



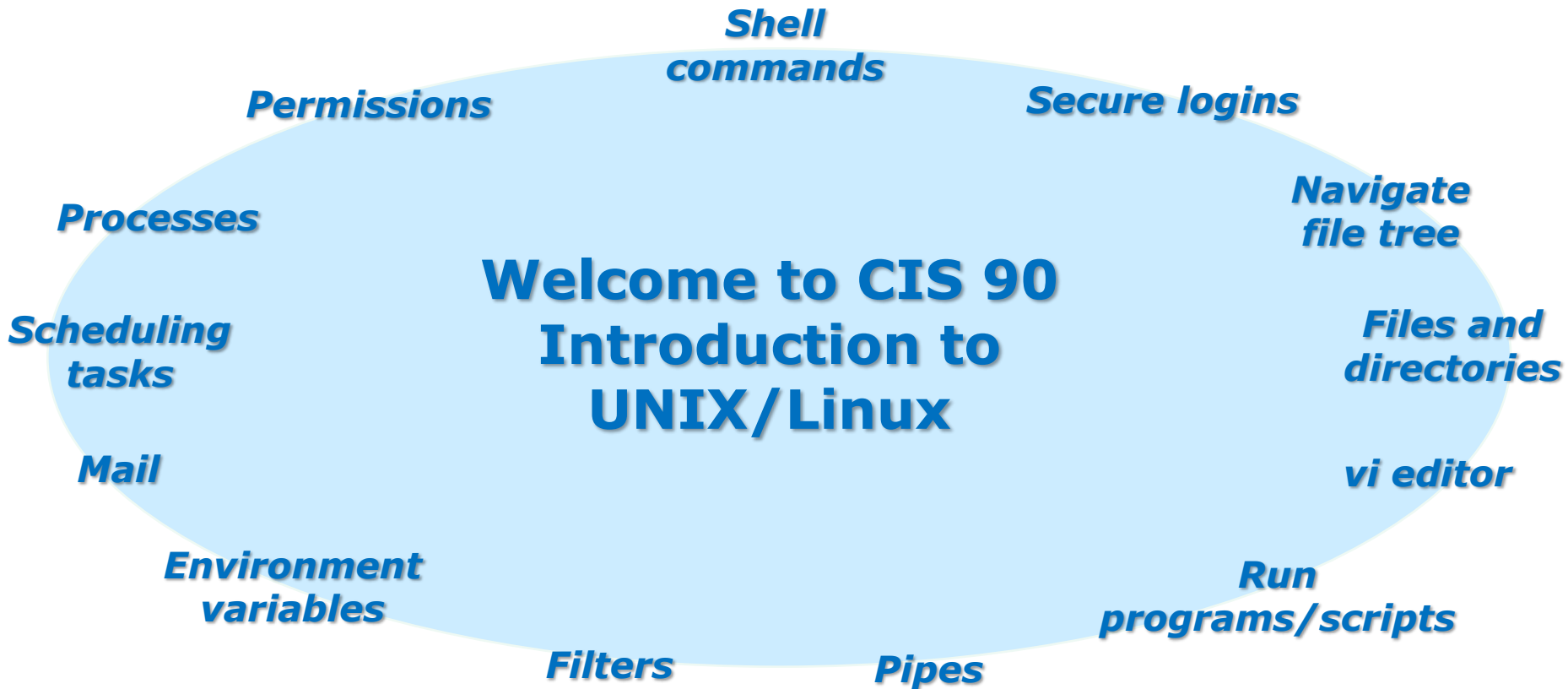
Rich's lesson module checklist

- Slides
- WB pre-generated

- Flash cards
- Page numbers
- 1st minute quiz
- Web Calendar summary
- Web book pages
- Commands

- Lab 4 tested
- feedbot activated for use by check4
- Schedule lock of turnin directory and submit
 - at 12:00 am Thursday
chmod 700 /home/cis90/bin/submit
chmod 700 /home/turnin/cis90
ctrl-d
 - at 9:00 am thursday
chmod 750 /home/cis90/bin/submit
chmod 755 /home/turnin/cis90
ctrl-d
- Enlightenment script tested

- 9V backup battery for microphone
- Backup slides, CCC info, handouts on flash drive



Student Learner Outcomes

1. Navigate and manage the UNIX/Linux file system by viewing, copying, moving, renaming, creating, and removing files and directories.
2. Use the UNIX features of file redirection and pipelines to control the flow of data to and from various commands.
3. With the aid of online manual pages, execute UNIX system commands from either a keyboard or a shell script using correct command syntax.

Introductions and Credits



Jim Griffin

- Created this Linux course
- Created Opus and the CIS VLab
- Jim's site: <http://cabrillo.edu/~jgriffin/>



Rich Simms

- HP Alumnus
- Started teaching this course in 2008 when Jim went on sabbatical
- Rich's site: <http://simms-teach.com>

And thanks to:

- John Govsky for many teaching best practices: e.g. the First Minute quizzes, the online forum, and the point grading system (<http://teacherjohn.com/>)



Student checklist for laying out screen when attending class

- Browse to the CIS 90 website Calendar page
 1. <http://simms-teach.com>
 2. Click CIS 90 link on left panel
 3. Click Calendar link near top of content area
 4. Locate today's lesson on the Calendar

- Download the presentation slides for today's lesson for easier viewing

- Click Enter virtual classroom to join CCC Confer session

- Connect to Opus using Putty or ssh command



Student checklist for laying out screen when attending class

Google

CCC Confer

Downloaded PDF of Lesson Slides

The screenshot shows a virtual classroom interface with several overlapping windows:

- Blackboard Course Page:** Displays 'Rich's Cabrillo College CIS 90 Calendar' with a sidebar containing navigation links like 'Login', 'Flashcards', 'Admin', and 'CIS 90 (Spring) Course Home'.
- CCC Confer Virtual Classroom:** The main window showing a video feed of 'Rich Simms' and a 'PARTICIPANTS' list with 'Rich Simms' and 'Benji Simms'. A 'CHAT' window shows a conversation about textbooks.
- Google Maps:** A window titled 'Class Activity - Where are you now?' displaying a map of the San Francisco Bay Area.
- Adobe Acrobat Pro:** A window titled 'cis90lesson01.pdf' showing a slide titled 'The CIS 90 System Playground' with a diagram of server racks.
- Terminal Window:** A window showing a login prompt for 'Opus' with a password field and a 'Welcome to Opus' message.

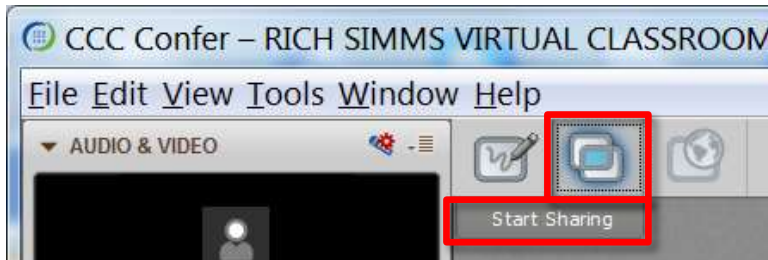
CIS 90 website Calendar page

One or more login sessions to Opus

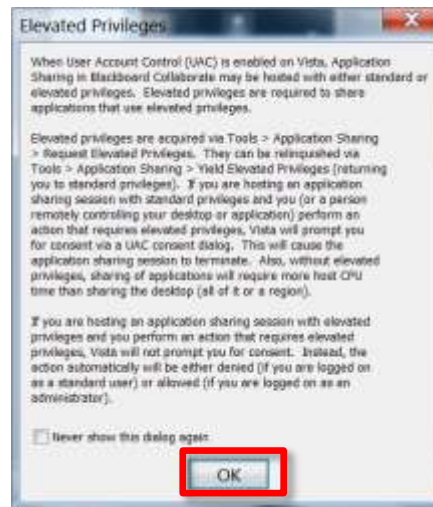


Student checklist for sharing desktop with classmates

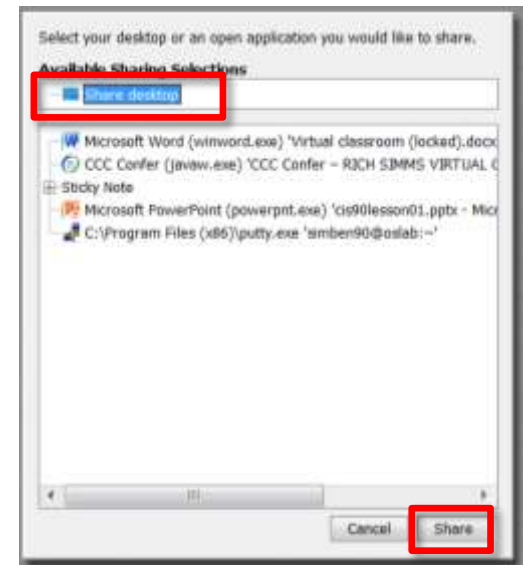
1) Instructor gives you sharing privileges



2) Click overlapping rectangles icon. If white "Start Sharing" text is present then click it as well.



3) Click OK button.



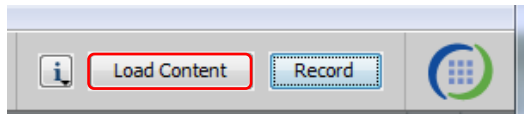
4) Select "Share desktop" and click Share button.



