Filesystems

CIS 2235 Advanced System Administration

Outline

- Filesystems as a database
- Links
- Partitions
- mount & fstab
- Other partition considerations: `du`

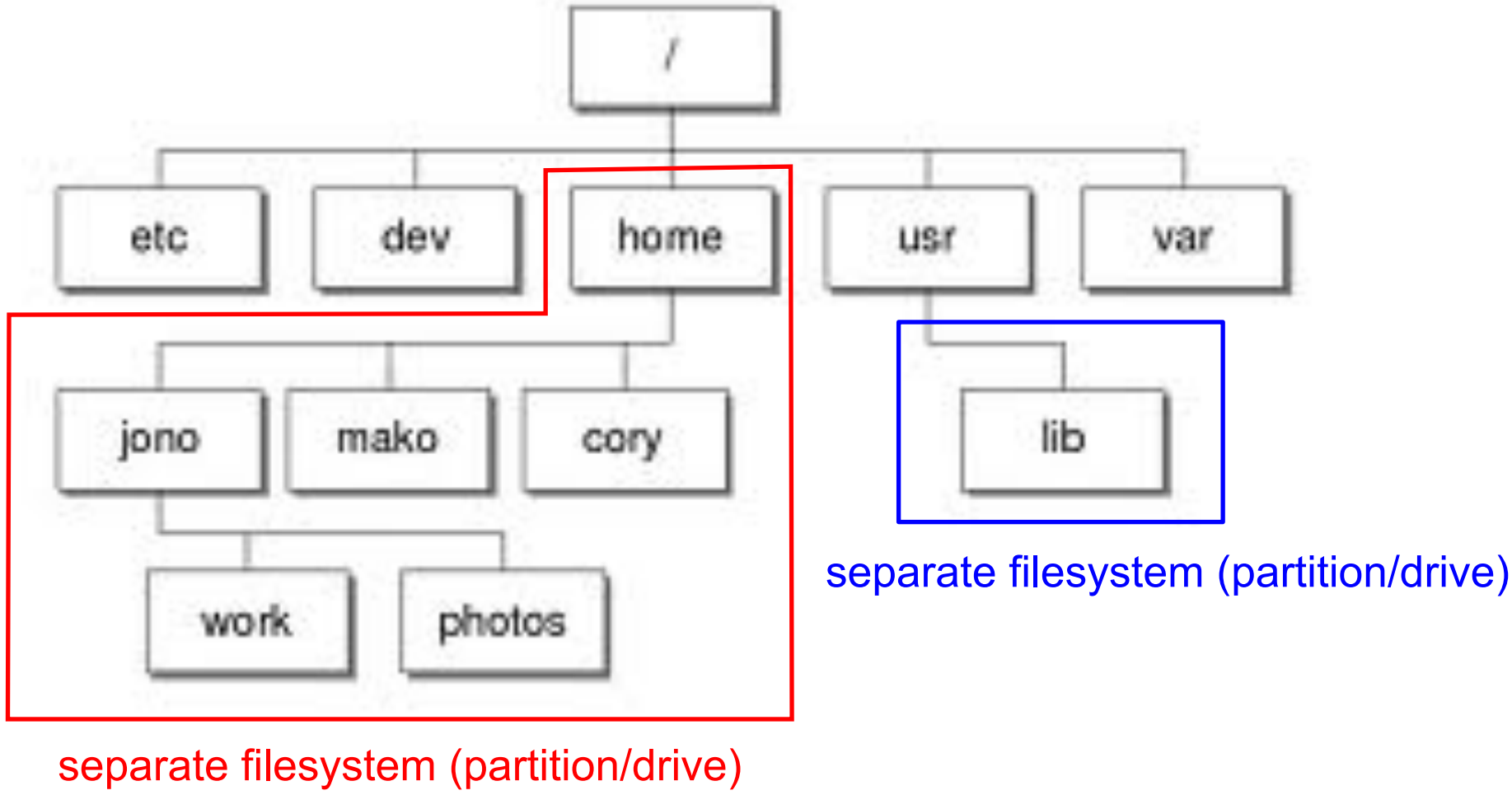
Filesystems

- Commonly used to refer to two distinct but related concepts:
  - The *hierarchy* of directories and files that humans use to organize data on a system ('unified filesystem'). Linux has a Filesystem Hierarchy Standard (FHS).
  - The *formatting* system that the kernel uses to store blocks of data on physical media such as disks ('filesystem types').
- We will cover *both* concepts.

The Unified Filesystem (UF)

- Unix and Linux systems have a unified filesystem
  - Any files can be accessed through a name beginning with /
  - How is Windows different?
- The unified filesystem can be made up of one or more *individual* filesystems.
  - Each filesystem has its own root (/)
  - That root can be grafted onto any directory in the unified filesystem
  - Called a ***mount point***
  - Same computer or a different one (network drive)

The Unified Filesystem



File Types

"Everything in Unix is a file."

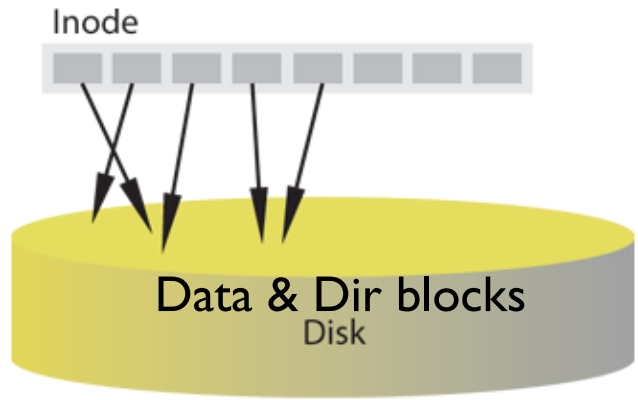
- *Files* directly contain data.
- *Directories* (or *folders*) provide a hierarchy of files.
- *Devices* are special files.
  - Device files provide a way of asking the kernel for access to a given device.
  - The data the device file contains is the raw sequence of bytes or sectors on the device itself.
  - Device files are, by convention, stored under the /dev

Filesystems

- The filesystem is made up of (basically) three types of elements:
  - Inodes
  - Data blocks
  - Directory blocks

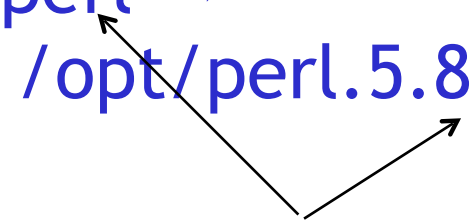
Inodes

- An inode is the data structure that describes a file on an individual filesystem.
- It contains information about the file, such as:
  - Type: file, directory, device or link
  - Metadata: exact size, modification time, permissions, owners, disk address, etc.
- The inode does **not** contain the filename.
- There is a *fixed number* of inodes within an individual filesystem.
- They are numbered.



