CST8177 – Linux II

Disks, Filesystems, Booting

Todd Kelley
kelleyt@algonquincollege.com

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
- lsof and fuser

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 has 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
- LVM
- 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 files ystem, 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

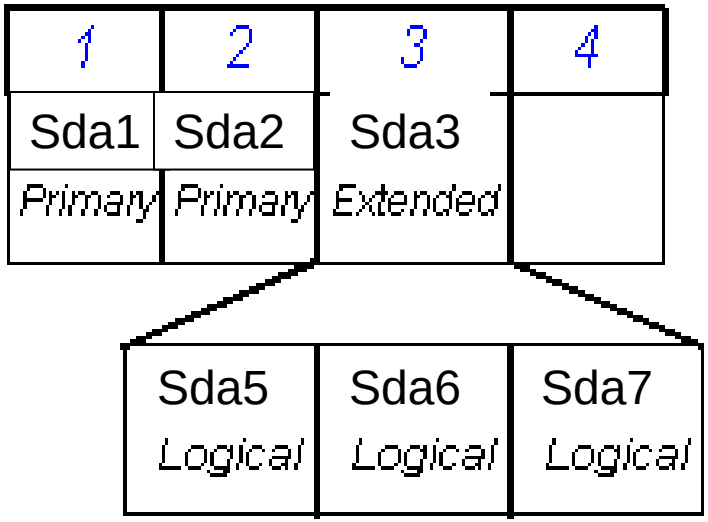
- We deal primarily with the MSDOS Partition Table type
- GPT partition tables getting common: GUID Partition Table
- Globally Unique IDentifier (but back to MSDOS Tables…)

- 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.