Rich's CCC Confer checklist - setup

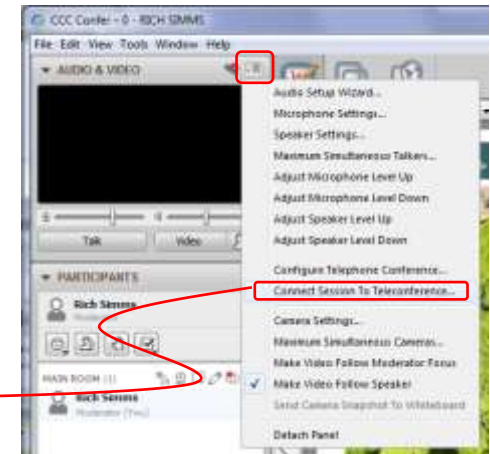
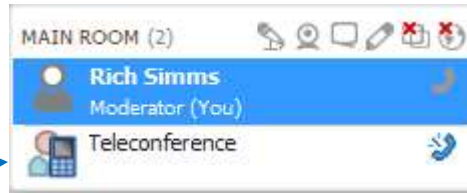


[] Preload White Board



[] Connect session to Teleconference

Session now connected to teleconference



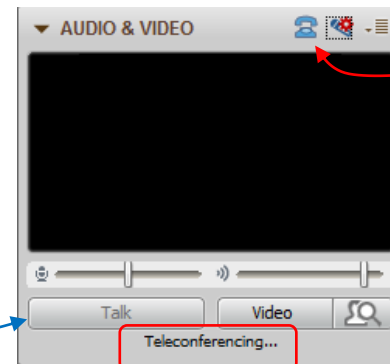
[] Is recording on?



Red dot means recording

[] Use teleconferencing, not mic

Should be greyed out



Should show as this live "off hook" telephone handset icon and the Teleconferencing ... message displayed



Rich's CCC Confer checklist - screen layout and share

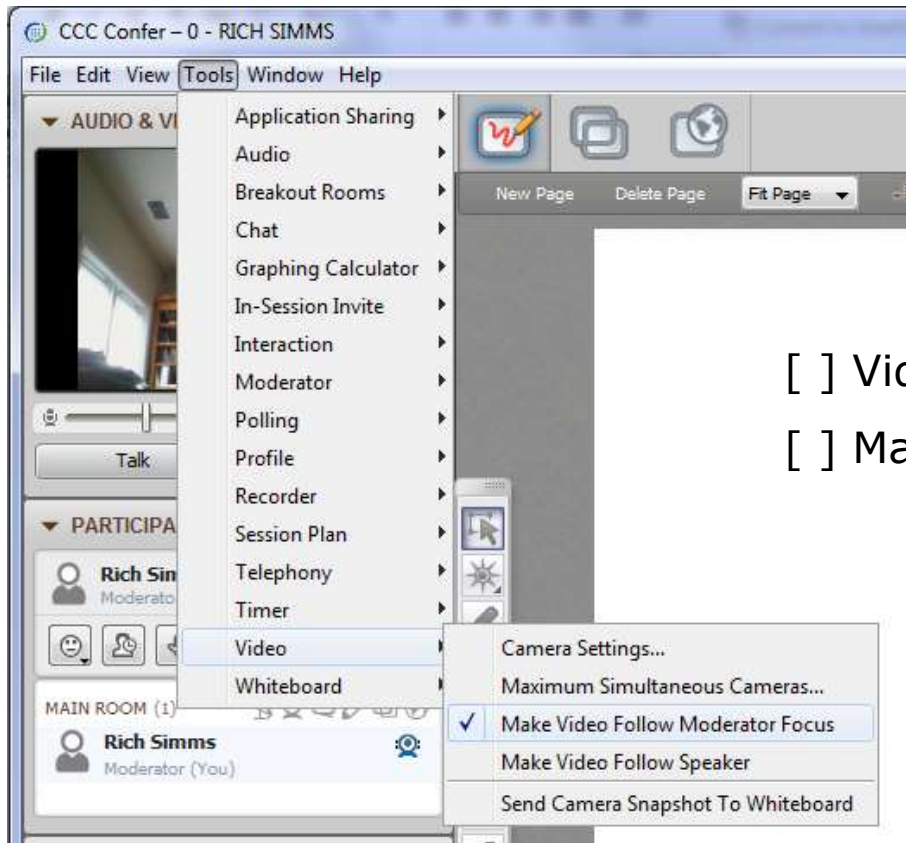


[] layout and share apps





Rich's CCC Confer checklist - webcam setup



[] Video (webcam)

[] Make Video Follow Moderator Focus



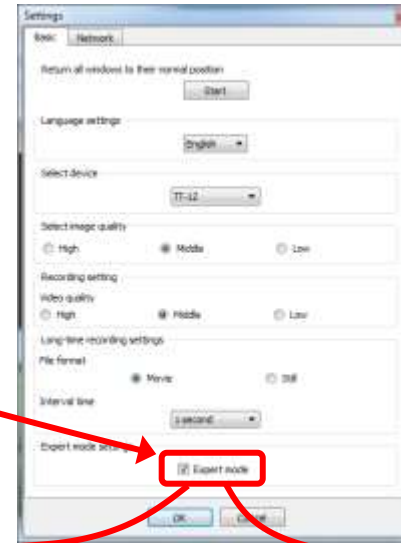
Rich's CCC Confer checklist - Elmo



Elmo rotated down to view side table



Run and share the Image Mate program just as you would any other app with CCC Confer



The "rotate image" button is necessary if you use both the side table and the white board.

Quite interesting that they consider you to be an "expert" in order to use this button!

Elmo rotated up to view white board





Rich's CCC Confer checklist - universal fix

Universal Fix for CCC Confer:

- 1) Shrink (500 MB) and delete Java cache
- 2) Uninstall and reinstall latest Java runtime
- 3) <http://www.cccconfer.org/support/technicalSupport.aspx>

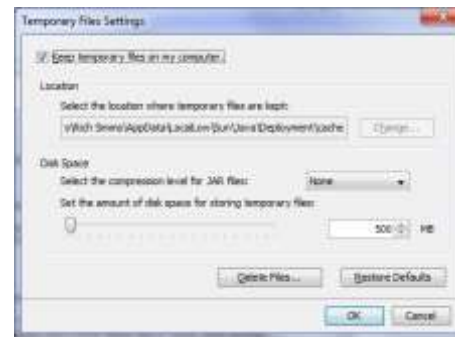
Control Panel (small icons)



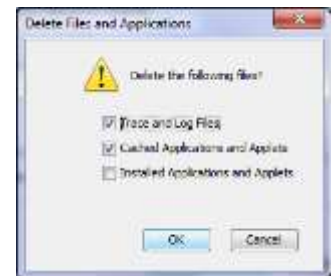
General Tab > Settings...



500MB cache size



Delete these



Google Java download





Start



Sound Check

*Students that dial-in should mute their line using *6 to prevent unintended noises distracting the web conference.*

*Instructor can use *96 to mute all student lines.*



Instructor: **Rich Simms**

Dial-in: **888-886-3951**

Passcode: **136690**



Chris



Jeremy



Jennifer



Cameron



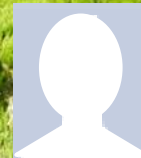
Joseph



Lisa



May



Sundance



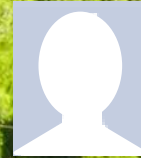
Charlie



Sean



Brenda



Anthony



Will H.



Josh



Michael



Danny



Vic



William D.



Taylor



Thomas



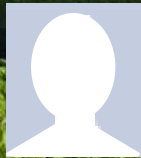
Stewart



Miguel



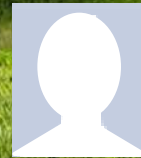
Akasha



Jairo



Tony



Joaquin

First Minute Quiz

Please answer these questions **in the order** shown:

Use CCC Confer White Board

email answers to: risimms@cabrillo.edu

(answers must be emailed within the first few minutes of class for credit)

The UNIX/Linux File System

Objectives

- Become familiar with the UNIX file hierarchy.
- Be able to navigate the hierarchy using `cd`, `ls` and `pwd` commands.
- Understand the key elements of a file.
- Be able to distinguish the different UNIX files types.
- Learn appropriate commands to view file contents.

Agenda

- Quiz
- Questions
- Housekeeping
- The UNIX file tree
- Navigating the file tree
- Unix files
- UNIX filename conventions
- Viewing text files
- Viewing binary files
- Basic file types
- Further classification of files
- Pathnames
- Absolute pathnames
- Relative pathnames
- `/` `..` and `~` directories
- Shell tips
- Using pathnames as arguments
- More on `cd`, `pwd` and `ls` commands
- Home directories
- Filename expansion with `*`
- The path to enlightenment
- Assignment and wrap up



Questions

Questions?

Lesson material?

Labs? Tests?

How this course works?

- Graded work in home directories
- Answers in /home/cis90/answers

Who questions much, shall learn much, and retain much.

- Francis Bacon

If you don't ask, you don't get.

- Mahatma Gandhi

Chinese
Proverb

他問一個問題，五分鐘是個傻子，他不問一個問題仍然是一個傻瓜永遠。

He who asks a question is a fool for five minutes; he who does not ask a question remains a fool forever.