Directories

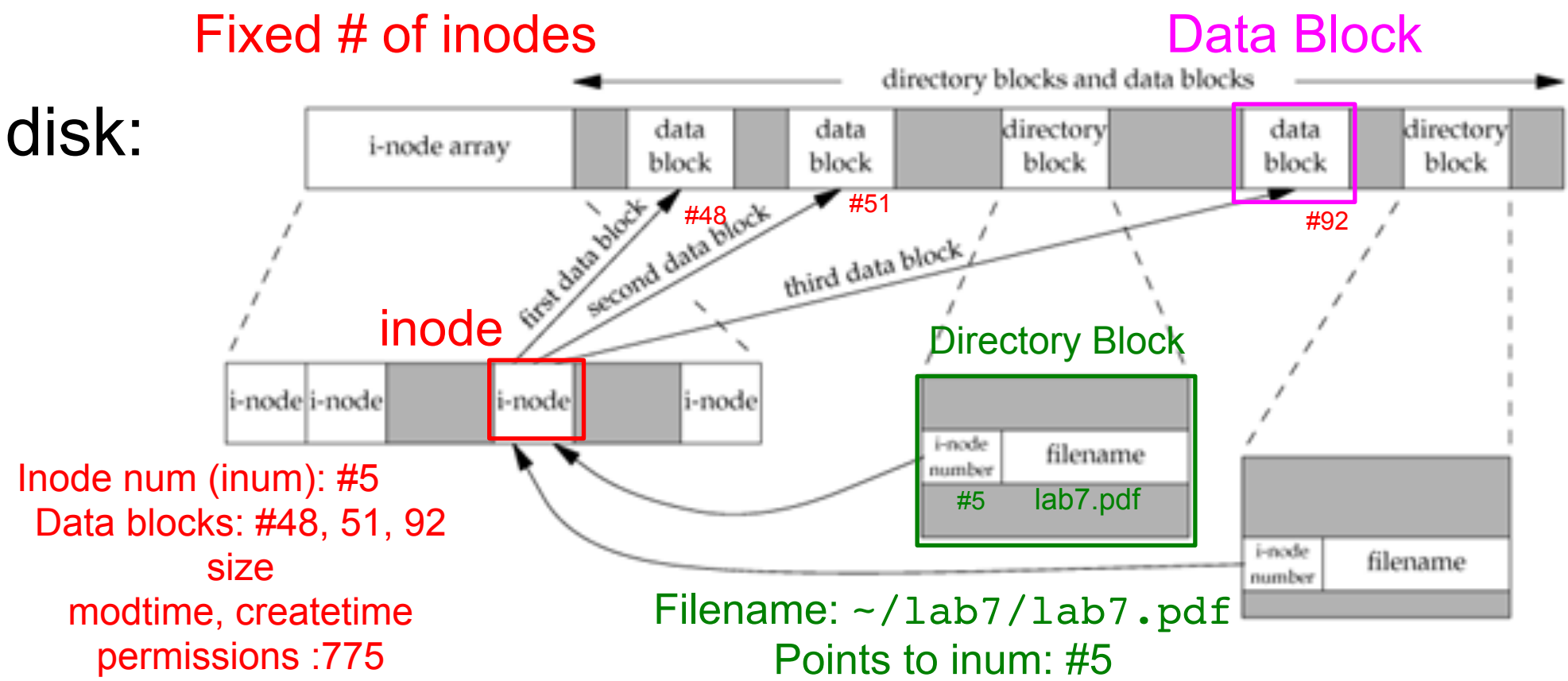
- A Directory Block is a list of inode/name pairs.
  - Names could be files or directories
  - This allows you to have multiple names for the same underlying file — i.e., links
  - /usr/bin/perl →
    /opt/perl.5.8.1/bin/perl581



same inode (actual executable)
but different filenames in different dirs

Remember the 3 Filesystem elements

inode, data blocks, directory blocks

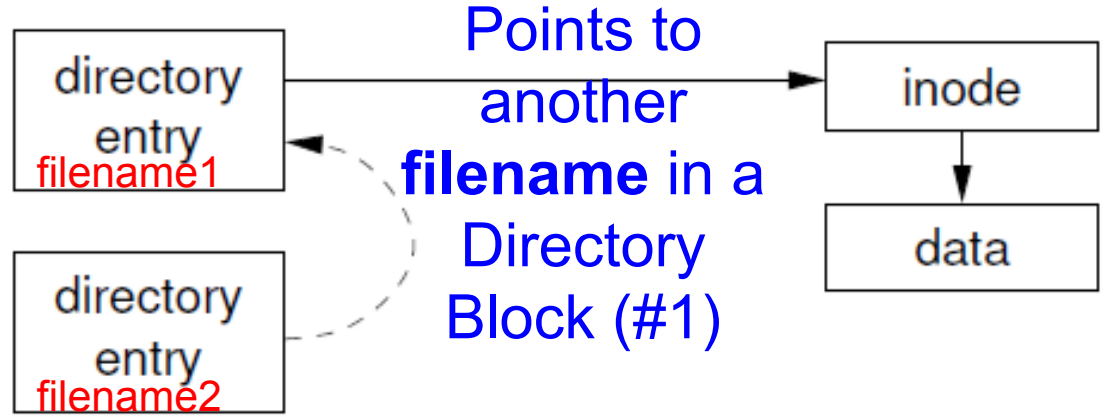


Symbolic Links

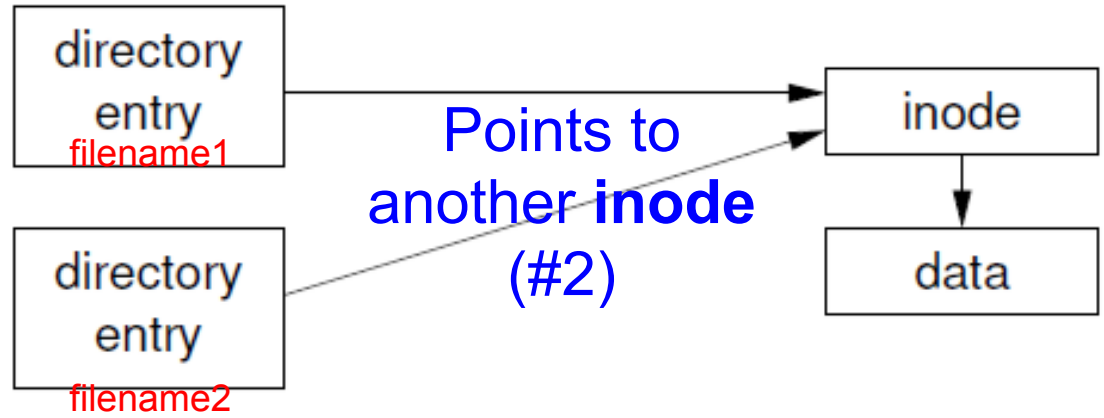
- A symbolic link (symlink) is a pointer to another file or directory.
- Symlinks allow you to keep a file (or directory) in one place but 'pretend' it lives in another.
- Typical Uses:
  - To ensure that an obsolete name continues to work for older software.
  - Or to spread data/programs from a single filesystem hierarchy over multiple disk partitions.
- Two types:
  - The link points to an inode (the actual data) => hard link
  - The link points to a filename in a directory block => soft link

Hard Link & Symbolic Link

- A symbolic link refers to filename, which in turn refers to an inode:



- A hard link is a normal directory entry, referring directly to an inode:



Links

- A hard link associates the same data (1 node) with two or more filenames
  - ~/labs/lab7.pdf → inode #5
  - ~/courses/cis22350/lab7/report.pdf → inode #5
- On the other hand, a soft link is a file that points to another filename
  - That filename then has another inode and data block.
  - ~/labs/lab7.pdf → inode #5
  - ~/courses/cis22350/lab7/report.pdf → ~/labs/

What's the Difference?

- Hard links
  - Cannot link directories
  - Cannot cross file system boundaries (b/c inode numbers don't match!)
- Symlinks
  - Can create links between directories
  - Can cross file system boundaries
- What if the source of the link is removed?
  - Symbolic links are not updated.
  - If the source is removed, the symlink doesn't know! Broken
  - Hard links always refer to the source.

Creating Symbolic Links

- A symlink is created using the `ln` command
- Similar to `cp` — the original name comes first, then the link name you want to create:

```
$ ln -s real-file file-link
$ ln -s real-dir dir-link
```

- The file looks 'normal'. Use $ ls -l to see that it's a link.

```
$ ls -l
lrwxrwxrwx 1 bob bob 9 Jan 11 15:22 file-link -> real-file
lrwxrwxrwx 1 bob bob 8 Jan 11 15:22 dir-link -> real-dir
```

Steps in Adding More Storage

What if we need more space?

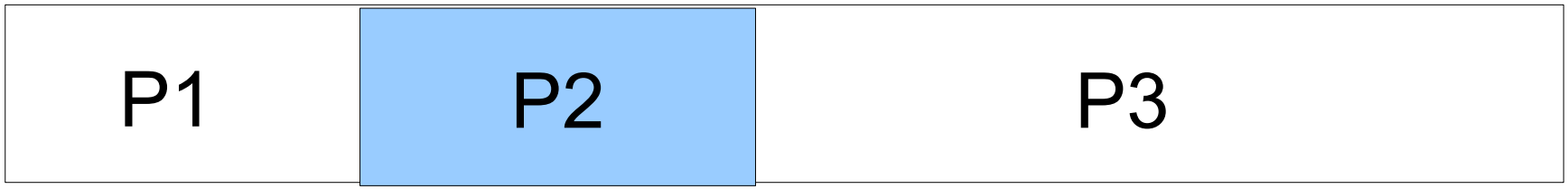
1. Physically connect the disk.
2. Create appropriate partitions.
3. Create a filesystem on each partition.
4. Mount the partition into the filesystem.