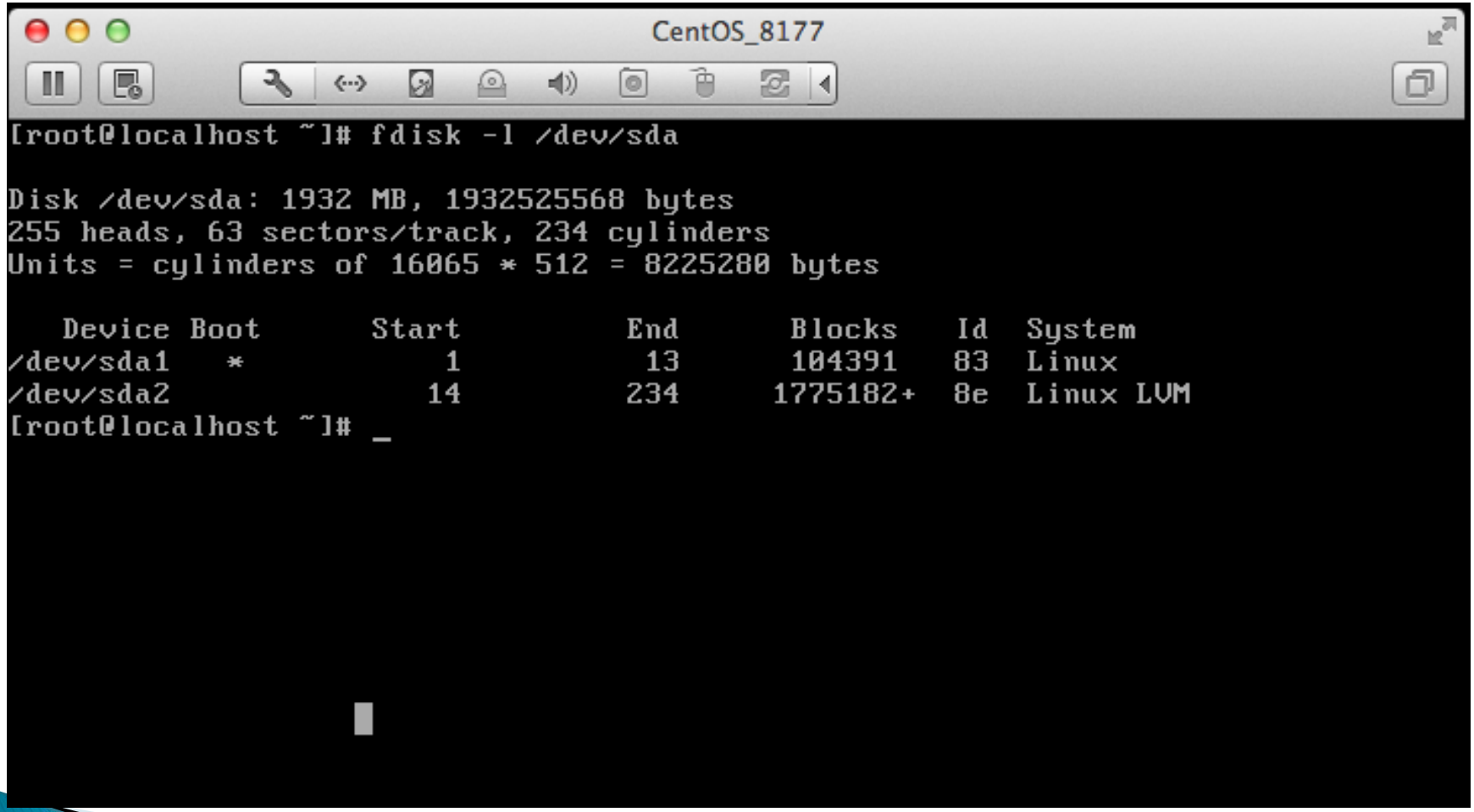
Options for Partitioning

- 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 –l /dev/sda

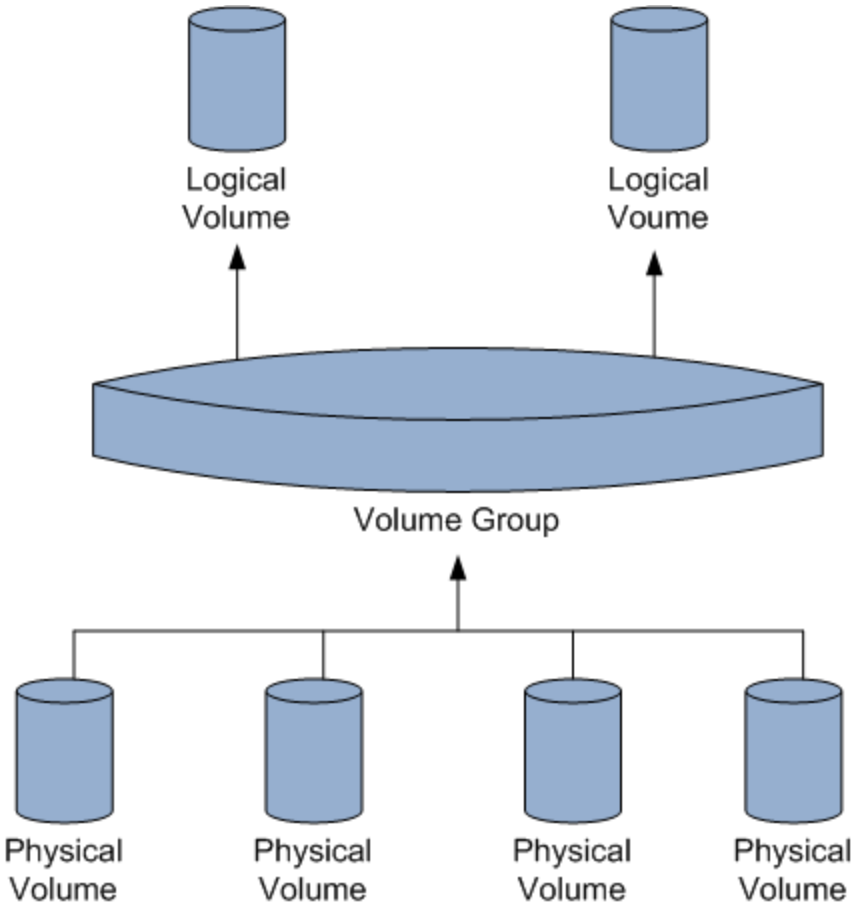
Our Partition Table



LVM

- Logical Volume Manager
- LVM tutorial:
  - http://www.howtoforge.com/linux_lvm
- disk partitions are physical volumes
- one or more physical volumes forms a volume group
- a volume group can be divided into logical volumes
- We create file systems on the logical volumes

LVM Logical Volume Components



So… What happened when we installed CentOS?

