

CST8177 – Linux II

Disks, Filesystems

Today's Topics

- ▶ sudo and PATH (environment)
- ▶ disks
- ▶ partitioning
- ▶ formatting file systems: mkfs command
- ▶ checking file system integrity: fsck command
- ▶ /etc/fstab
- ▶ mounting file systems: mount command
- ▶ unmounting file systems: umount command
- ▶ Isofs and fuse

Executing a command (review)

- ▶ builtin command (part of the shell itself, so there's no notion of "where" the command is)
 - `echo "Hello world"`
 - `exit 2` `#` inside a script, for example
- ▶ by absolute pathname (does not depend on `PATH` variable):
 - `/bin/ls -l`
 - `/usr/sbin/useradd newuser`
 - `/usr/bin/sudo -i`
 - `"$HOME"/bin/myscript.sh` `#` shell expands `$HOME` so this is really `/home/username/bin/myscript.sh`

Executing a command (cont'd)

- ▶ by relative pathname (does not depend on PATH variable, but DOES depend on your current directory – interactive shells only)
- ▶ You MUST NOT do any of these in a shell script
 - ./myscript.sh # script is in current directory
 - ../myprogram # script is in parent directory
 - ../../somedir/anotherscript.sh # two dirs up, then one directory down
 - bin/mycommand # assumes "bin" is a directory in the current directory

Executing a command (cont'd)

- ▶ using the PATH environment variable
 - `ls -l`
 - `cp foo ../bar`
 - `rm ../bar/foo`
- ▶ none of these commands will run unless they reside in a directory that is listed in the PATH environment variable
- ▶ Now that we are using root privileges, we need to be aware that root can have a different PATH than your non-root user

sudo and your environment

- ▶ sudo command # just run the command
 - you get 5 min by default to invoke sudo again without password
 - example\$ sudo head /etc/shadow
- ▶ sudo -s # superuser shell with current env
- ▶ sudo -i # simulate root login (root's env)
- ▶ sudo -s leaves you in the same directory, and with the same PATH
- ▶ to take on root's environment including PATH:
 - sudo -i
 - or
 - sudo -s followed by su -

Disks and disk management

- ▶ partitioning
- ▶ formatting file systems
- ▶ mounting file systems
- ▶ `/etc/fstab`

Overview of partitioning (8207 review)

- ▶ A partition is a section of disk forming a physical volume that contain a filesystem, or swap space, or be used as a component in LVM or RAID
- ▶ The **Master Boot Record** contains the **Disk Partition Table**, which can hold up to four entries due to the way in which the master boot record is structured
 - With certain specialty tools, you can create more than four partitions, but we'll stick to the MSDOS partition table format
- ▶ Each Disk Partition Table entry describes a partition by specifying its:
 - first cylinder
 - last cylinder
 - whether it is bootable
 - a partition type identifier.

Partitioning

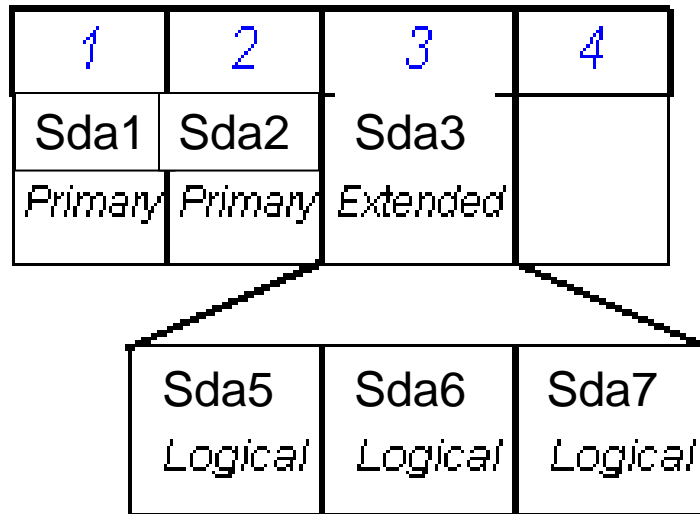
- ▶ We deal primarily with the MSDOS Partition Table type
- ▶ GPT partition tables getting common: GUID Partition Table
- ▶ Globally Unique Identifier (but back to MSDOS Tables...)