Concepts: Disks and Partitions

- A physical hard disk provides a single large storage space
- Usually split into *partitions*
  - Information about partitions is stored in the *partition table*.
  - Linux defaults to partition tables compatible with Microsoft Windows ('ms-dos').
  - For compatibility with Windows, at most four (4) primary partitions can be made (2-bits).
  - However, they can be *extended* partitions and split into smaller logical partitions.

Concepts: Disks and Partitions

800Mb hard drive



250Mb 150Mb 400Mb

Windows:	C:	D:	E:
Linux:	/	/home	/data

Why Partitions?

There are many reasons to use partitions.

- Differing backup strategies
- Corruption, which requires re-formatting
- Security
- Speed (put /tmp on own HD)
- OS swap/patch
- Share between dual boot

Partition strategies:

- Example partition setup for "small" computer
  - 3 partitions:
  - 1st OS, /home (backup), /data (no backup)
- For a server, you might have many more partitions:
  - /tmp, /usr, /var, /local, /home, etc.

IDE Controller Naming Convention

- IDE is the 'ribbon-cable' protocol
- Two channels: "primary" and "secondary."
- Each channel has two drives:
  - The first drive on each channel is the IDE 'primary,' and the second is the IDE 'secondary.'
- Channel/drive combos:
  - primary-primary, primary-secondary, secondary-primary, secondary-secondary
- In Linux
  - IDE drives start with "/dev/hd"
  - The four combos are: /dev/hda to /dev/hdd

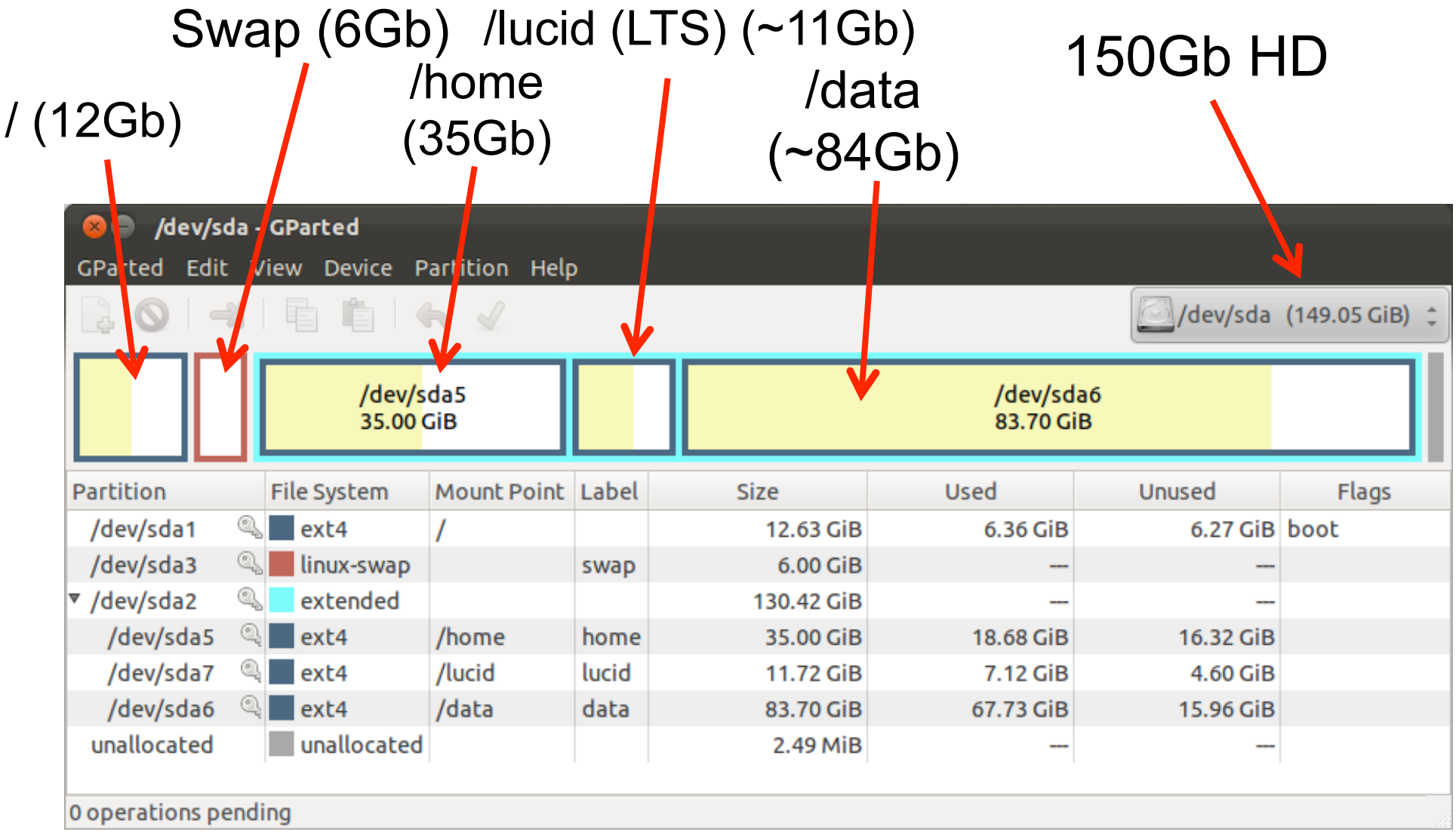
SCSI Controller Naming Convention

- SCSI disks start with "/dev/sd"
- SCSI controllers have 16-channels
- The disks are named /dev/sda, /dev/sdb, etc

Partition Naming Convention

- Partitions within the physical disk are numbers appended to the name
- *Primary* partitions are numbered from 1- 4
- *Logical* or *extended* partitions are numbered 5+
- Examples:
  - /dev/sda1 – Partition 1 on the first SCSI drive.
  - /dev/hdc2 – Partition 2 on the third IDE drive.
  - /dev/sde4 – Partition 4 on the fifth SCSI drive.

My Partitions



Partition Commands

- `fdisk` - partition table manipulator for Linux
- `parted` - partition editor
- `mkfs` - formats (makes) a Linux file system
- gparted - GUI partition editor
  - `http://gparted.sourceforge.net`
  - `https://help.ubuntu.com/community/GParted`

- Image to think about:
  - The disk is a bulky wooden *crate*
  - The partitions are *cardboard boxes* in the crate
  - The filesystem is a *foam liner* fit for the particular cardboard box size, ready to hold the product.

Filesystem Types

- The most common filesystem types are:

	Type	Usage
ext3, ext4	`ext2`	The standard Linux filesystem
	`iso9660`	The filesystem used on CD-ROMs
	`proc`	Not a real filesystem, so uses `none` as the device. Used as a way for the kernel to report system information to user processes
	`vfat`	The filesystem used by Windows 95
	`auto`	Not a real filesystem type. Used as a way of asking the `mount` command to probe for various filesystem types, particularly for removable media

- Networked filesystems include `nfs` (Unix-specific) and `smbfs` (Windows or Samba)
- Other, less common types exist; see mount(8)