Got stuck or having trouble getting started in this course?



If you would like some additional come over to the CIS Lab. There are student lab assistants and instructors there to help you.

*Tess, Michael, and Paul are
CIS 90 Alumni.*

*Mike Matera is the other
Linux instructor.*

*I'm in there Mondays
11:00-1:30pm.*

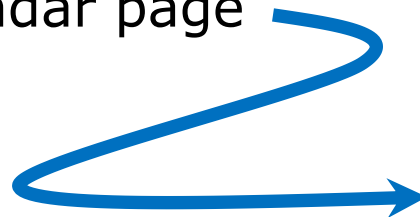


Housekeeping





- Lab 3 due tonight at 11:59PM (Opus time)
 - Use **mail -f uhistory** and **check3** to review your collection
 - Clean up duplicates before last submittal
 - I'll grade using a variation of **check3** script
 - Don't forget to use **submit** to turn in your work!
- Five forum posts due tonight at 11:59PM (Opus time)
- Reminder - all quizzes, all tests, all due dates for all work is on the website Calendar page



Linux Certifications

Red Hat / Linux Professional Institute (LPI) / Linux Foundation

Linux Professional Institute (LPI) certifications

- **Linux Essentials** The Linux Essentials Professional Development Certificate (PDC) is a great way to show employers that you have the foundational skills required for your next job or promotion. It also serves as an ideal stepping-stone to the more advanced LPIC Professional Certification track for Linux Systems Administrators.
 - [60 minute exam](#) at [PearsonVue](#) test center
- **LPIC-1** is a junior level certification for Linux administrators. You should be able to perform maintenance tasks with the command line, install & configure a workstation and be able to configure a basic network.
 - [LX0-101](#) exam CompTIA Linux+ Powered by LPI
 - [LX0-102](#) exam CompTIA Linux+ Powered by LPI
- **LPIC-2** is aimed at advanced Linux professionals. To be awarded LPIC level 2 you should be able administer small to medium sized mixed networks and provide suggestions to upper management.
 - [LX0-103](#) exam CompTIA Linux+ Powered by LPI
 - [LX0-104](#) exam CompTIA Linux+ Powered by LPI
- **LPIC-3** is designed for senior-level Linux professionals in an enterprise environment. You should be able to concept, architect, install and troubleshoot LDAP software and integrate with Active Directory.
- LPI Certification [Mapping Matrix](#) to Cabrillo College Linux classes

LPI Linux Essentials Certificate

Linux Essentials Certificate of Achievement				
Objective	# of Questions	Cabrillo	Urban Penguin	NDG Linux Essentials
Topic 1: The Linux Community and a Career in Open Source				
1.1 Linux Evolution and Popular Operating Systems	2	CIS90 Lesson 1	1.1	Module 1
1.2 Major Open Source Applications	2	CIS90 Lesson 1	1.2	Module 2
1.3 Understanding Open Source Software and Licensing	1	CIS90 Lesson 1	1.3	Module 2
1.4 ICT Skills and Working in Linux	2	not covered	1.4	Module 3
Topic 2: Finding Your Way on a Linux System				
2.1 Command Line Basics	2	CIS90 Lesson 2	2.1	Module 4
2.2 Using the Command Line to Get Help	2	CIS90 Lesson 2	2.2	Module 5
2.3 Using Directories and Listing Files	2	CIS 90 Lesson 4	2.3	Module 6
2.4 Creating, Moving and Deleting Files	2	CIS90 Lesson 5	2.4	Module 6
Topic 3: The Power of the Command Line				
3.1 Archiving Files on the Command Line	2	CIS 90 Lesson 14	3.1	Module 7
3.2 Searching and Extracting Data from Files	4	CIS 90 Lesson 8	3.2	Module 8
3.3 Turning Commands into a Script	4	CIS 90 Lesson 13 & 14	3.3	Module 9
Topic 4: The Linux Operating System				
4.1 Choosing an Operating System	1	not covered	4.1	Module 1
4.2 Understanding Computer Hardware	2	CIS 90 Lesson 1	4.2	Module 10
4.3 Where Data is Stored	3	CIS 90 Lesson 1	4.3	Module 11
4.4 Your Computer on the Network	2	CIS 192	4.4	Module 12
Topic 5: Security and File Permissions				
5.1 Basic Security and Identifying User Types	2	CIS 191	5.1	Module 13
5.2 Creating Users and Groups	2	CIS 191	5.2	Module 14
5.3 Managing File Permissions and Ownership	2	CIS 90 Lesson 7	5.3	Module 15
5.4 Special Directories and Files	1	CIS 90 Lesson 4	5.4	Module 16

The Urban Penguin

Home LPI

LINUX ESSENTIALS

Welcome to this self study video series of tutorials. These videos can be used in preparing you for the LPI/Linux Professional Institute, Linux Essentials, Certification. These materials are meant as a stand alone learning solution in readiness for your exam and are targeted towards anyone who is aiming for the certification or just wants to know more about what Linux is and what it can offer. The Urban Penguin is an **Approved LPI Training Partner** and we provide both free training via these videos and, if you prefer to assist direct with the penguin, then we can offer **video training** at a reasonable cost.

Objective	Description	Click to Access
000	What is LPI Linux Essentials	Click to Access
1.1	Linux evolution and popular operating systems	Click to Access
1.2	Major Open Source applications	Click to Access
1.3	Understanding Open Source Software and licensing	Click to Access
1.4	ECT, shells and working with Linux	Click to Access
2.1	Command line basics	Click to Access
2.2	Using the command line to get help	Click to Access
2.3	Using directories and listing files	Click to Access
2.4	Creating, moving and deleting	Click to Access
3.1	Archiving files from the command line	Click to Access
3.2	Backup and extracting data from files	Click to Access
3.3	Turning commands into a script	Click to Access
4.1	Choosing an operating system	Click to Access
4.2	Understanding computer hardware	Click to Access
4.3	Where data is stored	Click to Access
4.4	Your computer on the network	Click to Access
5.1	Basic security and user types	Click to Access
5.2	Creating users and groups	Click to Access
5.3	Manage file permissions and ownership	Click to Access
5.4	Special directories and files	Click to Access

Instructor led and live video based Linux Training

<http://www.theurbanpenguin.com/lpi/le.html>

No registration, no logging in, just click and watch the videos

NDG Linux Essentials via Cisco Networking Academy

NDG Linux Essentials

2.3 Major Open Source Applications

The Linux kernel is not a wide variety of software services (many hardware platforms, a computer can act as a server, which means it primarily handles data on other's behalf, or can act as a desktop), which means a user will be interacting with it directly. The machine can run software or it can be used as a development machine in the process of creating software. You can even use multiple sites as there is no restriction to Linux about the size of the machine, it's mainly a matter of configuring which applications run.

One advantage of this is that you can simulate almost all aspects of a production environment from developing, to testing, to deployment on actual server hardware, which saves costs and time. As someone learning Linux, you can run the same server applications on your desktop or inexpensive virtual server that you run on a large internet Service Provider. Of course, you will not be able to handle the volume a large provider results, as they will have much more expensive hardware, but you can simulate almost any configuration without needing powerful hardware or server licensing.

Linux software generally falls into one of three categories:

- Server software** - software that has no direct interaction with the monitor and keyboard of the machine it runs on. Its purpose is to serve information to other computers, called *clients*. Sometimes server software may not talk to other computers but will get all these and "force" it data.
- Desktop software** - a web browser, text editor, music player, or other software that you interact with. In many cases, such as a web browser, the software is talking to a server on the other end and downloading the data for you. While the desktop software is the client.
- Tools** - a broad category of software that tends to include a means to manage your system. You might have a tool that helps you configure your display, or something that provides a Linux shell, or even more sophisticated tools that convert entire code to something that the computer can read.

Additional, we will consider mobile applications, available for the kernel of the Linux.

Previous Next

Test_Support...zip Show all downloads

<https://www.netacad.com/>

Complete course with reading, live VM and tests.

Contact me if you would like a student account for the NDG Linux Essentials course.

Perkins/VTEA Survey

The screenshot shows a forum post on the 'Cabrillo College: Computer and Information Systems' board. The post is titled 'Carl D. Perkins Vocational and Technical Education Act' and is authored by 'Rich Stevens'. The post text explains that the act was originally authorized by Congress in 1964 and is reauthorized every two years. It states that for Cabrillo College to receive a portion of this funding, students in technical classes must fill out a survey. The post includes instructions on how to complete the survey online using WebAdvisor, including logging in, selecting 'STUDENTS: Click Here', and navigating to the 'Career Technical Information' section. It also provides instructions on how to answer questions and submit the survey.

<http://oslab.cis.cabrillo.edu/forum/viewtopic.php?f=114&t=3863>

This is an important source of funding for Cabrillo College.

*Send me an email stating you completed this survey for **three points extra credit!***

The screenshot shows a survey form titled 'Career Technical Information'. The form asks several questions with 'Yes' and 'No' radio button options. The questions are:

- Are you currently receiving benefits from:
 - TAFICALWORKS:
 - UI (Supplemental Security Income)
 - GA (General Assistance)
- Does your [SSN](#) qualify you for a tax waiver?
- Are you a single parent with custody of one or more minor children?
- Are you a [dependent care](#) allowing Cabrillo to receive job aids?
- Have you moved in the preceding 30 months to attend, or to accompany parents or spouses to attend, temporary or seasonal employment or agricultural, dairy, or fishing?