- ▶ Up to four Primary Partitions are possible in a single table

- ▶ At most one of the four **Primary partitions** can be an **Extended Partition**

- ▶ **Logical Partitions** can be created inside an Extended Partition

Identifying Partitions



Naming partitions

- **sd x 1 - sd x 4**
 - Primary Partitions recorded in the partition table
- **sd x 5 - sd x 63**
 - Logical partitions

Note: You can have up to 4 primary partitions created in your system, while there can be only one extended partition.

Options for Partitioning

- ▶ DOS **fdisk** program
 - Very limited Linux support
- ▶ Linux **fdisk** program (we use this)
 - similar to DOS fdisk, but more features available
 - can only be used under Linux/UNIX
- **parted** can handle more partition table types (e.g. GPT)
- ▶ **Disk Druid** program
 - Part of the Fedora installation system
 - Cannot be run on its own
- ▶ **gparted** (*Fedora, Ubuntu*)
 - Gnome Partitioning Editor: GUI based partitioning
 - only runs from within Linux/UNIX

Linux **fdisk** command

- ▶ **fdisk [options] device**
 - command-line partition table manipulator for Linux
 - allows for viewing or modifying existing partition table and/or creating new partition(s) for a specified device
 - can set Partition Type for most of the common files systems in use today
 - `fdisk -cul /dev/sda`

Add a disk

- ▶ power down machine (or virtual machine)
- ▶ add hard disk
- ▶ power up machine
- ▶ verify the new disk was detected (following slide)
- ▶ if the disk was brand new, it won't be partitioned (our example is this case)
- ▶ if the disk is being reused, be sure you can identify its partitions and you do not need the data