- By default, LVM was used to set up the /dev/sda2 partition.
- Disk Druid set up **/dev/sda1** and **/dev/sda2**
- **/dev/sda1** was set as bootable and contains **/boot** (no LVM involved with /dev/sda1)
- To actually see where things are you can do the following:
  - **mount**
  - **lvdisplay # show logical volumes**
  - **pvdisplay # show physical volumes**

Our CentOS LVM setup

- /dev/sda divided into 2 partitions:
  - /dev/sda1 : boot partition (no LVM)
  - /dev/sda2 : physical volume for LVM
- /dev/sda2 is the only physical volume in VolGroup00

```
[root@localhost ~]# pvdisplay
  --- Physical volume ---
  PV Name               /dev/sda2
  VG Name               VolGroup00
  PV Size               1.69 GB / not usable 5.58 MB
  Allocatable           yes (but full)
  PE Size (KByte)       32768
  Total PE              54
  Free PE               0
  Allocated PE          54
  PV UUID               AXOKPV-ille-XAms-O2t2-njXZ-60jH-d1vFpV

[root@localhost ~]# _
```

Logical Volumes

- VolGroup00 is divided into 2 logical volumes
- LogVol00 is root filesystem, LogVol01 is swap

```
  --- Logical volume ---
  LV Name                /dev/VolGroup00/LogVol00
  VG Name                VolGroup00
  LV UUID                xbRiUo-O64p-jvMg-7Ahu-kec6-9KoX-7UWZQv
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                1.34 GB
  Current LE             43
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0

  --- Logical volume ---
  LV Name                /dev/VolGroup00/LogVol01
  VG Name                VolGroup00
  LV UUID                5Qqj03-N4GN-gd2E-OFsy-vchx-80ZM-asXPzr
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                352.00 MB
  Current LE             11
--More--
```

LVM commands

- We could do by hand what the Red Hat installer did:
- pvcreate /dev/sda2 # initialize /dev/sda2 as physical volume for LVM
- vgcreate VolGroup00 /dev/sda2 #create volume group (a group of 1: /dev/sda2 is the only physical volume in group)
- lvcreate --name LogVol00 --size 1.34G VolGroup00
  - create a logical volume LogVol00 in volume group VolGroup00
- lvcreate --name LogVol01 --size 352M VolGroup00
  - create a second logical volume LogVol01 in volume group VolGroup00
- mkfs -t ext3 /dev/VolGroup00/LogVol00
  - make a file system in logical volume LogVol00
- mkswap /dev/VolGroup00/LogVol01
- swapon /dev/VolGroup00/LogVol01
  - use the other logical volume LogVol01 for swap space

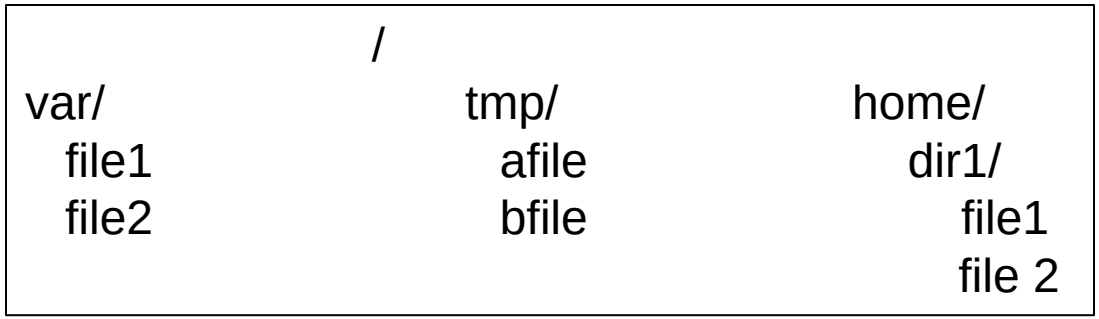
File systems (8207 review)

- http://teaching.idallen.com/cst8207/13w/notes/720_partitions_and_file_systems.html

Linux/Unix mounting

- no drive letters!