The screenshot shows a web browser window with the URL `opus.cabrillo.edu/forum/viewforum.php?f=25`. The forum header includes the phpBB logo and the text "Cabrillo College: Computer and Information Systems". A search bar is visible in the top right. The main content area is a forum listing for "CIS 90". A red rectangular box is overlaid on the page, containing the following instructions:

- 1st five post deadline is 11:59PM tonight Opus time! (worth 20 points)
- Only your posts in the **CIS 90** forum will earn points (**not** the Practice forum or other classes)
- Your username must be your **full first** and **last** name to get credit on posts

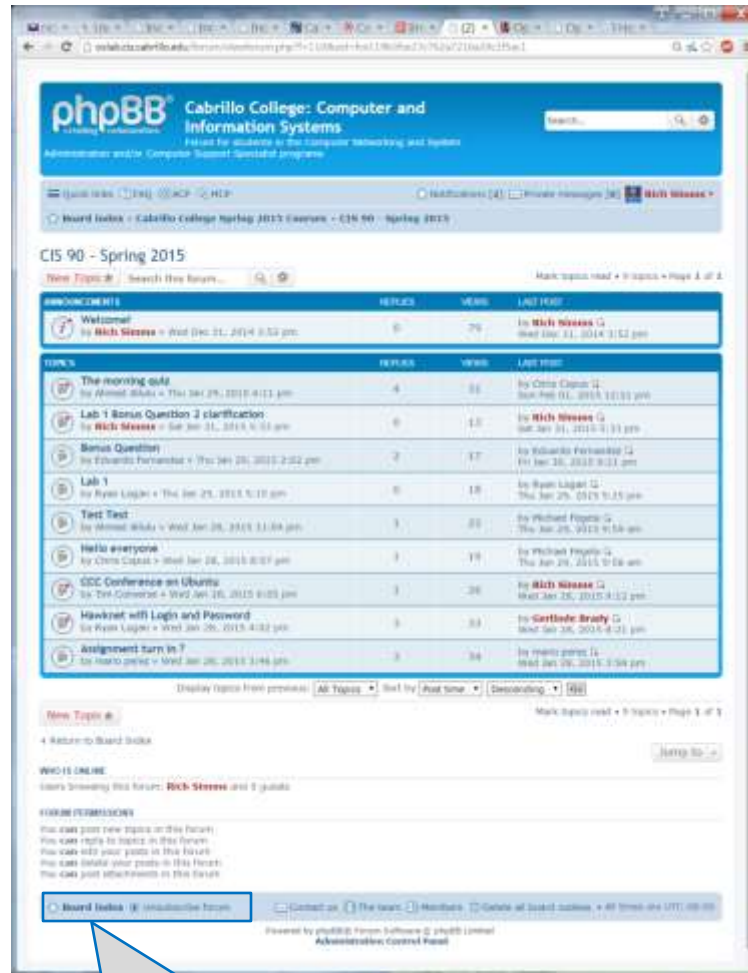
Below the instructions, the forum listing shows several topics:

Topic	Replies	Views	Author	Date
Lab 2 submittals	3	21	by Rich Simms	Mon Sep 13, 2010 6:29 am
Some interesting Linux bc command/examples	0	11	by mike_delfin	Sun Sep 12, 2010 8:59 pm
answer to quz question	1	18	by Jacob Salinas	Sun Sep 12, 2010 3:15 pm
Watch Star Wars using Telnet	?	?	by Jacob Salinas	

To get notifications of new forum posts

2) Go to the CIS 90 forum

1) Login to the forum



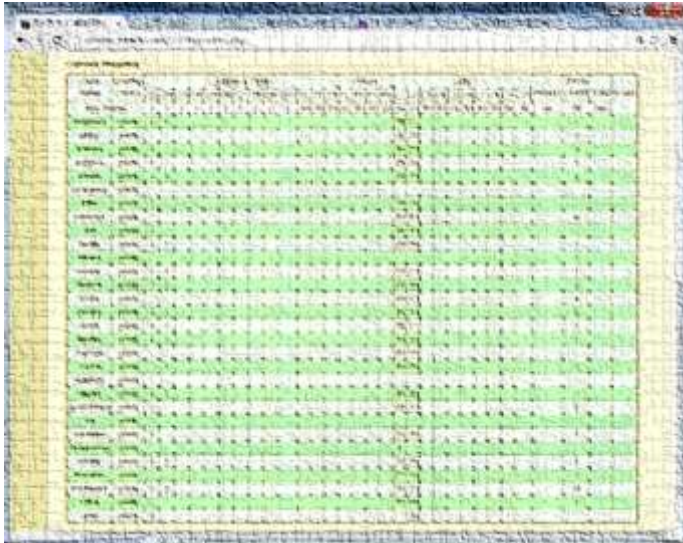
3) Click the "Subscribe" link at the bottom so that it changes to "Unsubscribe".



Where to find your grades

Send me your survey to get your LOR code name.

The CIS 90 website



<http://simms-teach.com/cis90grades.php>

Percentage	Total Points	Letter Grade	Pass/No Pass
90% or higher	504 or higher	A	Pass
80% to 89.9%	448 to 503	B	Pass
70% to 79.9%	392 to 447	C	Pass
60% to 69.9%	336 to 391	D	No pass
0% to 59.9%	0 to 335	F	No pass

At the end of the term I'll add up all your points and assign you a grade using this table

On Opus

checkgrades *codename*
(where *codename* is your LOR codename)



Written by Jesse Warren a past CIS 90 Alumnus

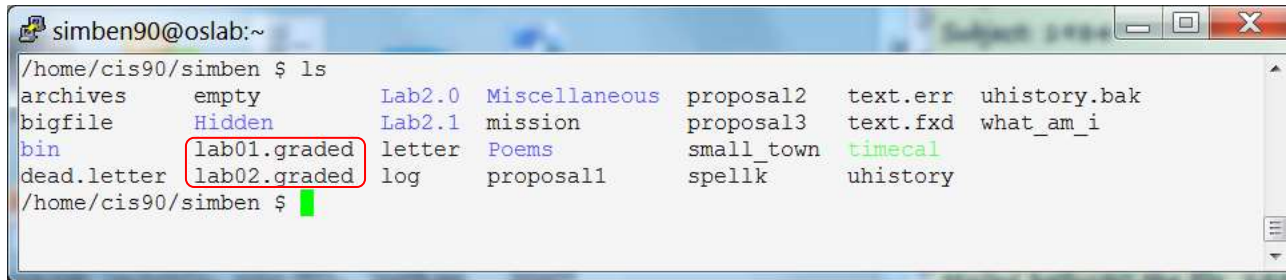
grades *codename*
(where *codename* is your LOR codename)



Written by Sam Tindell a past CIS 90 Alumnus.
Try his tips, schedule and forums scripts as well!

Graded work is copied to your home directories

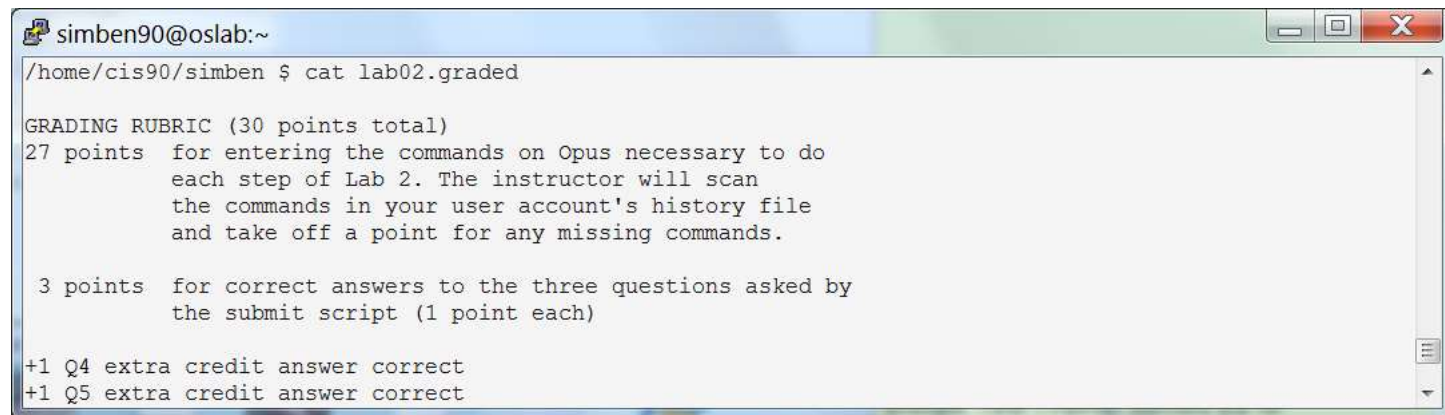
ls



```
simben90@oslab:~
/home/cis90/simben $ ls
archives      empty          Lab2.0  Miscellaneous  proposal2  text.err  uhistory.bak
bigfile       Hidden         Lab2.1  mission        proposal3  text.fxd  what_am_i
bin           lab01.graded  letter  Poems          small_town timecal
dead.letter   lab02.graded  log     proposal1      spellk     uhistory
/home/cis90/simben $
```

