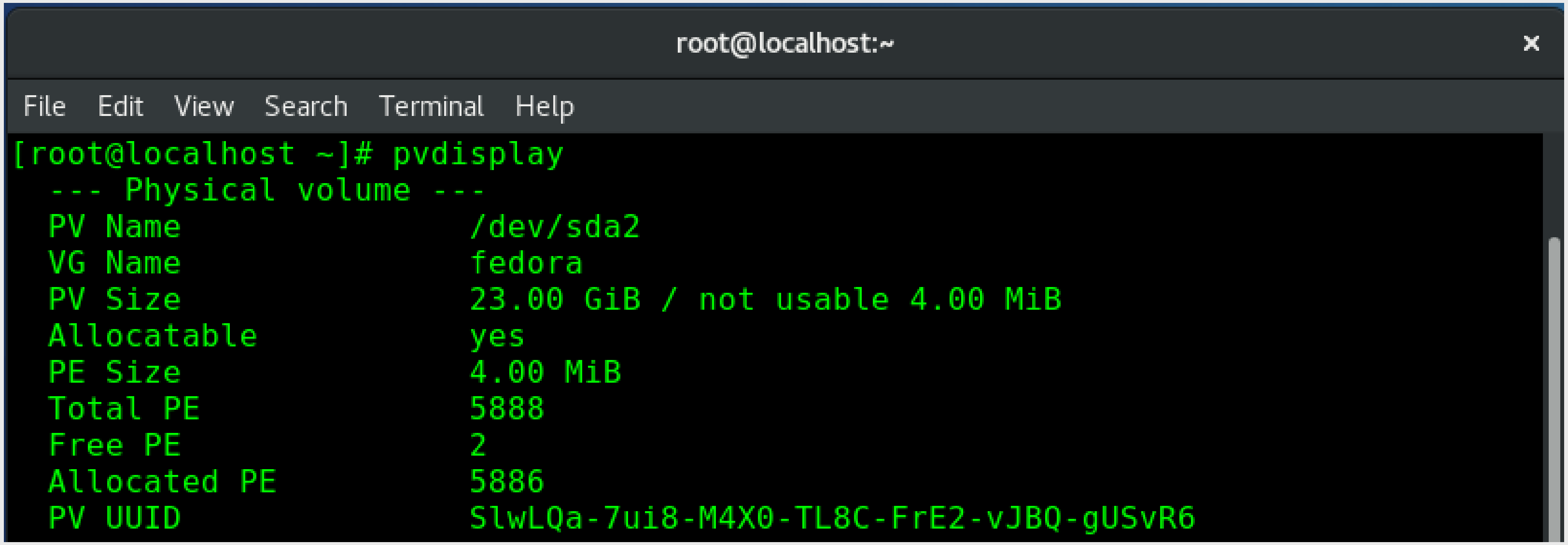
Working with the LVM

- Physical Extent
  - Block size for saving data in a VG
  - Should be set when creating a VG
  - Can use **`vgcreate -s`** to set the PE
- **`vgdisplay`** command
  - Display detailed information about each VG

Working with the LVM

- **`lvcreate`** command
  - Create LVs from available space in a VG
- **`lvdisplay`** command
  - Display information about each LV
- Work with mount points of LVs as you would work with any other hard disk partition device file
  - Edit **`/etc/fstab`** to ensure LVs are automatically mounted at system startup