Common File Systems

Table

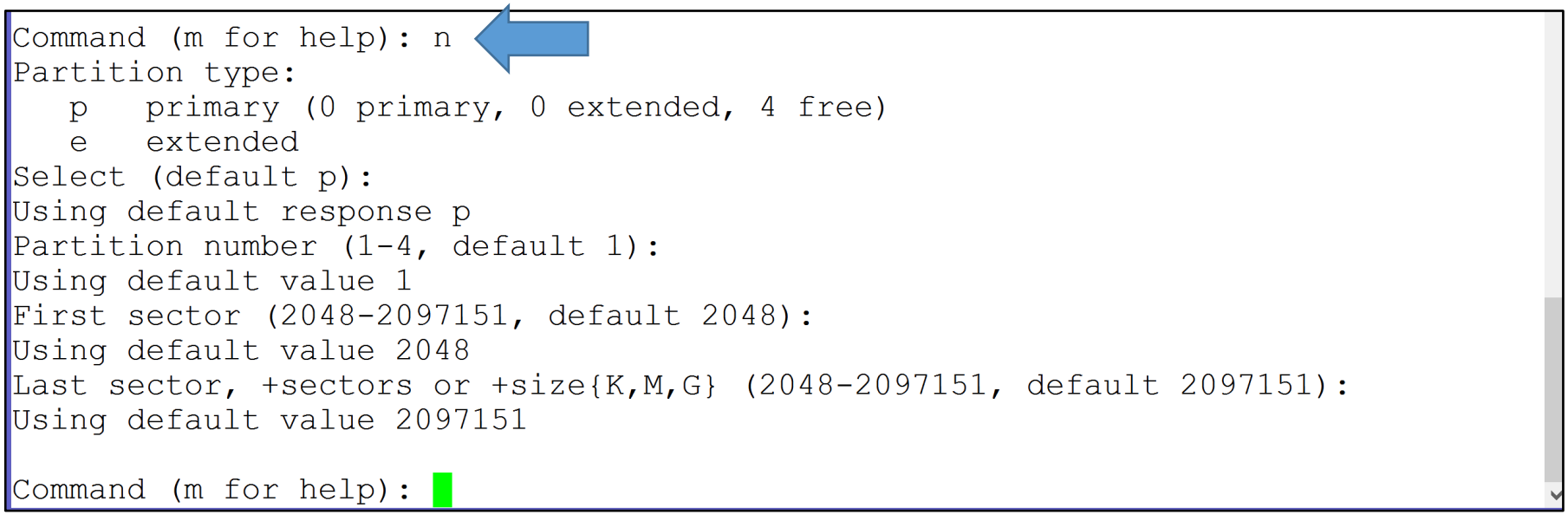
Now below is a very brief comparison of the most common file systems in use with the Linux world.

File System	Max File Size	Max Partition Size	Journaling	Notes
Fat16	2 GB	2 GB	No	Legacy
Fat32	4 GB	8 TB	No	Legacy
NTFS	2 TB	256 TB	Yes	(For Windows Compatibility) NTFS-3g is installed by default in Ubuntu, allowing Read/Write support
ext2	2 TB	32 TB	No	Legacy
ext3	2 TB	32 TB	Yes	Standard linux filesystem for many years. Best choice for super-standard installation.
ext4	16 TB	1 EiB	Yes	Modern iteration of ext3. Best choice for new installations where super-standard isn't necessary.
reiserFS	8 TB	16 TB	Yes	No longer well-maintained.
JFS	4PB	32PB	Yes (metadata)	Created by IBM - Not well maintained.
XFS	8 EB	8 EB	Yes (metadata)	Created by SGI. Best choice for a mix of stability and advanced journaling.
GB = Gigabyte (1024 MB) :: TB = Terabyte (1024 GB) :: PB = Petabyte (1024 TB) :: EB = Exabyte (1024 PB)				