*Log in to Opus and use the **ls**, **cat**, or **more** commands to see your graded work*

cat lab02.graded



```
simben90@oslab:~
/home/cis90/simben $ cat lab02.graded

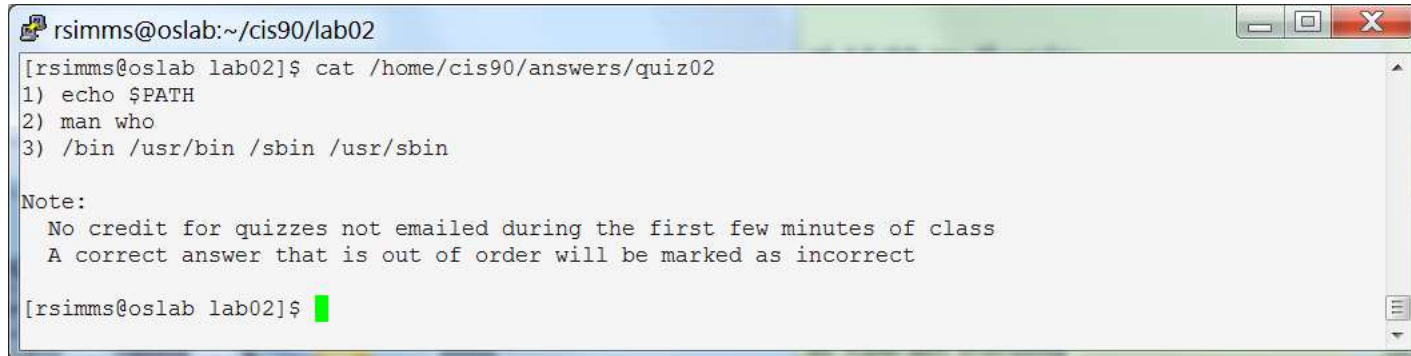
GRADING RUBRIC (30 points total)
27 points for entering the commands on Opus necessary to do
each step of Lab 2. The instructor will scan
the commands in your user account's history file
and take off a point for any missing commands.

3 points for correct answers to the three questions asked by
the submit script (1 point each)

+1 Q4 extra credit answer correct
+1 Q5 extra credit answer correct
```

The answers/ directory on Opus

```
cat /home/cis90/answers/quiz02
```

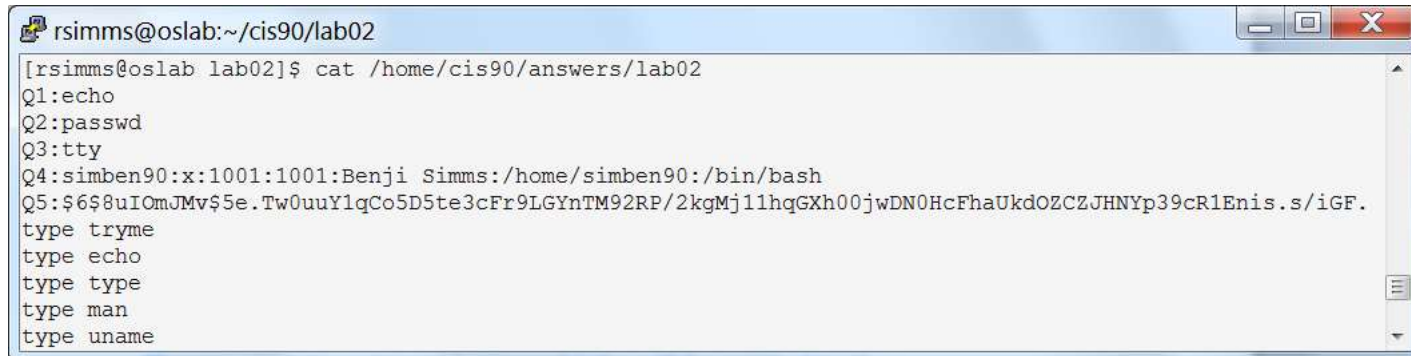


```
rsimms@oslab:~/cis90/lab02
[rsimms@oslab lab02]$ cat /home/cis90/answers/quiz02
1) echo $PATH
2) man who
3) /bin /usr/bin /sbin /usr/sbin

Note:
  No credit for quizzes not emailed during the first few minutes of class
  A correct answer that is out of order will be marked as incorrect

[rsimms@oslab lab02]$
```

```
cat /home/cis90/answers/lab02
```



```
rsimms@oslab:~/cis90/lab02
[rsimms@oslab lab02]$ cat /home/cis90/answers/lab02
Q1:echo
Q2:passwd
Q3:tty
Q4:simben90:x:1001:1001:Benji Simms:/home/simben90:/bin/bash
Q5:$6$8uI0mJMv$5e.Tw0uuY1qCo5D5te3cFr9LGYnTM92RP/2kgMj11hqGXh00jwDN0HcFhaUkdOZCZJHNYp39cR1Enis.s/iGF.
type tryme
type echo
type type
type man
type uname
```

The answers to quizzes, tests and labs will be posted to the /home/cis90/answers/ directory after the due date has passed.

Extra Credit

SS
SS
se. **Another 90 points is available** from **extra credit** assignments. Students can track their overall progress on the chart below. Contact the instructor by email with any questions.

		Forum				Labs										Final Project	Total	
Q10	T1	T2	T3	F1	F2	F3	F4	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	60
3	30	30	30	20	20	20	20	30	30	30	30	30	30	30	30	30	30	30
								30										
								30										

Note the caps on extra credit.

Rich's Cabrillo College CIS Classes
CIS 90 Extra Credit

Home Resources Forums CIS Lab CTC

Login
Hashcards
Admin

[CIS 90](#)
[Previous Classes](#)

95 days till term ends!

[Cabrillo College](#)
[Web Advisor](#)
[CCC Confer](#)
[Static IPs](#)
[Quick Ref](#)
[VM Repairs](#)
[GAH!](#)

CIS 90 Extra Credit
[Course Home](#) [Grades](#)

General Options
Any combination of the following can be done to earn extra credit up to the maximum amount shown on the Grades page:

- **Web site content review** - The first person to email the instructor pointing out an error or typo on this website will get one point of extra credit per content error found. This includes any errors found on the instructor's downloaded materials that have been covered in class. It does not include lesson PowerPoints or Labs that have not yet been covered in class but are pre-published on the website. **(Up to 20 points total)**
- **Develop new Howtos** - Investigate and develop a Howto on a new topic area you are interested in. At the Instructor's discretion and your permission, these Howtos will be published on this web site on the Resources page. Make a proposal first to the instructor on the topic area and to determine the amount of extra credit. Submittals must follow the format of the instructor's Howtos on the Resources web page and be web publishable. **(Up to 20 points per Howto)**
- **Optional activities in lab assignments** - Some of the lab assignments will have optional activities that can be worked for extra credit.
- **Lab assignments** - Some courses may have one or more extra credit labs. Check the Calendar web page. (Point amount varies)