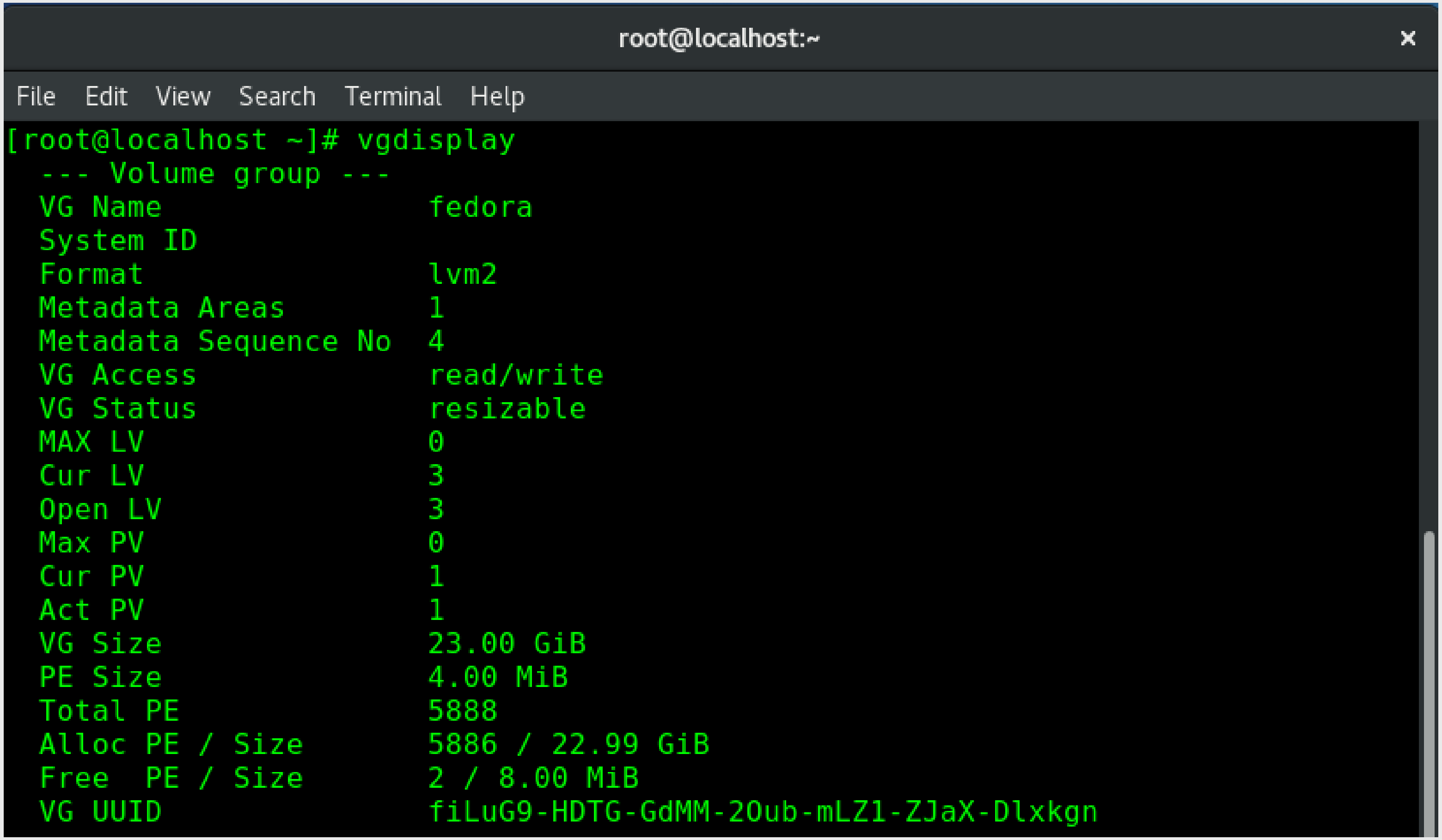
Working with the LVM



```
[root@localhost ~]# pvdisplay
  --- Physical volume ---
  PV Name               /dev/sda2
  VG Name               fedora
  PV Size               23.00 GiB / not usable 4.00 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              5888
  Free PE               2
  Allocated PE          5886
  PV UUID               SlwLQa-7ui8-M4X0-TL8C-FrE2-vJBQ-gUSvR6
```