Preparing a disk - step 1: create the partitions

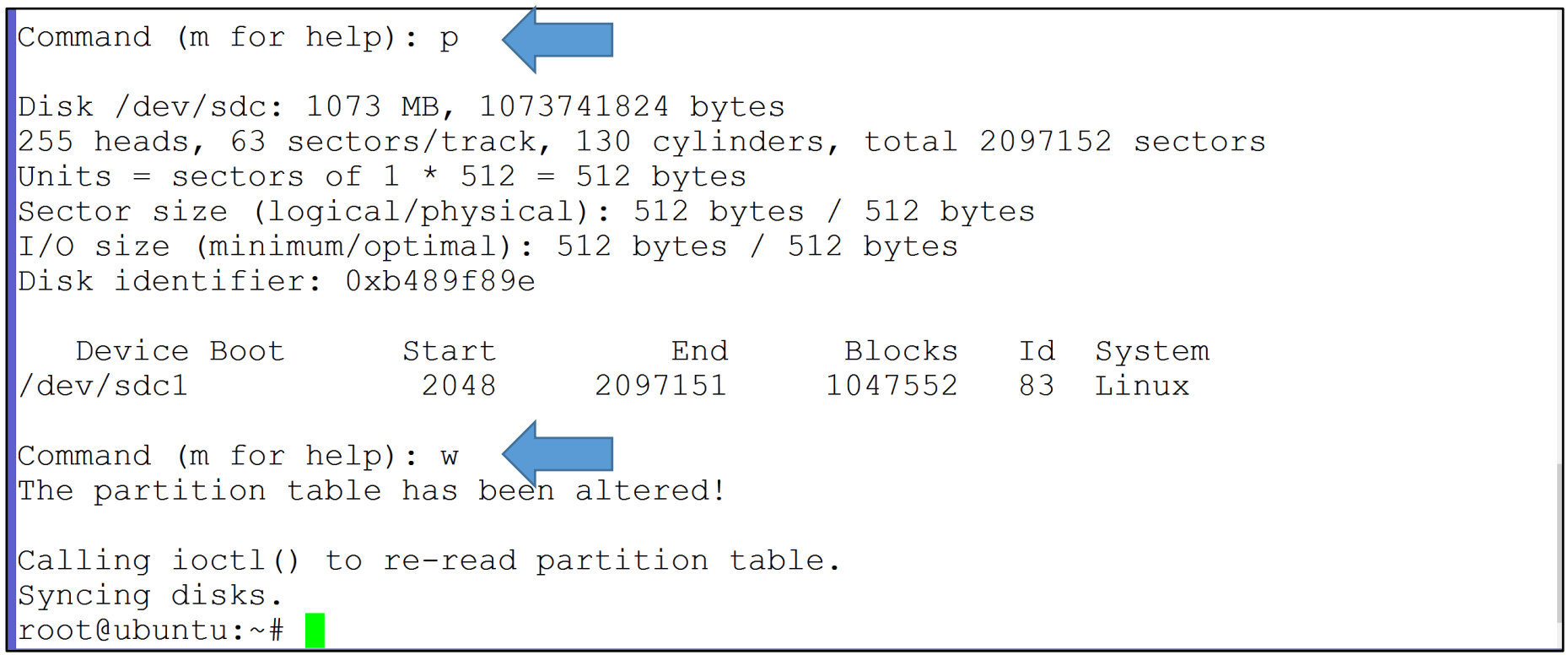
```
# fdisk /dev/sdc
```

m for help
n for new partition



Preparing a disk - step 2: writing the partition map

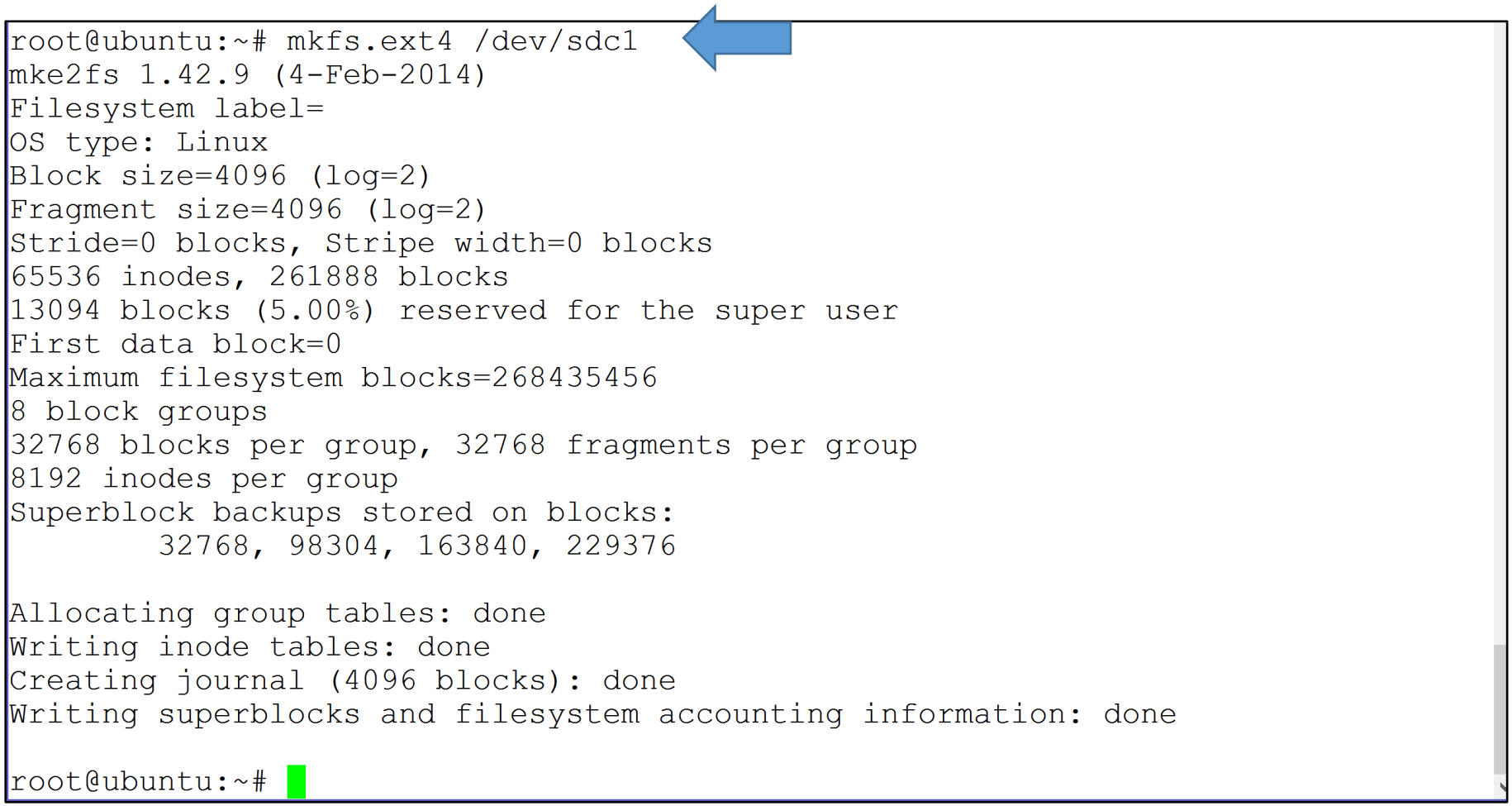
p to print - always print before writing!
w to write



```
Command (m for help): p

Disk /dev/sdc: 1073 MB, 1073741824 bytes
255 heads, 63 sectors/track, 130 cylinders, total 2097152 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: 0xb489f89e

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1            2048     2097151     1047552   83  Linux

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
root@ubuntu:~#
```

Preparing a disk - step 3: add the filesystem

```
# mkfs.ext4 /dev/sdc1    or:
# mkfs -t ext4 /dev/sdc1
```



Outline

- Filesystems as a database
- Links
- Partitions
- gparted demo
- mount & fstab
- Other partition considerations: `du`

Mounting Filesystems

- Multiple (physical) file systems can be combined into a single logical filesystem.
- The 'root' of the new physical filesystem is grafted onto some directory of the overall logical filesystem.
- Filesystems are joined at **mount points**
- A mount point...
  - ... is a directory.
  - ... should be empty.
  - ... needs to exist before the mount command is issued.

Mounting Other Filesystems

- Filesystems can be mounted at any time.
- The `mount` command is used

```
# mount <what> <where>
# mount /dev/sda2 /mnt/oldPrat
```

- You may occasionally need to specify the filesystem type explicitly:

```
# mount -t vfat /dev/hdd1 /mnt/windows
```

- To see a list of the filesystems currently mounted, run mount with no options

```
$ mount
```

Filesystem Types

-t, --types *vfstype*

The argument following the **-t** is used to indicate the filesystem type. The filesystem types which are currently supported include: *adfs*, *affs*, *autofs*, *cifs*, *coda*, *coherent*, *cramfs*, *debugfs*, *devpts*, *efs*, *ext*, *ext2*, *ext3*, *ext4*, *hfs*, *hfsplus*, *hpfs*, *iso9660*, *jfs*, *minix*, *msdos*, *ncpfs*, *nfs*, *nfs4*, *ntfs*, *proc*, *qnx4*, *ramfs*, *reiserfs*, *romfs*, *squashfs*, *smbfs*, *sysv*, *tmpfs*, *ubifs*, *udf*, *ufs*, *umsdos*, *usbfs*, *vfat*, *xenix*, *xfs*, *xiafs*. Note that coherent, sysv and xenix are equivalent and that *xenix* and *coherent* will be removed at some point in the future - use *sysv* instead. Since kernel version 2.1.21 the types *ext* and *xiafs* do not exist anymore. Earlier, *usbfs* was known as *usbdevfs*. Note, the real list of all supported filesystems depends on your kernel.

- Others:
  - smbfs
  - CIFS — successor to samba
  - nfs
  - sshfs

Unmounting a Filesystem

- A filesystem can be removed from the filesystem with the `umount command`
- Two ways:

  `# umount /mnt/extra`

  unmounts whatever is on the /mnt/extra mount point.

  `# umount /dev/sdb3`

  unmounts the filesystem in the /dev/sdb3 device, wherever it is mounted.

- You need to have root permission

Busy Filesystems

- To unmount a filesystem, it can't be busy:

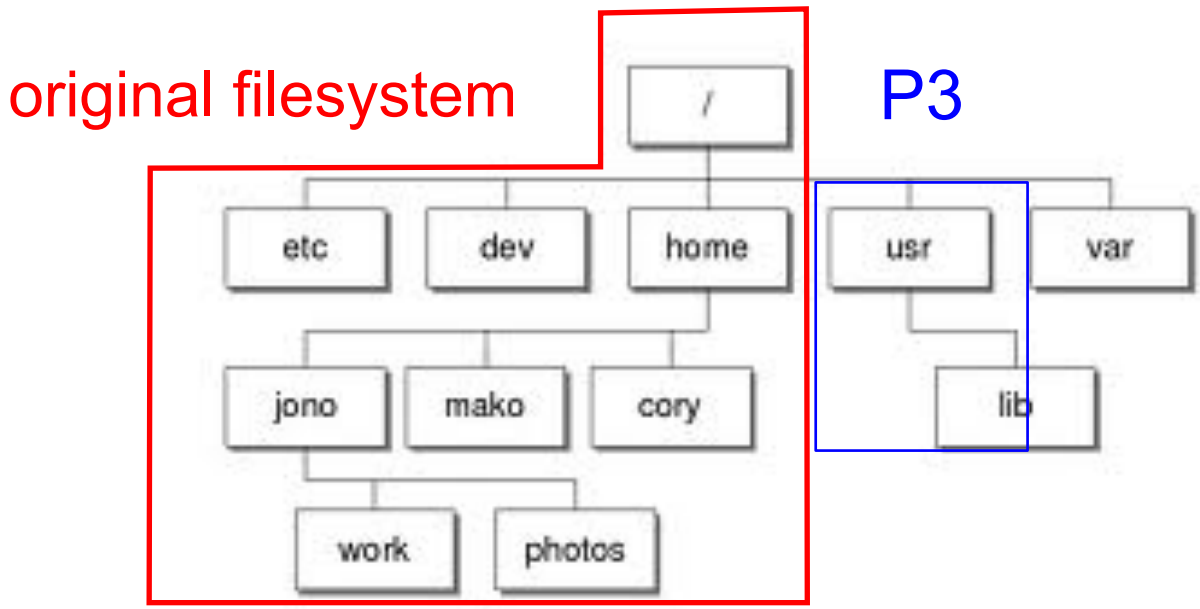
  A filesystem is busy if a running process has a file on it open or