The UNIX File Tree

UNIX File Tree

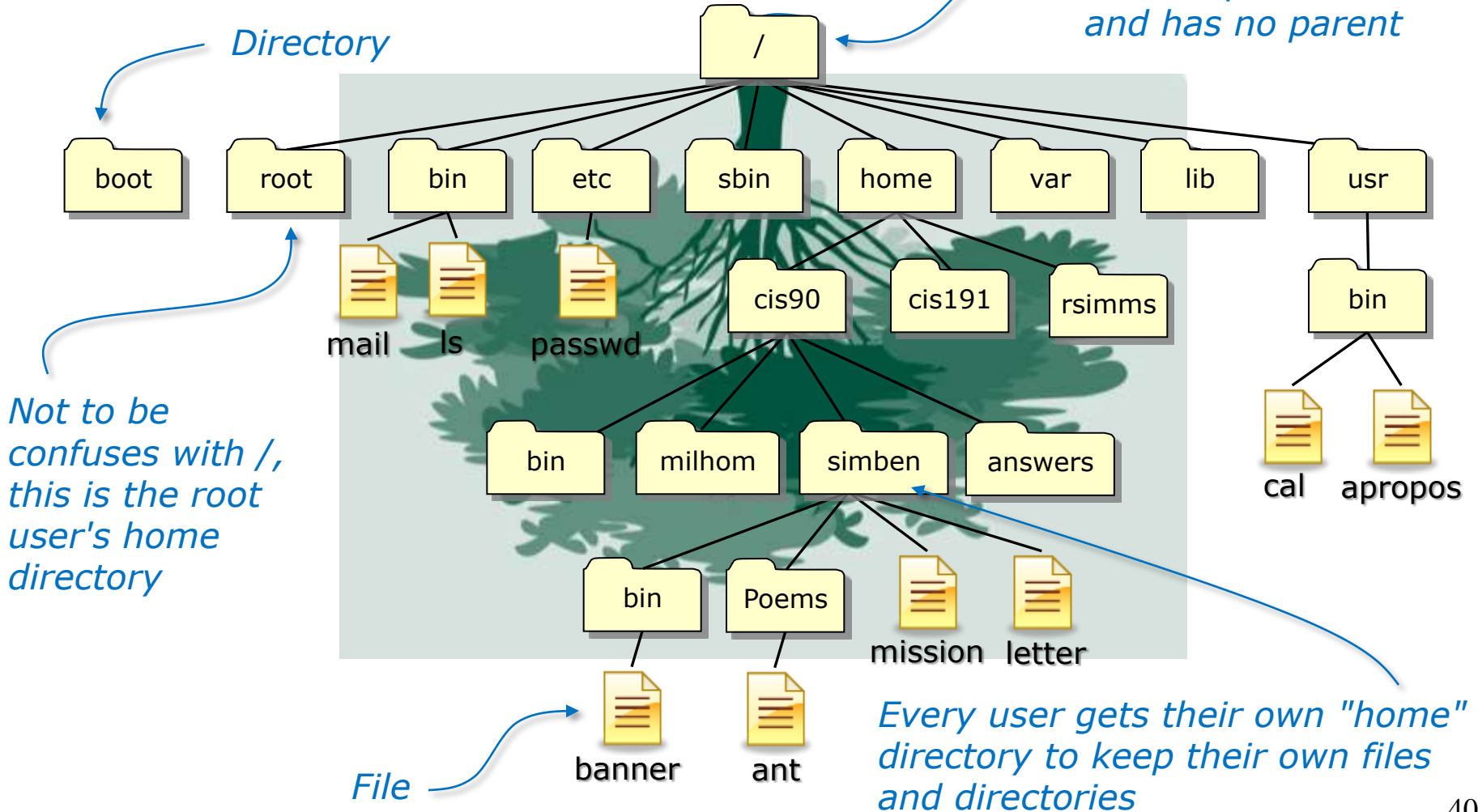
/ = root of the tree



UNIX File Tree

/ = root of the tree

The / "slash" directory is the top of the tree and has no parent



The UNIX/Linux File System Hierarchy

Top-Level Directory	Contents
/bin	binary files forming the commands and shells used by the system administrator and users
/boot	files used during the initial bootup process including the kernel
/dev	device files, like terminals and drives for connected hardware
/etc	system configuration files
/home	individual directories owned by each user
/lib	shared libraries needed to boot the system and run the commands in the root filesystem (i.e. commands in /bin and /sbin)
/lost+found	recovered files that were corrupted by power failures or system crashes
/mnt	mount points for floppies, cds, or other file systems
/opt	add-on software packages and/or commercial applications
/proc	kernel level process information
/root	home directory for the root user
/sbin	system administration commands reserved for the superuser (root)
/tmp	temporary files that are deleted when the system is rebooted or started
/usr	program files and related files for use by all users
/var	log files, print spool files, and mail queues

The CIS 90 student home directories

```

simben90@oslab:~$ ls /
/home/cis90/simben $ ls /
archive boot dev home lost+found misc net proc sbin srv tmp usr
bin cgroup etc lib media mnt opt root selinux sys u var
/home/cis90/simben $

simben90@oslab:~$ ls /home
/home/cis90/simben $ ls /home
backup cis175 cis192 cis98 gerlinde jimg madams rick turnin
cis172 cis191 cis90 dgilmore guest lost+found mmatera rsimms
/home/cis90/simben $

simben90@oslab:~$ ls /home/cis90
/home/cis90/simben $ ls /home/cis90
answers cerjai depot fertho hipmig linmay neljoa primic schrob
beycha cis drydan frisea johjos locjer petaka prites seasky
bin daafad dulste gamant juetay mcgcam popchr remlis simben
brevic davwil fegmic hawwil koujen milhom porjos rodduk spiive
/home/cis90/simben $
  
```

Do you see your home directory in the /home/cis90 directory?



Navigating the UNIX file tree

Navigating the tree



- Use the **cd** command to change directories
(*your legs*)



- Use the **ls** command to list files at your current location
(*your eyes*)



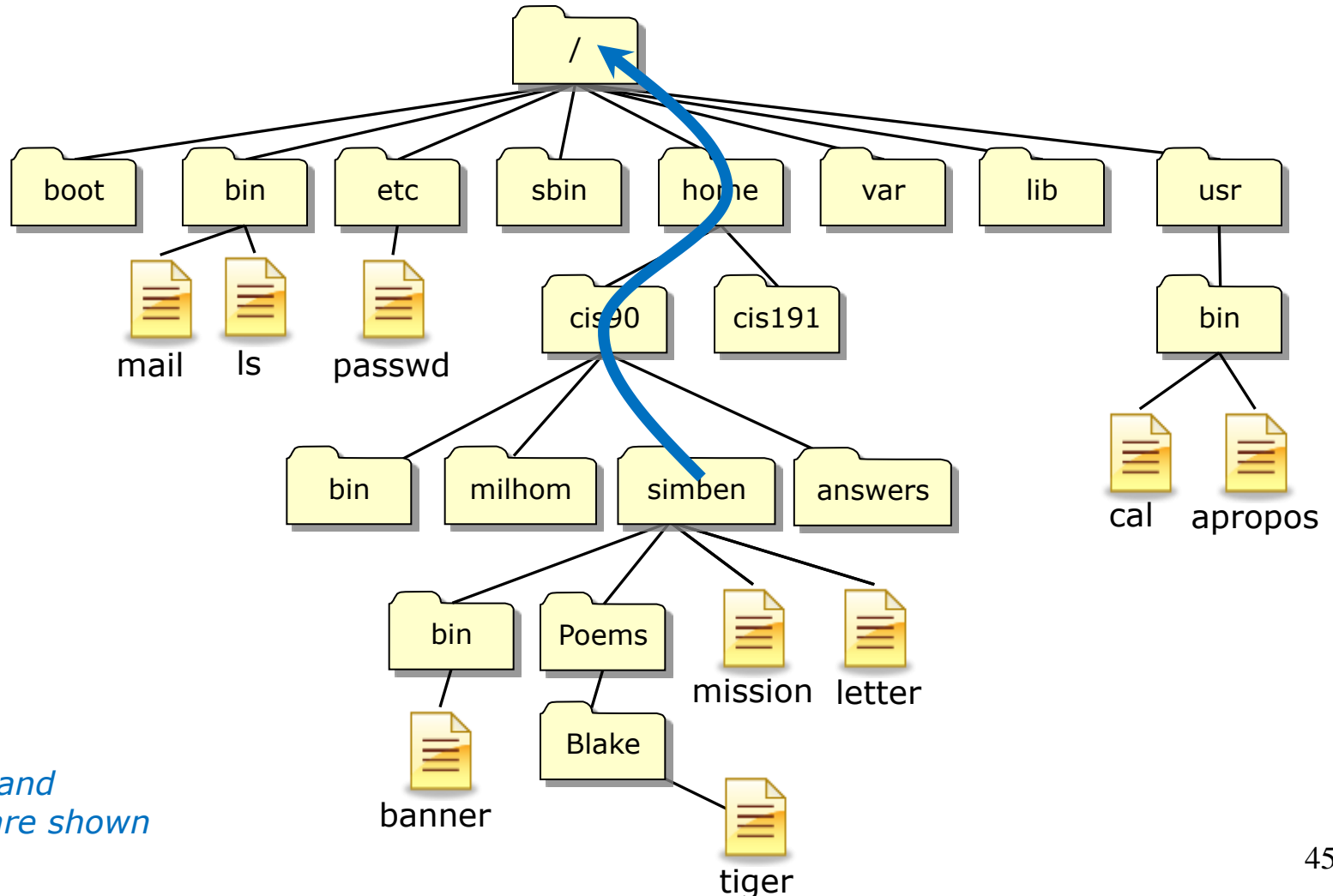
- Use the **pwd** command to show your location
(*your GPS*)

Note, as CIS 90 students your shell prompt uses the PWD variable. As you move around the tree your command prompt will change to show your current location.