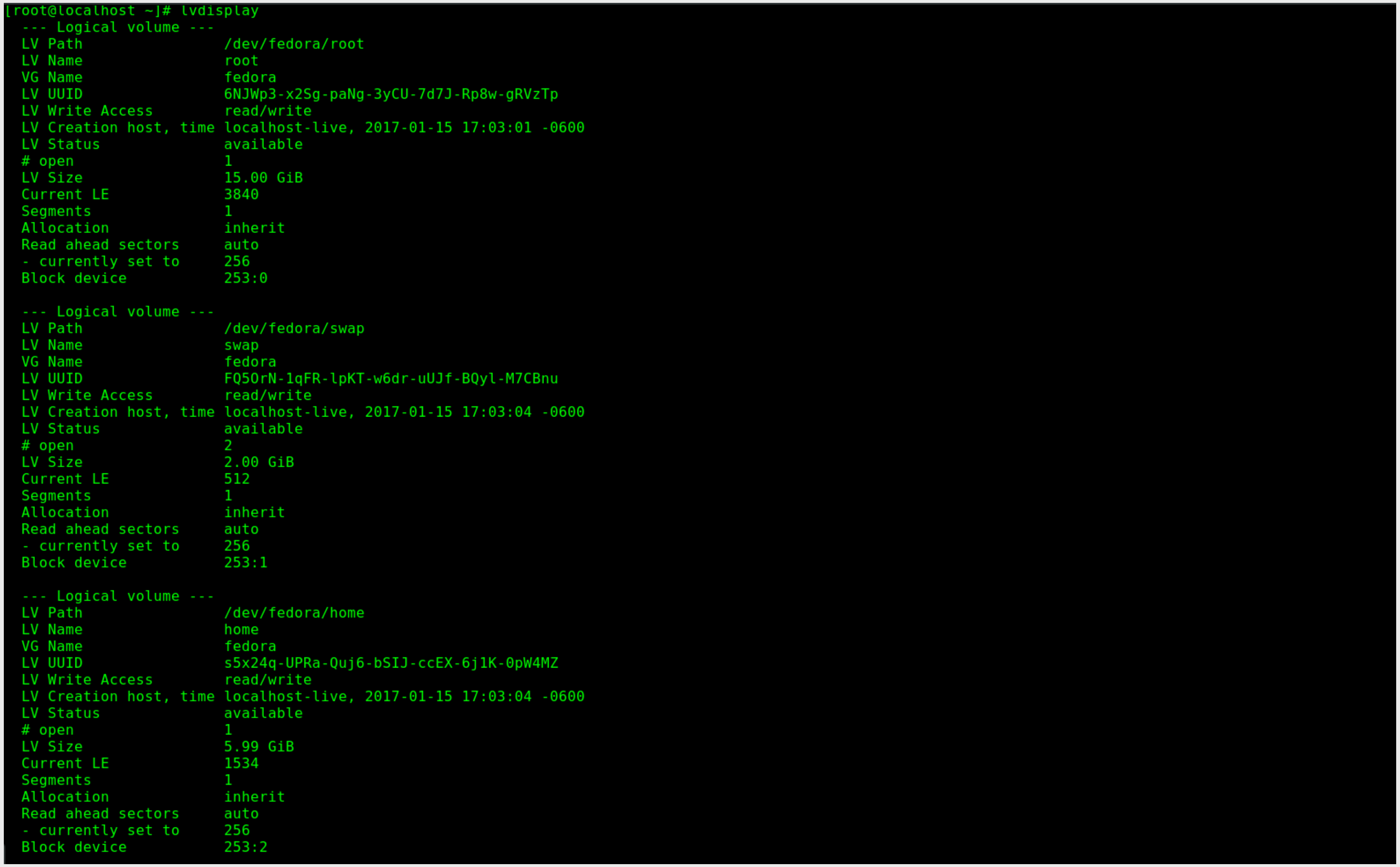
LINUX & OPEN SOURCE

Working with the LVM



```
[root@localhost ~]# vgdisplay
  --- Volume group ---
  VG Name               fedora
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  4
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                3
  Open LV               3
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               23.00 GiB
  PE Size               4.00 MiB
  Total PE              5888
  Alloc PE / Size       5886 / 22.99 GiB
  Free  PE / Size       2 / 8.00 MiB
  VG UUID               fiLuG9-HDTG-GdMM-2Oub-mLZ1-ZJaX-Dlxkgn
```

Working with the LVM



```
[root@localhost ~]# lvdisplay
  --- Logical volume ---
  LV Path                /dev/fedora/root
  LV Name                root
  VG Name                fedora
  LV UUID                6NJWp3-x2Sg-paNg-3yCU-7d7J-Rp8w-gRVzTp
  LV Write Access        read/write
  LV Creation host, time localhost-live, 2017-01-15 17:03:01 -0600
  LV Status              available
  # open                 1
  LV Size                15.00 GiB
  Current LE             3840
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0

  --- Logical volume ---
  LV Path                /dev/fedora/swap
  LV Name                swap
  VG Name                fedora
  LV UUID                FQ5OrN-1qFR-lpKT-w6dr-uUJf-BQyl-M7CBnu
  LV Write Access        read/write
  LV Creation host, time localhost-live, 2017-01-15 17:03:04 -0600
  LV Status              available
  # open                 2
  LV Size                2.00 GiB
  Current LE             512
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:1

  --- Logical volume ---
  LV Path                /dev/fedora/home
  LV Name                home
  VG Name                fedora
  LV UUID                s5x24q-UPRa-Quj6-bSIJ-ccEX-6j1K-0pW4MZ
  LV Write Access        read/write
  LV Creation host, time localhost-live, 2017-01-15 17:03:04 -0600
  LV Status              available
  # open                 1
  LV Size                5.99 GiB
  Current LE             1534
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:2
```