  if a process has a directory within it as its current directory

gparted

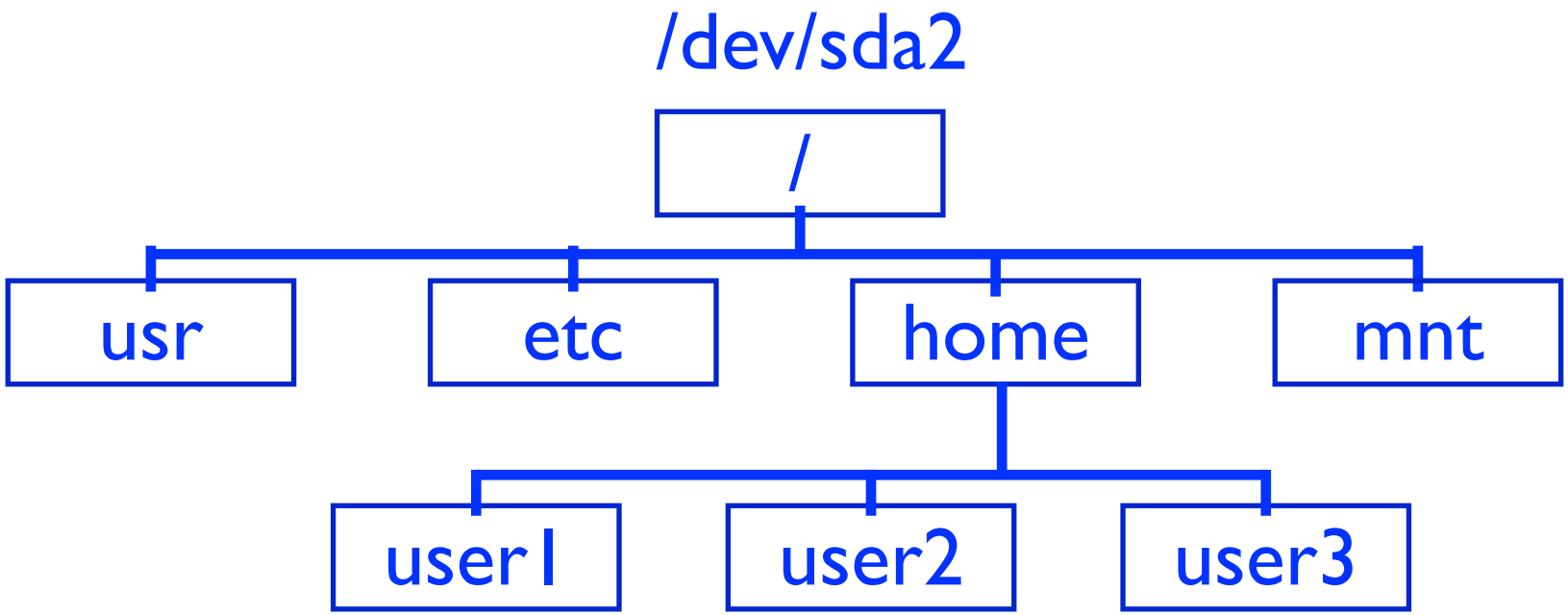
demo time!