*To see why compare the output of the commands: **pwd** and **echo \$PWD***

UNIX File Tree

Navigate from your home directory up to the / directory



Not all files and directories are shown

Navigate from your home directory to the / directory

```

simben90@oslab:/
/home/cis90/simben $ ls
archives      Hidden      lab04-mydata  Miscellaneous  proposal3     text.fxd
bigfile       lab01-collection  Lab2.0        mission        small_town    timecal
bin           lab01.graded   Lab2.1        Poems          spellk        uhistory
dead.letter   lab02-collection  letter        proposal1      submit        what_am_i
empty         lab02.graded    log           proposal2      text.err

/home/cis90/simben $ cd ..
/home/cis90 $ ls
albjon  bin    depot  guest  keichr  maradr  porrya  smimat  tbd08  tbd13  valjos
answers bincam desmat hardyl lamnav milhom quifra  specod  tbd09  tbd14  wrenic
asngab  bownic diljam howmil leeron nieabr  rodduk  tamjim  tbd10  tinsam  zahpau
atirob  boyjef dobtho isoric  lishe  nordak  rodjus  tamtak  tbd11  tranad  zemric
ayalui  cis    espale kadlei locaar  pikann  simben  tbd07  tbd12  urijes

/home/cis90 $ cd ..
/home $ ls
backup  cis175  cis192  cis98  gerlinde  jimg  madams  rick  turnin
cis172  cis191  cis90  dgilmore  guest  lost+found  mmatera  rsimms

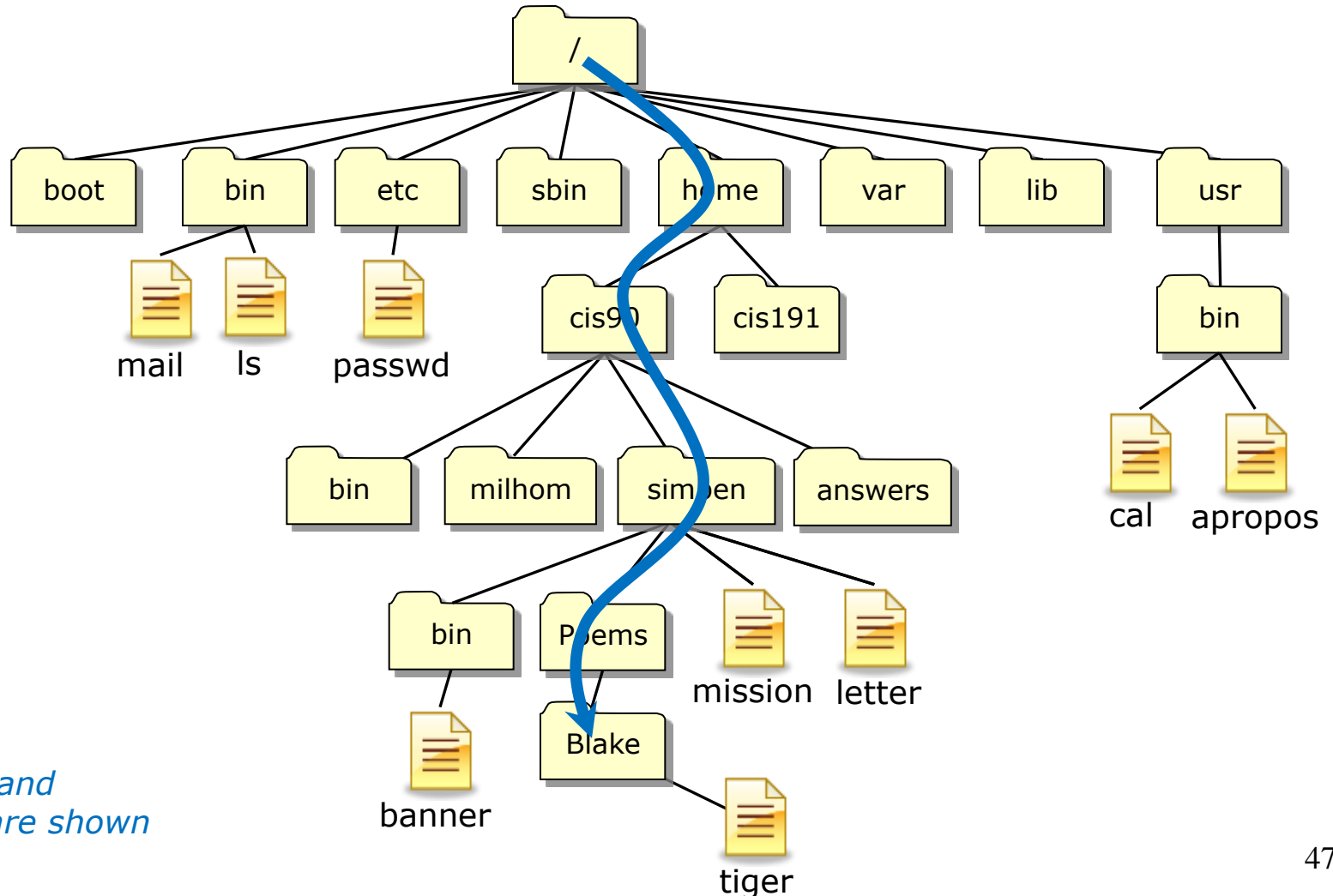
/home $ cd ..
/ $ ls
archive  boot    dev  home  lost+found  misc  net  proc  sbin  srv  tmp  usr
bin      cgroup etc  lib  media      mnt  opt  root  selinux  sys  u  var

/ $
  
```

Use **cd ..** to climb up to the parent directory and **ls** to view the directory contents as you go. Notice how the shell prompt reflects your current location in the tree.

UNIX File Tree

Navigate from the / directory down to your Blake directory



Not all files and directories are shown

Navigate down to the directory of Blake's poems

```

simben90@oslab:~/Poems/Blake
/ $ ls
archive  boot      dev  home  lost+found  misc  net  proc  sbin  srv  tmp  usr
bin      cgroup   etc  lib  media      mnt  opt  root  selinux  sys  u  var
/ $ cd home
/home $ ls
backup  cis175  cis192  cis98  gerlinde  jimg  madams  rick  turnin
cis172  cis191  cis90  dgilmore  guest  lost+found  mmatera  rsimms
/home $ cd cis90
/home/cis90 $ ls
albjon  bin      depot  guest  keichr  maradr  porrya  smimat  tbd08  tbd13  valjos
answers  bincam  desmat  hardyl  lamnav  milhom  quifra  specod  tbd09  tbd14  wrenic
asngab  bownic  diljam  howmil  leeron  nieabr  rodduk  tamjim  tbd10  tinsam  zahpau
atirob  boyjef  dobtho  isoric  lishe  nordak  rodjus  tamtak  tbd11  tranad  zemric
ayalui  cis     espale  kadlei  locaar  pikann  simben  tbd07  tbd12  urijes
/home/cis90 $ cd simben/
/home/cis90/simben $ ls
archives  Hidden  lab04-mydata  Miscellaneous  proposal3  text.fxd
bigfile   lab01-collection  Lab2.0  mission  small_town  timecal
bin       lab01.graded  Lab2.1  Poems  spellk  uhistory
dead.letter  lab02-collection  letter  proposal1  submit  what_am_i
empty      lab02.graded  log  proposal2  text.err
/home/cis90/simben $ cd Poems/
/home/cis90/simben/Poems $ ls
Angelou  ant  Blake  Dickenson  Neruda  nursery  Shakespeare  twister  Yeats
/home/cis90/simben/Poems $ cd Blake/
/home/cis90/simben/Poems/Blake $ ls
jerusalem  tiger
/home/cis90/simben/Poems/Blake $

```

Use `cd <directory>` to climb down directory by directory. Notice how the prompt changes to show your location in the Unix file tree

Navigate back to your home directory



#Part 4

```

simben90@oslab:~/home/cis90/simben/Poems/Blake $ ls
jerusalem tiger
simben90@oslab:~/home/cis90/simben/Poems/Blake $ cd
simben90@oslab:~/home/cis90/simben $ ls
archives      Hidden      lab04-mydata  Miscellaneous  proposal3     text.fxd
bigfile       lab01-collection  Lab2.0        mission        small_town    timecal
bin           lab01.graded     Lab2.1        Poems          spellk        uhistory
dead.letter   lab02-collection  letter        proposal1      submit        what_am_i
empty         lab02.graded     log           proposal2      text.err
simben90@oslab:~/home/cis90/simben $

```

*You always have the power to go home. Just use the **cd** with no arguments to change back to your home directory*

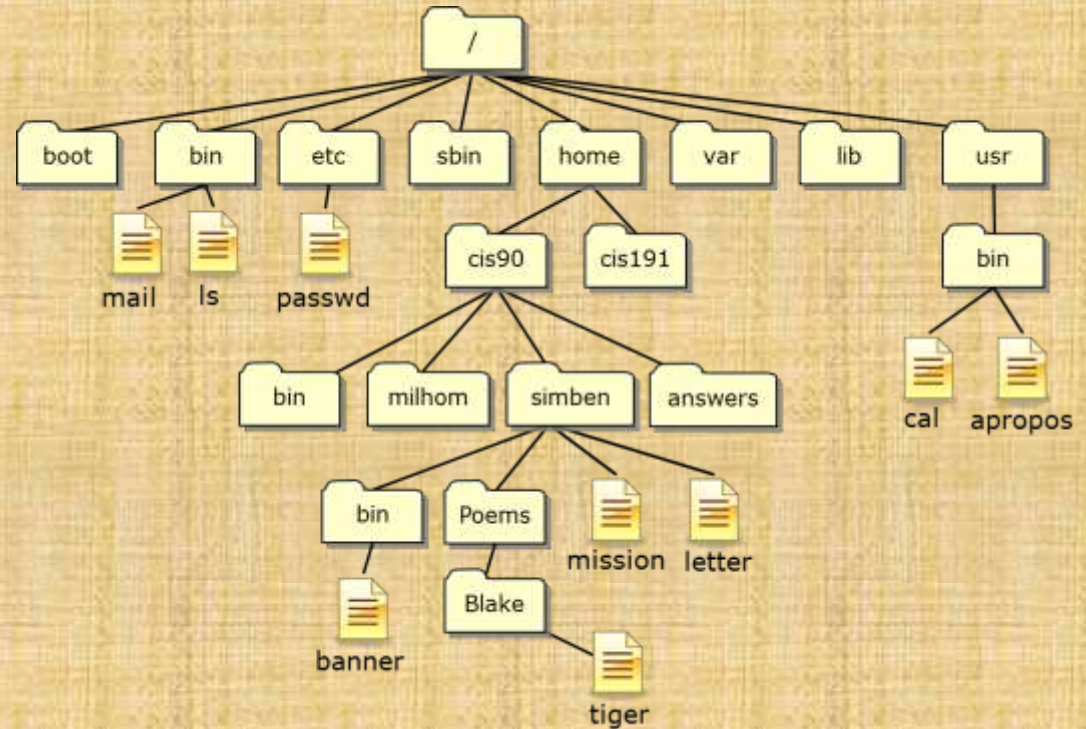


<http://vivandlarry.com/wp-content/uploads/2011/05/oz.jpg>

Dorothy: Oh, will you help me? Can you help me?
 Glinda: You don't need to be helped any longer. You've always had the power to go back to Kansas.
 Dorothy: I have?
 Scarecrow: Then why didn't you tell her before?
 Glinda: Because she wouldn't have believed me. She had to learn it for herself.