Working with the LVM

- **`pvscan`**, **`vgscan`**, and **`lvscan`** commands
  - Display information about PVs, VGs, and LVs, respectively
- **`vgextend`** command
  - Add a new PV to an existing VG
- **`lvextend`** command
  - Increase the size of an LV, e.g., to use space extended onto an existing VG

Working with USB/Firewire Devices

- Most removable storage devices emulate SCSI protocol in the device firmware
- Devices normally will automatically mount to a new directory under the **`/run/media`** directory named for the label on the device

Monitoring Filesystems

- Mounted filesystems should be checked periodically
  - Errors
  - Disk Space usage
  - Inode usage
- Minimizes problems due to damaged filesystems
  - Reduces likelihood that a file cannot be saved due to insufficient disk space

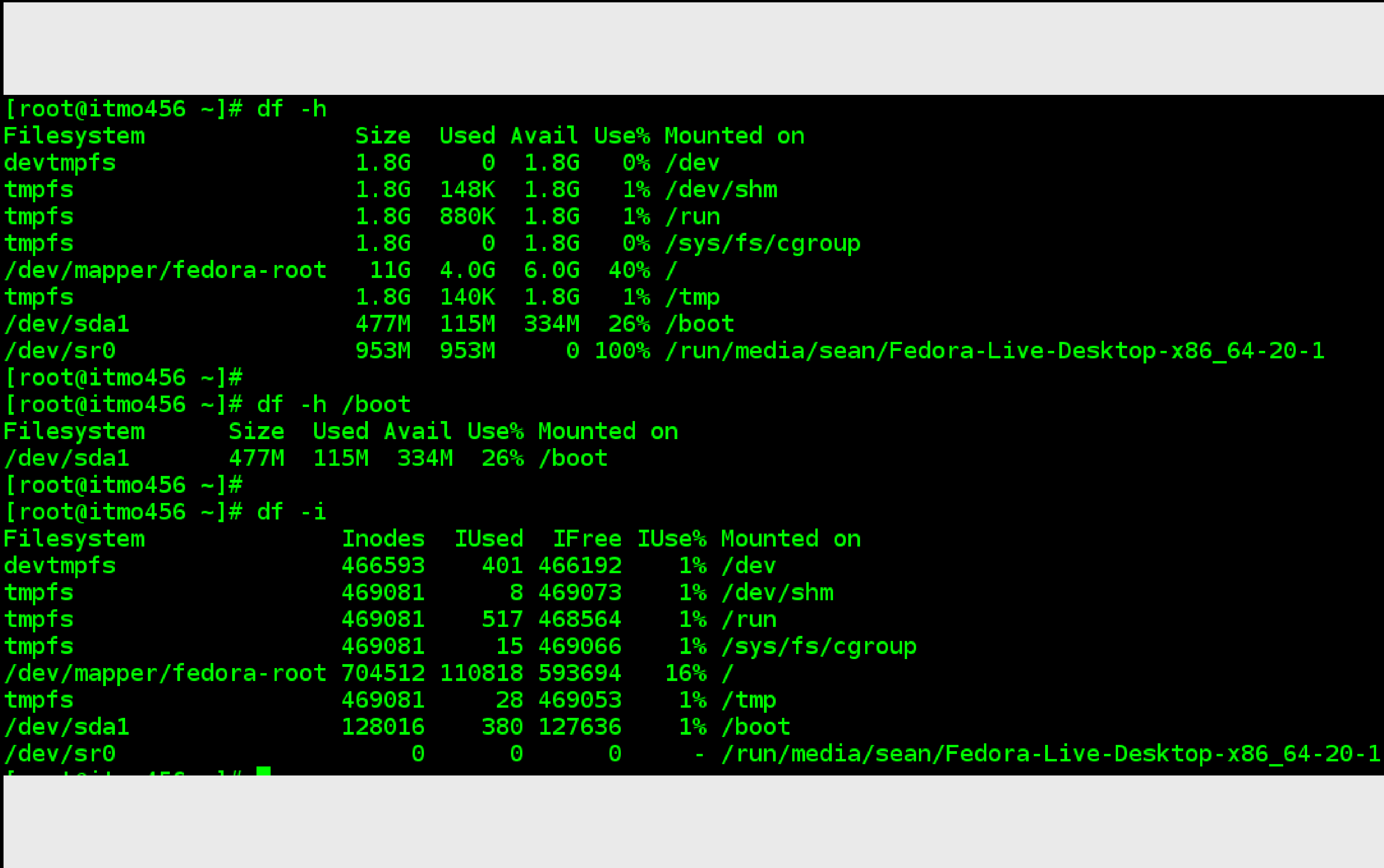
Disk Usage

- Using more filesystems typically results in less hard disk space per filesystem
  - Errors when filesystems fill up with data
    - e.g., free space on / filesystem falls below 10%
  - Periodically remove obsolete files such as old log files to make room for new ones