Mounting Filesystems

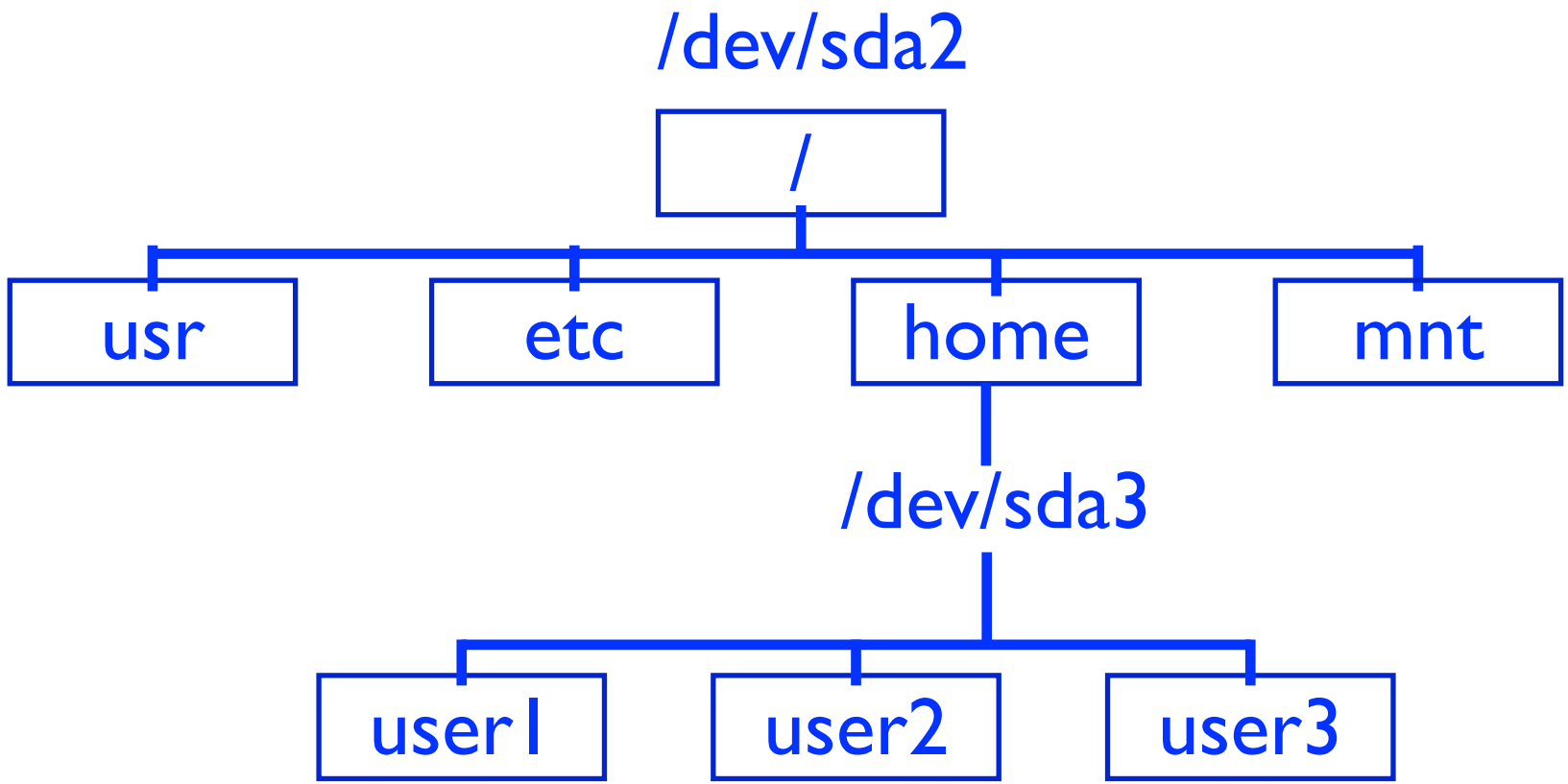


Original fs is mounted as /
P3 is mounted as /usr inside the original fs

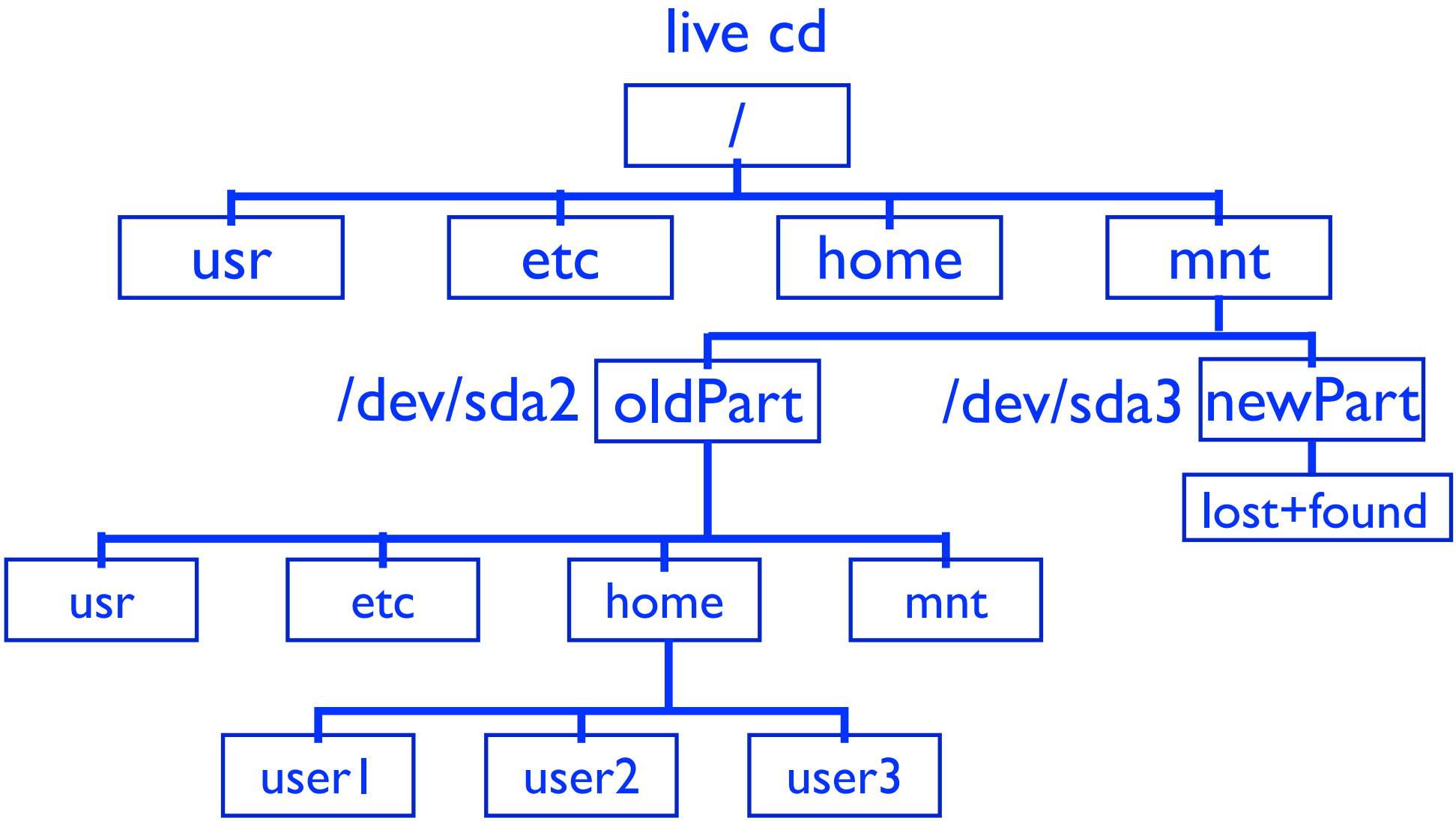
Filesystem - Initial State



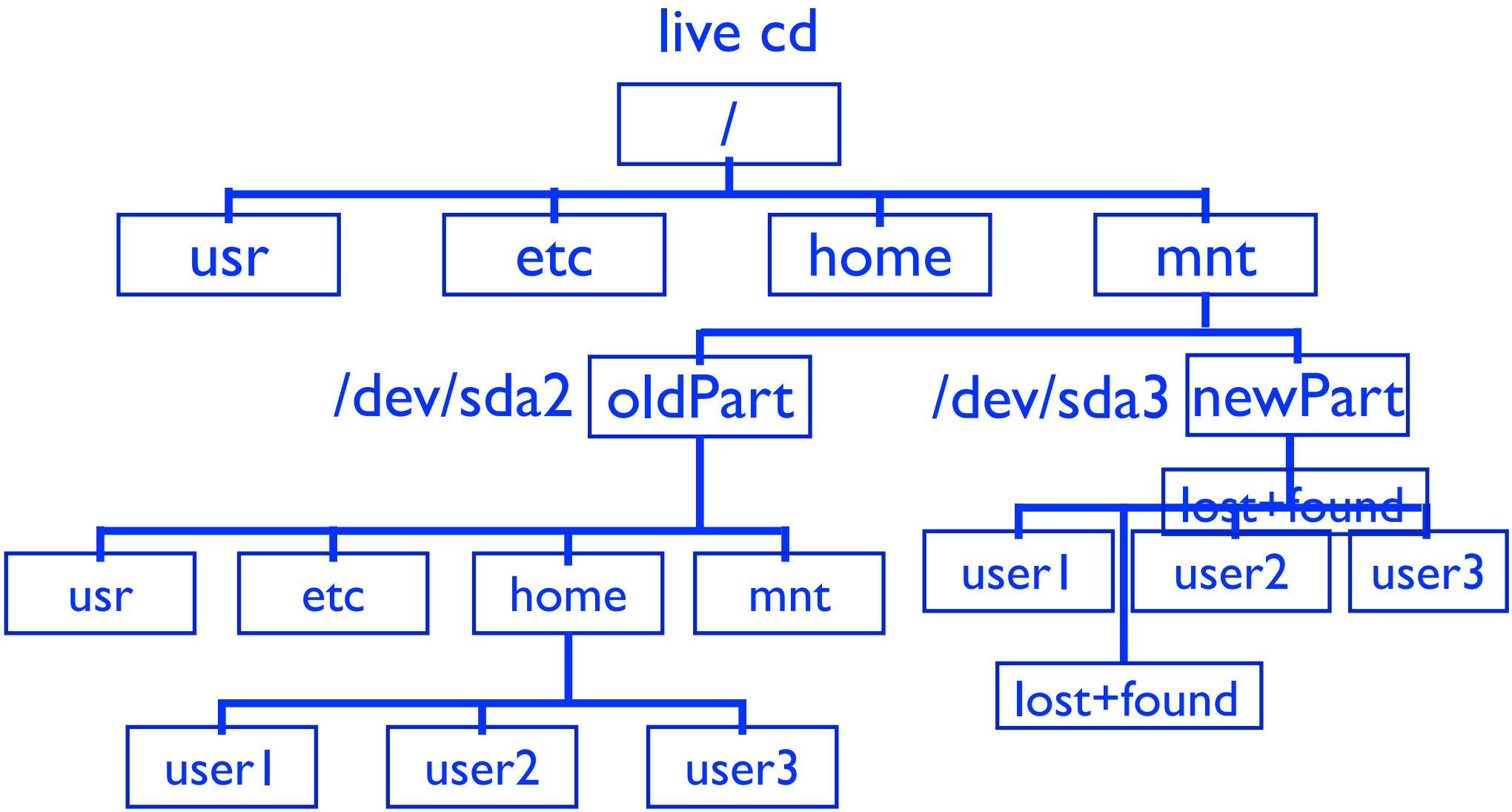
Filesystem - Final State



Gparted with Partitions Mounted



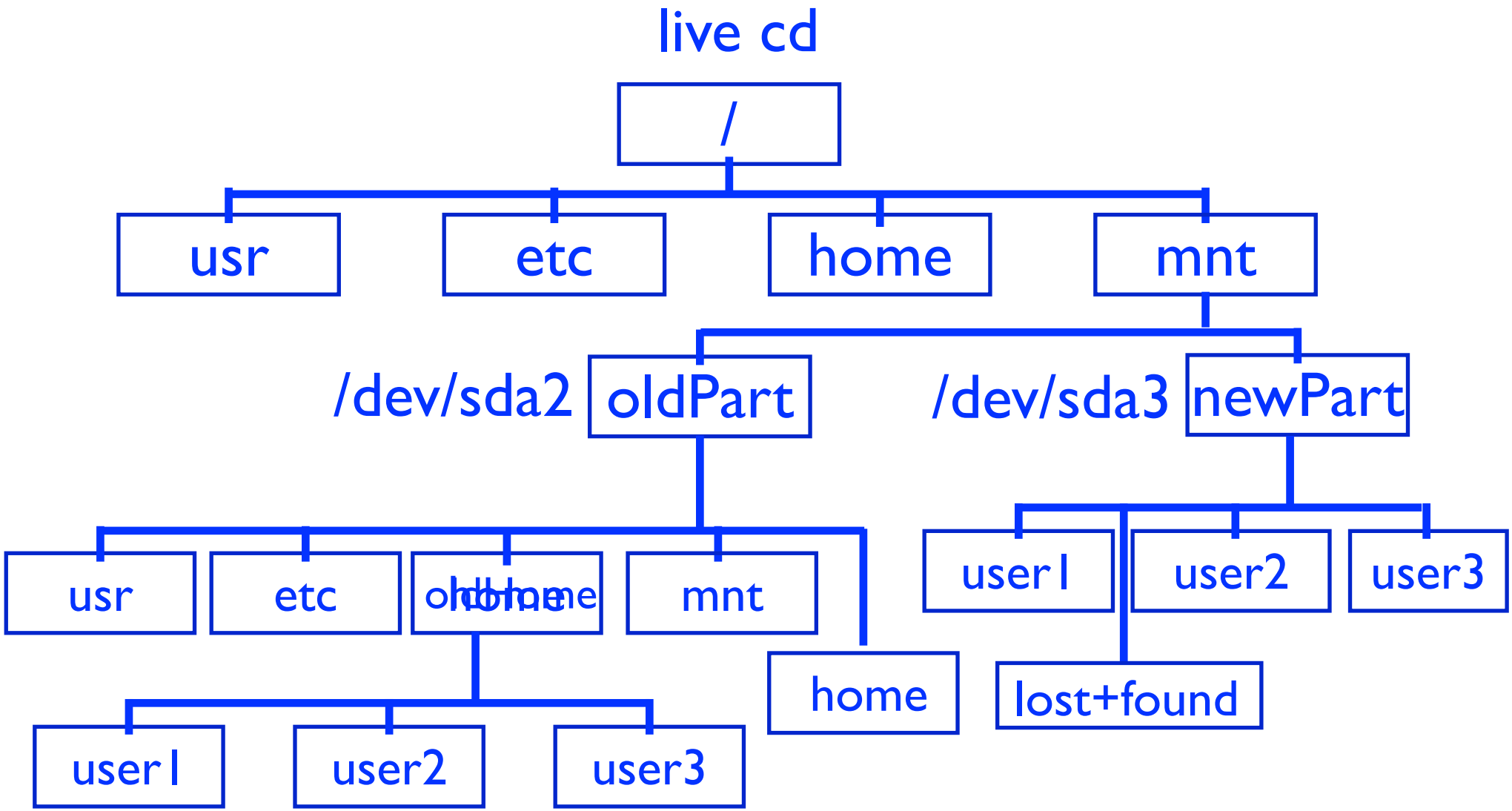
Gparted Filesystem after copy of Home



Making Changes Permanent

- So far, changes are not permanent
  - only within gparted, which doesn't save state
  - if rebooted, everything unmounted
- Need to make changes to FS on /dev/sda2
  - want to always mount /dev/sda3
  - mount point should be /home on /dev/sda2
- But first…
  - need to get rid of existing contents of /home

Gparted Filesystem after rename of home



Configuring mount: `/etc/fstab`

- The `/etc/fstab` file contains information about known filesystems
- Specifying a filesystem in `/etc/fstab` makes it possible to use its mount point as the only argument to mount
- /etc/fstab also configures which fs should be mounted at boot time
- Each line describes one filesystem
- Six columns per line

Sample /etc/fstab

- A sample `/etc/fstab` file:

```
# device mount-point type options (dump) pass-no
/dev/hda3   /           ext2      defaults 1 1
/dev/hda1   /boot       ext2      defaults 1 2
/dev/hda5   /usr        ext2      defaults 1 2
/dev/hdb1   /usr/local  ext2      defaults 1 2
/dev/hdb2   /home       ext2      defaults 1 2
none        /proc       proc      defaults 0 0
/dev/scd0   /mnt/cdrom  iso9660   noauto,users,ro 0 0
/dev/fd0    /mnt/floppy auto      noauto,users 0 0
```

first 3 columns should make sense

Ubuntu fstab format - mount by uuid

```
$ cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' 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=f28a8e35-e451-43a1-82ce-64f6f2634d28 / ext4 errors=remount-ro 0  1

# /home was on /dev/sda5 during installation
UUID=48f7db11-727f-4f76-95ec-5590df2d45c9 /home ext4 defaults,user_xattr   0  2

# swap was on /dev/sda3 during installation
UUID=98427a23-e66b-4967-9ad3-0bade26af54e none    swap    sw   0   0

# /data is on /dev/sda6
UUID=23c4cc02-e0cc-4c64-818d-8f0fa3936208 /data   ext4  defaults 0   0

# / (Lucid) is on /dev/sda7
UUID=cefd3fc4-7272-4b1f-b45e-46079fa75eaf /lucid  ext4    defaults  0   1
```

fstab mount options

- Comma-separated options in /etc/fstab
- Alternatively, use comma-separated options with -o on the mount command line
- no need to memorize, look them up as needed

Option	Description
`noauto`	In */etc/fstab*, prevents the filesystem being mounted at bootup. Useful for removable media
`ro`	Mount the filesystem read-only
`users`	Let non-root users mount and unmount this filesystem
`user`	Like `users`, but non-root users can only unmount filesystems that they themselves mounted

last 2 columns in /etc/fstab

- The fifth column is called dump
  - used by the dump and restore backup utilities
  - no longer commonly used
  - just use 1 for filesystems you'd typically backup, and 0 for others
- last column is pass-no field
  - controls the order in which automatically-mounted filesystems are checked by fsck
  - use 1 for the root filesystem