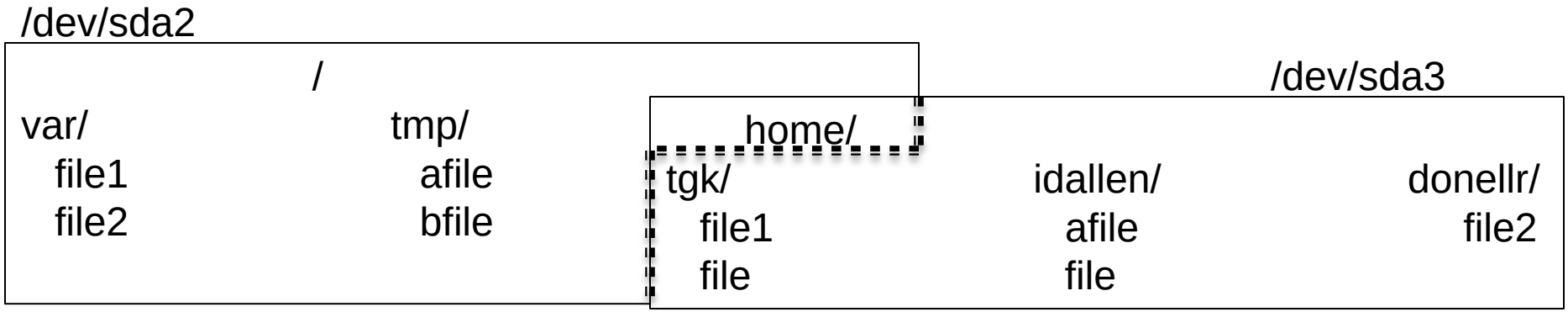
/dev/sda2



/dev/sda3

Linux/Unix mounting

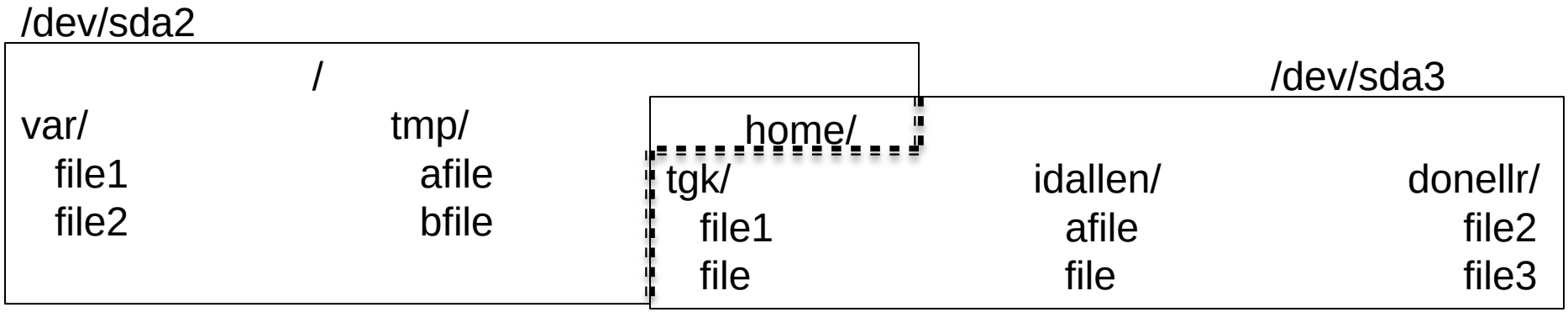
- mount /dev/sda3 /home



- 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

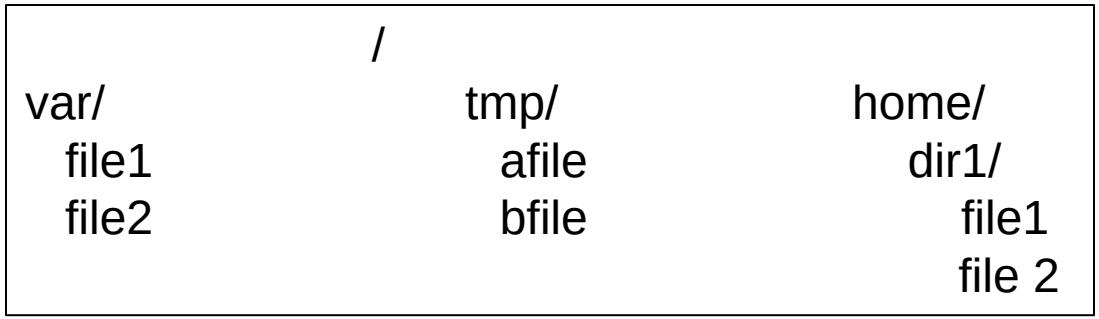
- touch /home/donellr/file3



Linux/Unix mounting

- umount /dev/sda3

/dev/sda2



/dev/sda3

/etc/fstab

- 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 (cont'd)

- mount options
  - on CentOS 5.8, "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
- other options:
  - these are for quota utilities to see rather than mount
    - usrquota
    - grpquota

dmesg: kernel ring buffer

- http://teaching.idallen.com/cst8207/13w/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

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!)
- lsof /mountpoint # list open files in the filesystem mounted on /mountpoint

lsof +D /directory

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

lsof 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 lsof:
  - http://www.thegeekstuff.com/2012/08/lsof-command-examples/
- fuser: similar in purpose to lsof
- 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/

Booting

- http://teaching.idallen.com/cst8207/13w/notes/750_booting_and_grub.html
- page numbers for Fifth Edition Sobell:
  - Chapter 11: 424-431
  - Chapter 15: 551-552