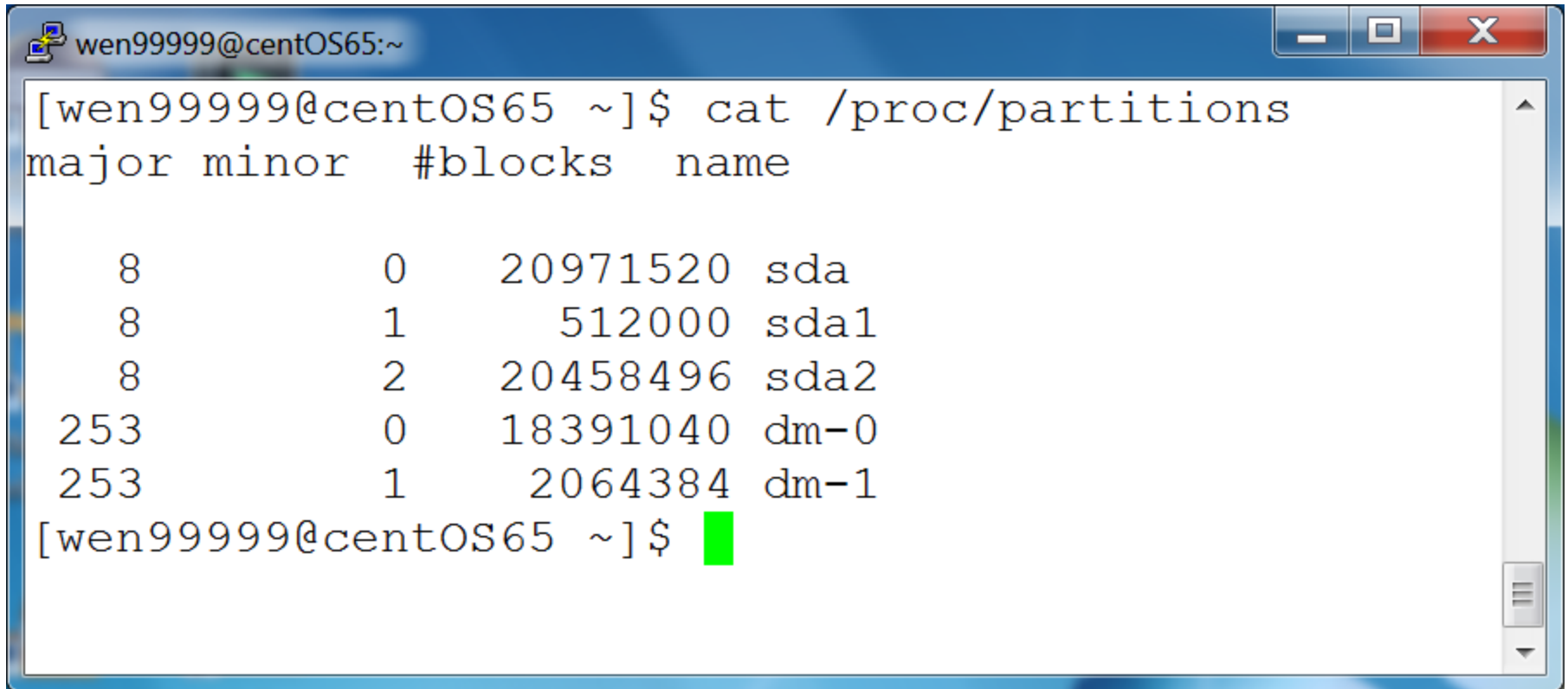
dmesg: kernel ring buffer

- ▶ http://teaching.idallen.com/cst8207/14f/notes/580_system_log_files.html
- ▶ kernel messages are kept in a ring buffer
- ▶ common way to access the boot messages, including device discovery
- ▶ dmesg
- ▶ example: look for disk discovery:
 - `dmesg | grep sd`
- ▶ (another way): look at disks/partitions that the kernel knows about:
 - `cat /proc/partitions`

dmesg

```
wen99999@centOS65:~  
[wen99999@centOS65 ~]$ dmesg |grep sd  
sd 2:0:0:0: [sda] 41943040 512-byte logical blocks: (21.4 GB/20.0 GiB)  
sd 2:0:0:0: [sda] Write Protect is off  
sd 2:0:0:0: [sda] Mode Sense: 61 00 00 00  
sd 2:0:0:0: [sda] Cache data unavailable  
sd 2:0:0:0: [sda] Assuming drive cache: write through  
sd 2:0:0:0: [sda] Cache data unavailable  
sd 2:0:0:0: [sda] Assuming drive cache: write through  
sda: sda1 sda2  
sd 2:0:0:0: [sda] Cache data unavailable  
sd 2:0:0:0: [sda] Assuming drive cache: write through  
sd 2:0:0:0: [sda] Attached SCSI disk  
dracut: Scanning devices sda2 for LVM logical volumes vg_centos65/lv_swap vg_centos65/lv_root  
sd 2:0:0:0: Attached scsi generic sg1 type 0  
EXT4-fs (sda1): mounted filesystem with ordered data mode. Opts:  
sd 2:0:0:0: [sda] CDB: Write(10): 2a 00 01 14 5d 00 00 00 10 00  
sd 2:0:0:0: [sda] CDB: Write(10): 2a 00 00 11 56 78 00 00 08 00  
sd 2:0:0:0: [sda] CDB: Write(10): 2a 00 01 14 5e 88 00 00 10 00  
sd 2:0:0:0: [sda] CDB: Write(10): 2a 00 00 11 56 78 00 00 08 00  
[wen99999@centOS65 ~]$ █
```

/proc/partitions



```
wen99999@centOS65:~  
[wen99999@centOS65 ~]$ cat /proc/partitions  
major minor #blocks name  
  
8 0 20971520 sda  
8 1 512000 sda1  
8 2 20458496 sda2  
253 0 18391040 dm-0  
253 1 2064384 dm-1  
[wen99999@centOS65 ~]$
```

| major | minor | #blocks | name |
|-------|-------|----------|------|
| 8 | 0 | 20971520 | sda |
| 8 | 1 | 512000 | sda1 |
| 8 | 2 | 20458496 | sda2 |
| 253 | 0 | 18391040 | dm-0 |
| 253 | 1 | 2064384 | dm-1 |

Create partition on new disk

- ▶ use fdisk to partition the new disk

```
root@centOS65:~  
[root@centOS65 ~]# fdisk -cu /dev/sda  
  
Command (m for help): p  
  
Disk /dev/sda: 21.5 GB, 21474836480 bytes  
255 heads, 63 sectors/track, 2610 cylinders, total 41943040 sectors  
Units = sectors of 1 * 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
Disk identifier: 0x0006a03a  
  
   Device Boot      Start         End      Blocks   Id  System  
/dev/sda1  *          2048     1026047     512000   83   Linux  
/dev/sda2             1026048     41943039    20458496   8e   Linux LVM  
  
Command (m for help): █
```

File systems (8207 review)

- ▶ http://teaching.idallen.com/cst8207/14f/notes/720_partitions_and_file_systems.html