  - use 0 for filesystems that aren't mounted at boot-up
  - use 2 for other filesystems

System administration fs concerns

- 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 like these

Monitoring Space: df

- Run df with no arguments to get a listing of free space on all mounted filesystems
- -h gives human readable sizes

```
$ df -h
Filesystem    Size  Used  Avail  Use% Mounted on
/dev/hda8     248M  52M    183M  22%  /
/dev/hda1      15M  5.6M   9.1M  38%  /boot
/dev/hda6      13G  5.0G   7.4G  41%  /home
/dev/hda5      13G  4.6G   7.8G  37%  /usr
/dev/hda7     248M  125M   110M  53%  /var
```

Monitoring Space: `df`

can use `df -h <directory>` to see just the space information for that partition

```
$ df -h /usr
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       9.8G  4.5G  4.8G  49% /
```

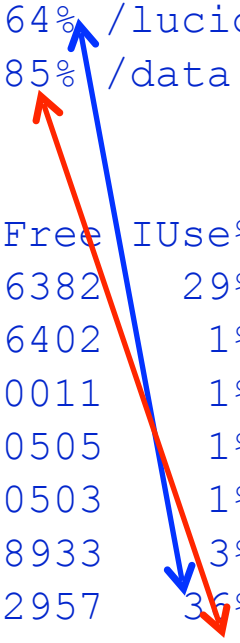
Monitoring Inodes: `df -i`

- if a fs has lots of small files, could run out of inodes (rare)
- Run `df -i` to get information on inode usage on all mounted filesystems:

```
$ df -i
Filesystem  Inodes   IUsed   IFree IUse% Mounted
on
/dev/hda8    65736    8411   57325   13%  /
/dev/hda1     4160      30    4130    1%  /boot
/dev/hda6  1733312  169727 1563585   10%  /home
/dev/hda5  1733312  138626 1594686    8%  /usr
/dev/hda7    65736    1324   64412    2%  /var
```

What can you tell about the types of files in /lucid and /data?

```
laptop:~$ df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda1              13G  6.2G  5.7G  53% /
udev                  2.0G  4.0K  2.0G   1% /dev
tmpfs                 785M  940K  784M   1% /run
none                  5.0M   24K  5.0M   1% /run/lock
none                  2.0G  208K  2.0G   1% /run/shm
/dev/sda5              35G   19G   15G  56% /home
/dev/sda7              12G  7.0G  4.1G  64% /lucid
/dev/sda6              83G   67G   12G  85% /data

laptop:~$ df -i
Filesystem            Inodes   IUsed   IFree IUse% Mounted on
/dev/sda1             829056  232674  596382   29% /
udev                  207009     607  206402    1% /dev
tmpfs                 210518     507  210011    1% /run
none                  210518      13  210505    1% /run/lock
none                  210518      15  210503    1% /run/shm
/dev/sda5            2293760   64827 2228933    3% /home
/dev/sda7             768544  275587  492957   36% /lucid
/dev/sda6            5488640   45842 5442798    1% /data
```



Monitoring Disk Usage: du

- df shows free space on a partition
- du, shows disk space used in a directory tree
- Takes directory or directories as argument
- Popular options:

```
$ du -shc *
   44Masics
   25Mdiagtools
  146Mespresso
  520MOban
    8MP7P
   30MPackaging
   61Msas
   29Mserver
  1.9Gvejle
    9Mx1
  2.8Gtotal
```

du Options

- common options:
  - -a Show all files, not just directories
  - -c Print a cumulative total for all directories named on the command line
  - -h Print disk usage in human-readable units
  - -s Print only a summary for each directory named on the command line
  - -S Make the size reported for a directory be the size of only the files in that directory, not the total including the sizes of its subdirectories