Disk Usage

- **`df`** (disk free space) command
  - Monitor free space used by mounted filesystems
  - **–h** option: More user friendly
  - To get information about different filesystems, you must mount them prior to using **`df`** command

Disk Usage

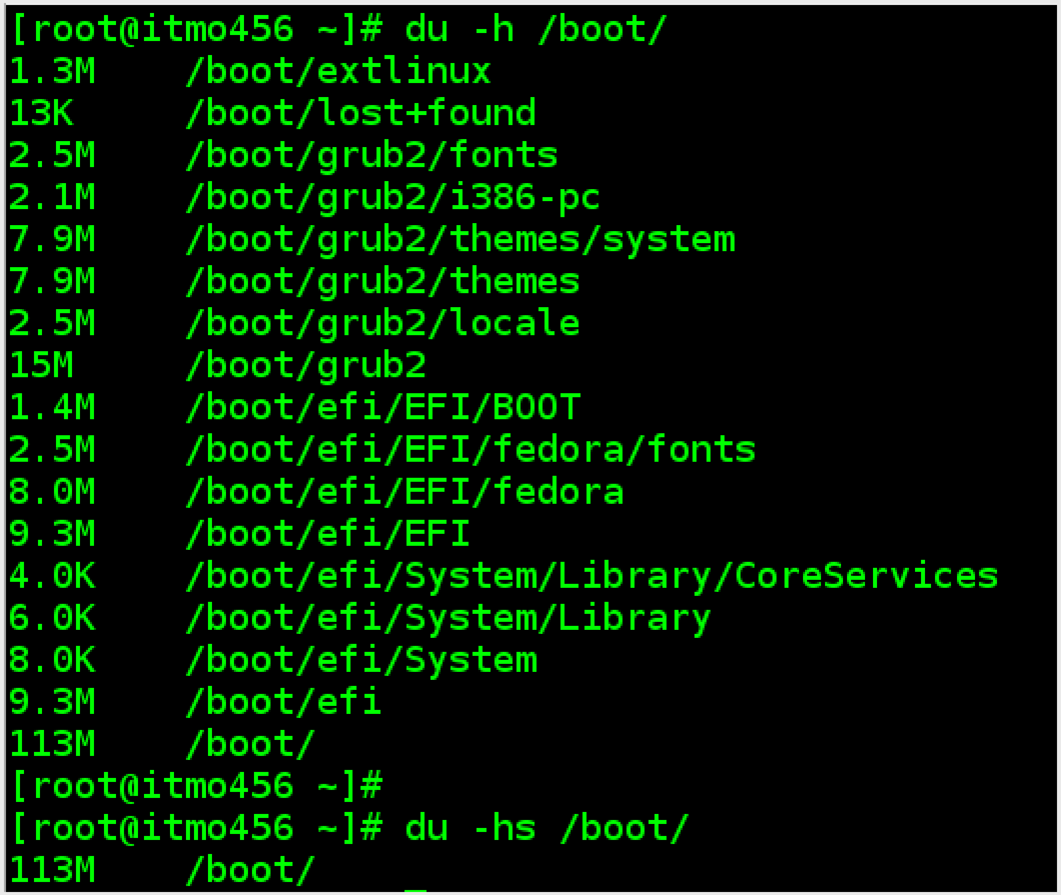


```
[root@itmo456 ~]# df -h
Filesystem               Size  Used Avail Use% Mounted on
devtmpfs                 1.8G     0  1.8G   0% /dev
tmpfs                    1.8G  148K  1.8G   1% /dev/shm
tmpfs                    1.8G  880K  1.8G   1% /run
tmpfs                    1.8G     0  1.8G   0% /sys/fs/cgroup
/dev/mapper/fedora-root   11G  4.0G  6.0G  40% /
tmpfs                    1.8G  140K  1.8G   1% /tmp
/dev/sda1                477M  115M  334M  26% /boot
/dev/sr0                 953M  953M     0 100% /run/media/sean/Fedora-Live-Desktop-x86_64-20-1
[root@itmo456 ~]#
[root@itmo456 ~]# df -h /boot
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       477M  115M  334M  26% /boot
[root@itmo456 ~]#
[root@itmo456 ~]# df -i
Filesystem              Inodes  IUsed  IFree IUse% Mounted on
devtmpfs                466593    401 466192    1% /dev
tmpfs                   469081      8 469073    1% /dev/shm
tmpfs                   469081    517 468564    1% /run
tmpfs                   469081     15 469066    1% /sys/fs/cgroup
/dev/mapper/fedora-root 704512 110818 593694   16% /
tmpfs                   469081     28 469053    1% /tmp
/dev/sda1               128016    380 127636    1% /boot
/dev/sr0                     0      0      0     - /run/media/sean/Fedora-Live-Desktop-x86_64-20-1
[root@itmo456 ~]#
```