Linux/Unix mounting

/dev/sda2

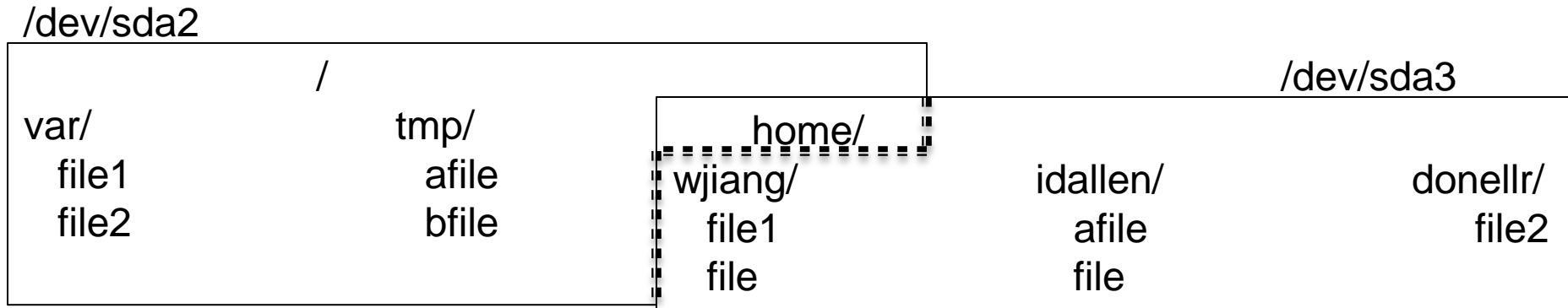


/dev/sda3



Linux/Unix mounting

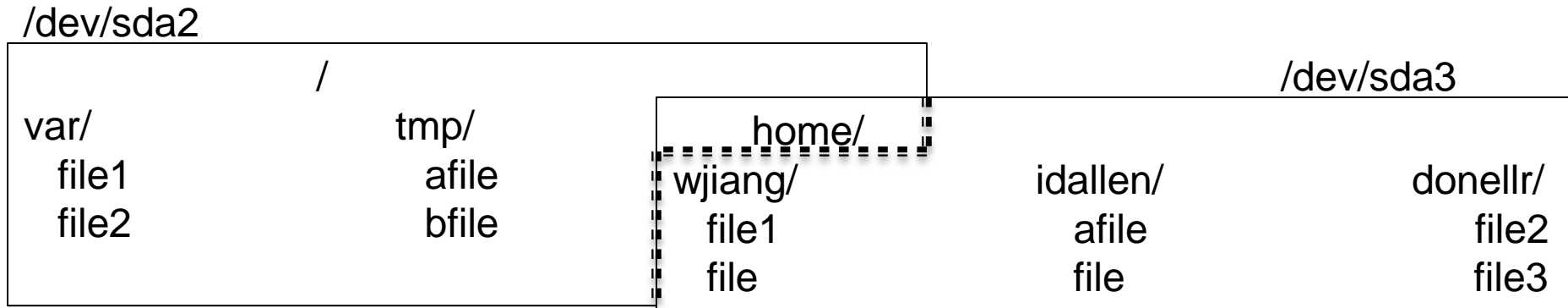
- ▶ `mount /dev/sda3 /home`



- ▶ the `/home` directory name still on `/dev/sda2`
- ▶ the contents of `/home` are on `/dev/sda3`
- ▶ the previous contents of `/home` are hidden

Linux/Unix mounting

- ▶ touch /home/donellr/file3



Linux/Unix mounting

- ▶ `umount /dev/sda3`

`/dev/sda2`



`/dev/sda3`



/etc/fstab

- ▶ `fsck`, `mount`, and `umount` use this file
- ▶ `man 5 fstab`
- ▶ note that records for swap space appear in `/etc/fstab`, although swap space is not a filesystem (files are not stored in swap space)
- ▶ first field: device name
- ▶ second field: mount point
- ▶ third field: type
- ▶ fourth field: mount options
- ▶ fifth field: backup related (dump program)
- ▶ sixth field: file system check order

/etc/fstab commands

- ▶ `mount -a`
 - issued as part of the boot process
 - all file systems listed in `/etc/fstab` will be mounted accordingly (except those with "noauto" option or "ignore" file system type)
- ▶ `mount <mount point>`
 - `mount` will consult `/etc/fstab` to find the device and options for that mount point, and mount it
- ▶ `mount <device>`
 - `mount` will consult `/etc/fstab` to find the mount point and options for that mount point, and mount it