Disk Usage

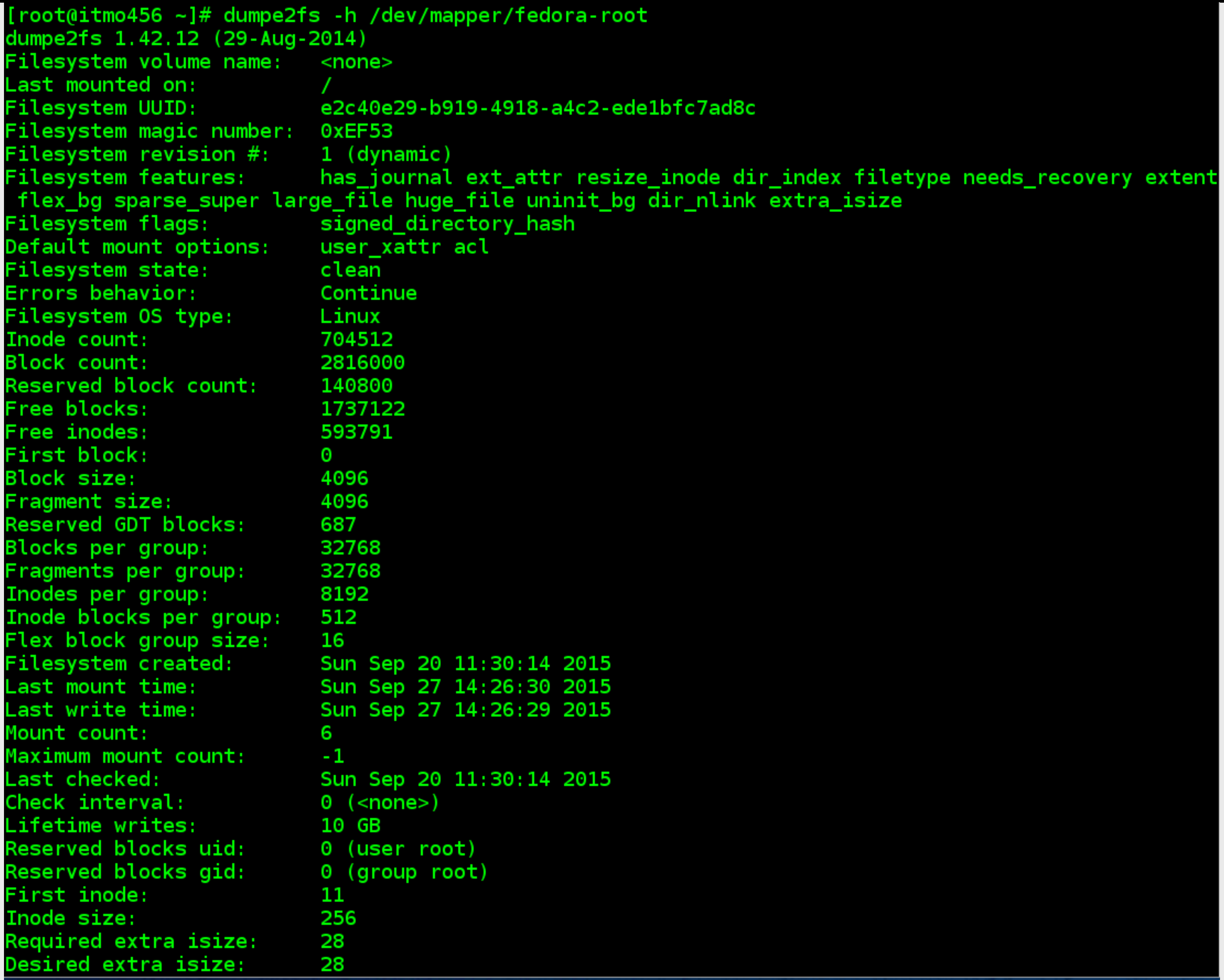
- **du** (directory usage) command
  - View size of a directory and contents in Kilobytes
  - **–s** option: Summarizes output
  - **–h** option: More user friendly
- **dumpe2fs** command
  - View total number of inodes and free inodes for ext2, ext3 or ext4 filesystem
  - Use **–h** option

Disk Usage



```
[root@itmo456 ~]# du -h /boot/
1.3M    /boot/extlinux
13K     /boot/lost+found
2.5M    /boot/grub2/fonts
2.1M    /boot/grub2/i386-pc
7.9M    /boot/grub2/themes/system
7.9M    /boot/grub2/themes
2.5M    /boot/grub2/locale
15M     /boot/grub2
1.4M    /boot/efi/EFI/BOOT
2.5M    /boot/efi/EFI/fedora/fonts
8.0M    /boot/efi/EFI/fedora
9.3M    /boot/efi/EFI
4.0K    /boot/efi/System/Library/CoreServices
6.0K    /boot/efi/System/Library
8.0K    /boot/efi/System
9.3M    /boot/efi
113M    /boot/
[root@itmo456 ~]#
[root@itmo456 ~]# du -hs /boot/
113M    /boot/
```

Disk Usage



```
[root@itmo456 ~]# dumpe2fs -h /dev/mapper/fedora-root
dumpe2fs 1.42.12 (29-Aug-2014)
Filesystem volume name:   <none>
Last mounted on:          /
Filesystem UUID:          e2c40e29-b919-4918-a4c2-ede1bfc7ad8c
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype needs_recovery extent
 flex_bg sparse_super large_file huge_file uninit_bg dir_nlink extra_isize
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              704512
Block count:              2816000
Reserved block count:     140800
Free blocks:              1737122
Free inodes:              593791
First block:              0
Block size:               4096
Fragment size:            4096
Reserved GDT blocks:      687
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         8192
Inode blocks per group:   512
Flex block group size:    16
Filesystem created:       Sun Sep 20 11:30:14 2015
Last mount time:          Sun Sep 27 14:26:30 2015
Last write time:          Sun Sep 27 14:26:29 2015
Mount count:              6
Maximum mount count:      -1
Last checked:             Sun Sep 20 11:30:14 2015
Check interval:           0 (<none>)
Lifetime writes:          10 GB
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               256
Required extra isize:     28
Desired extra isize:      28
```
]