/etc/fstab: device name

- ▶ device name, the first field, names the block special device (e.g. `/dev/sda1`) on which the file system resides
- ▶ the first field can also be expressed in terms of LABEL or UUID (e.g. LABEL=root) (e.g. see the `/etc/fstab` on our CentOS 6.5 machines)
 - `blkid` command prints the UUIDs of the system's block devices
 - `e2label` command prints/sets file system labels

/etc/fstab: mount point

- ▶ The mount point is the directory on which the file system should be mounted
- ▶ swap is not a file system but is still controlled by /etc/fstab, so the mount point is `none`

/etc/fstab: file system type

- ▶ `ext4` is the file system type we use often
- ▶ `/proc/filesystems` contains the list of file systems supported by the currently running kernel
- ▶ `swap` for swap space
- ▶ `ignore` for an unused filesystem
- ▶ `none` for bind mounts

`/etc/fstab`: mount options

- ▶ fourth field in `/etc/fstab`
- ▶ expressed as a comma-separated list
- ▶ different file systems support different options (see `man 8 mount`)
- ▶ `defaults`: a set of default options
- ▶ example options common to all file system types:
 - `noauto`: do not mount when "mount -a" called
 - `user`: allow a user to mount
 - `owner`: allow device owner to mount

/etc/fstab: options (cont'd)

▶ mount options

◦ on CentOS 6.6, "defaults" means

- rw: read and write
- dev: interpret device nodes
- suid: setuid and setgid bits take effect
- exec: permit execution of binaries
- auto: mount automatically due to "mount -a"
- nouser: regular users cannot mount
- async: file I/O done asynchronously
- relatime: update access times a certain way

▶ other options:

- ▶ these are for quota utilities to see rather than mount
 - ▶ usrquota
 - ▶ grpquota

/etc/fstab: dump

- ▶ the dump program uses this field to determine which file systems should be backed up by the dump command
- ▶ the dump program can back up an entire file system to tape, for example
- ▶ dump supports incremental backups
- ▶ when restoring, it can provide an index of what's in the file system, do partial restores, etc
- ▶ we don't use dump in this course

/etc/fstab: fsck order

- ▶ the sixth and last field is used to determine the order in which file system checks are done at boot
- ▶ root file system: 1
- ▶ other file systems: 2
- ▶ no fsck: 0

`/etc/mtab` and `/proc/mounts`

- ▶ `/etc/mtab` is used by `mount` and `umount` to keep track of what is currently mounted
- ▶ `mount` command (no args) prints this file
- ▶ `/proc/mounts` is the kernel's list of what's mounted, and might be more up-to-date than `/etc/mtab`

Adding a disk

- ▶ # migrating the /usr directory to be a separate partition on new disk
- ▶ shut down machine
- ▶ connect new disk to machine
- ▶ power on machine
- ▶ partition new disk (fdisk command)
- ▶ make filesystem in new partition (mkfs command)
- ▶ single user mode (shutdown command)
- ▶ ensure target directory is backed up
- ▶ move the target directory out of way (/usr to /usr1) (mv command)
- ▶ create the mount point (to replace dir we just moved, same name)
- ▶ mount new filesystem (mount command)
- ▶ /usr1/bin/rsync -aHv /usr1/. /usr (notice where rsync is!)
- ▶ add a record for the new filesystem /etc/fstab
- ▶ exit, to return to runlevel 3
- ▶ remove /usr1 (content should be backed up)

device busy

- ▶ when trying to unmount a filesystem, you might get an error:

```
umount: /dirname: device is busy
```

- ▶ probably some process is using the filesystem (it's busy -- make sure you're not in that directory!)
- ▶ `lsdf /mountpoint # list open files in the filesystem mounted on /mountpoint`

```
lsdf +D /directory
```

this will show you what processes are using the directory or (+D) any directory under it

Isof and fuser

- ▶ Note the difference between a mountpoint and a directory
 - mountpoint: both of these commands will apply to the entire filesystem mounted there
 - directory: both of these commands will apply to just that directory, not recursively every subdirectory underneath it
- ▶ summary of Isof:
 - <http://www.thegeekstuff.com/2012/08/Isof-command-examples/>
- ▶ fuser: similar in purpose to Isof
- ▶ examples:
 - `fuser /mountpoint # all processes using the filesystem mounted at /mountpoint`
 - `fuser /home/dir # all processes using the directory dir`
- ▶ summary of fuser:
 - <http://www.thegeekstuff.com/2012/02/linux-fuser-command/>