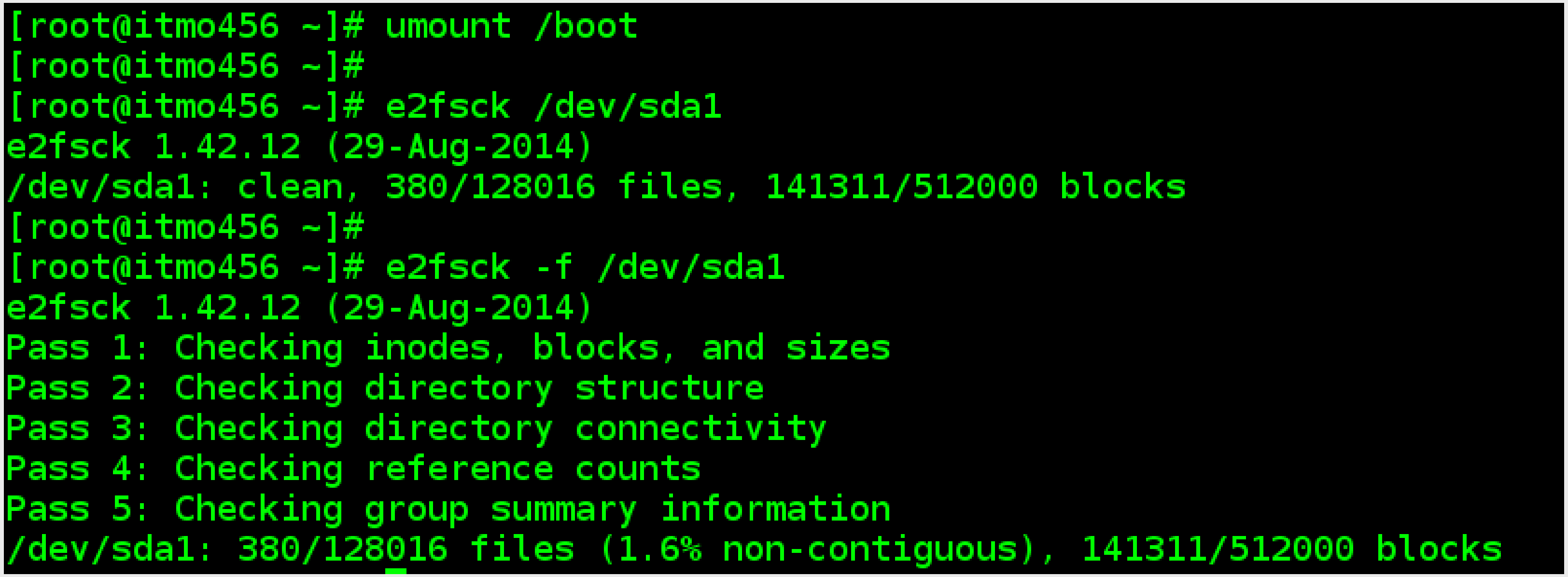
Checking Filesystems for Errors

- Filesystem corruption
  - Errors in a filesystem structure that prevent the retrieval of stored data
  - Commonly occurs due to improper system shutdown
- Syncing
  - Process of writing data stored in RAM to the HDD
- Bad blocks
  - Unusable areas of a disk
  - Cannot hold a magnetic charge

Checking Filesystems for Errors

- **`fsck`** (filesystem check) command
  - Check a filesystem for errors
  - Filesystem must be unmounted
  - **`–f`** option used to perform full check
- **`e2fsck`** command
  - Check ext2 ext3 and ext4 filesystems
- **`tune2fs`** command
  - Used to change filesystem parameters
  - **`-i`** option sets interval to forcing full system check

Checking Filesystems for Errors



```
[root@itmo456 ~]# umount /boot
[root@itmo456 ~]#
[root@itmo456 ~]# e2fsck /dev/sda1
e2fsck 1.42.12 (29-Aug-2014)
/dev/sda1: clean, 380/128016 files, 141311/512000 blocks
[root@itmo456 ~]#
[root@itmo456 ~]# e2fsck -f /dev/sda1
e2fsck 1.42.12 (29-Aug-2014)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sda1: 380/128016 files (1.6% non-contiguous), 141311/512000 blocks
```

Checking Filesystems for Errors

Option	Description
-f	Performs a full filesystem check
-a	Allows fsck to repair any errors automatically
-A	Checks all filesystems in /etc/fstab that have a 1 or 2 in the sixth field
-Cf	Performs a full filesystem check and displays a progress line
-AR	Checks all filesystems in /etc/fstab that have a 1 or 2 in the sixth but skips field the / filesystem
-V	Displays verbose output

Table 5-6: Common options to the **fsck** command

Checking Filesystems for Errors

Option	Description
-f	Forces checking even if filesystem seems clean
-p **-a**	Allows fsck to repair any errors automatically (-p is preferred; -a is for backwards compatibility)
-C 0	Performs a full filesystem check and displays a progress line
-D	Optimize directories by reindexing or sorting
-n	Opens filesystem read-only and assumes answer of “no” to all questions; allows e2fsck to be used on mounted drives
-y	Assumes answer of “yes” to all questions; allows e2fsck to be used non-interactively

Common options to the **e2fsck** command

Hard Disk Quotas

- If several users on a system, must be enough hard disk space for each user’s files
- Hard disk quotas
  - User limits on filesystem usage
  - Restrict number of files/directories or total disk space usage
- Soft limits
  - Limit imposed that can be exceeded for a certain period of time
- Hard limit
  - Limit imposed that cannot be exceeded

Hard Disk Quotas (continued)

- **`quotaon`** and **`quotaoff`** commands
  - Toggle quotas on and off
- **`edquota`** command
  - Edit user quotas
- **`repquota`** command
  - Report user quotas
- **`quota`** command
  - Allows regular users to view quotas and current usage

Hard Disk Quotas

- By default, quota support is not installed on Fedora
  - Must run **`yum install quota`** command from the command line to use

Summary

- Disk devices are represented by device files that reside in the /dev directory
- Each disk drive must contain a filesystem, which is then mounted to the Linux directory tree for usage using the **`mount`** command
- Hard disks must be partitioned into distinct sections before filesystems are created on those partitions
- Many different are filesystems available to Linux

Summary

- The LVM can be used to create logical volumes from the free space within multiple partitions
- Swap partions can be enabled with **`swapon`** and disabled with **`swapoff`**
- USB and FireWire storage devices are recognized as SCSI disks by the Linux system

Summary

- Important to monitor disk usage using the **`df`**, **`du`**, and **`dumpe2fs`** commands to avoid running out of storage space
- If hard disk space is limited, you can use hard disk quotas to limit the space that each user has on filesystems

The End…

- Questions?