Class Field Trip

- 1) /boot
 - The kernel
- 2) /etc
 - motd
 - passwd
- 3) /var
 - mail/
 - www/html
- 4) /home/bin
 - depot
 - bin
 - answers
- 5) /home/simben/Poems
 - various poem directories





UNIX Files

File Systems

Linux

A typical hard drive

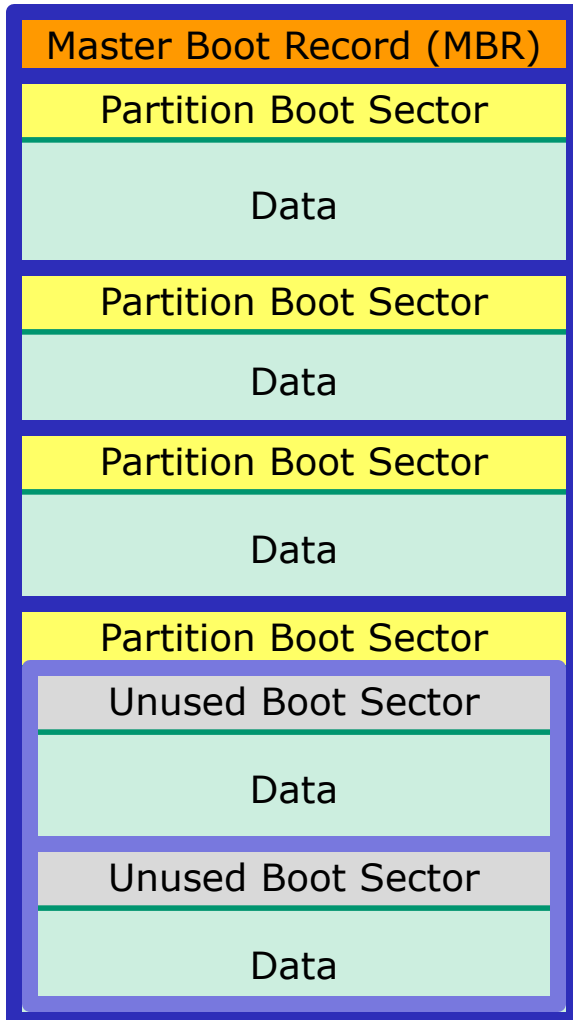


This is where your files actually reside

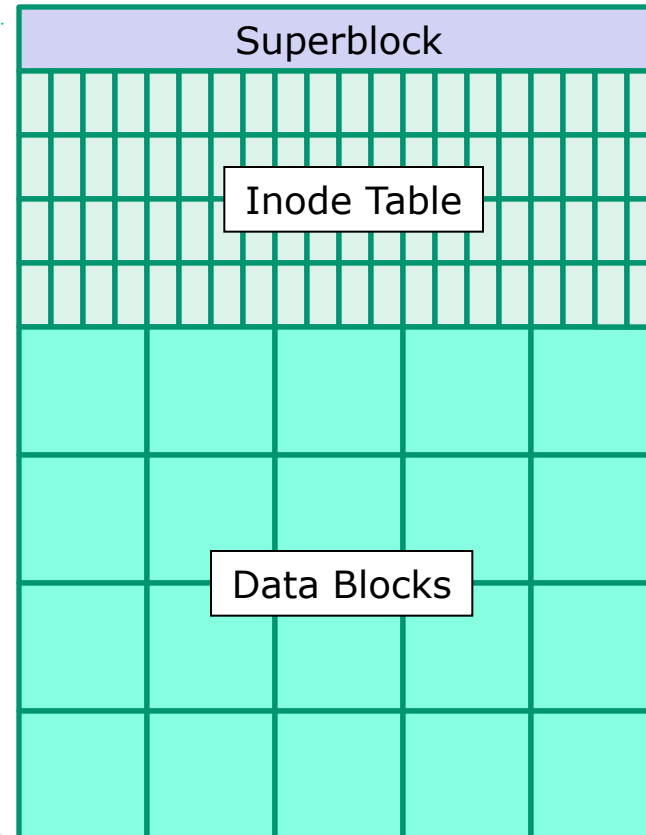


Linux File Systems

The hard drive is partitioned and the data areas can be formatted as a file system. Linux typically uses ext2, ext3 and ext4 file systems. Windows uses FAT32 and NTFS file systems.



extx file system



The three elements of a UNIX file

```
/home/cis90/simben/Poems $ ls
ant Blake nursery Shakespeare twister Yeats
```

```
/home/cis90/simben/Poems $ ls -li twister
102625 -rw-r--r-- 1 simben90 cis90 151 Jul 20 2001 twister
```

```
/home/cis90/simben/Poems $ cat twister
A tutor who tooted the flute,
tried to tutor two tooters to toot.
Said the two to the tutor,
"is it harder to toot? Or to
tutor two tooters to toot?"
```

filename

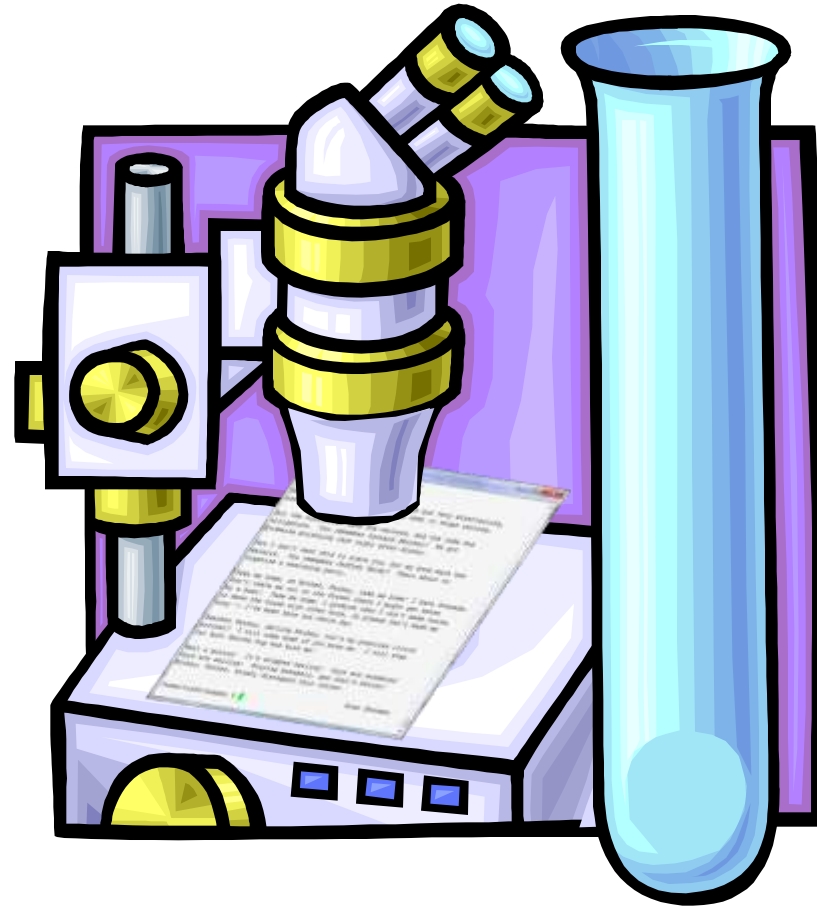
+

inode

+

data

Let's look at the file named letter in Benji's home directory



`ls -il letter` *will show the inode number and a long listing of the letter file*
`cat letter` *will show the data contents of the letter file*

```
/home/cis90/simben $ ls -li letter
```

```
9662 -rw-r--r--. 1 simben90 cis90 1044 Jul 20 2001
```

```
letter
```

filename



inode

```
/home/cis90/simben $ cat letter
```

```
Hello Mother! Hello Father!
```

```
Here I am at Camp Granada. Things are very entertaining,  
and they say we'll have some fun when it stops raining.
```

```
All the counselors hate the waiters, and the lake has  
alligators. You remember Leonard Skinner? He got  
ptomaine poisoning last night after dinner.
```

```
Now I don't want this to scare you, but my bunk mate has  
malaria. You remember Jeffrey Hardy? Their about to  
organize a searching party.
```

```
Take me home, oh Mother, Father, take me home! I hate Granada.  
Don't leave me out in the forest where I might get eaten  
by a bear! Take me home, I promise that I won't make noise,  
or mess the house with other boys, oh please don't make me  
stay -- I've been here one whole day.
```

```
Dearest Father, darling Mother, how's my precious little  
brother? I will come home if you miss me. I will even  
let Aunt Bertha hug and kiss me!
```

```
Wait a minute! It's stopped hailing! Guys are swimming!  
Guys are sailing! Playing baseball, gee that's better!  
Mother, Father, kindly disregard this letter.
```

```
Alan Sherman
```

data



bigfile 19470
bin 9628
letter 9662

*filenames are stored
in directories, **not** in
inodes*

/home/cis90/simben

Hello Mother! Hello Father!

Here I am at Camp Granada. Things are very entertaining, and they say we'll have some fun when it stops raining.

All the counselors hate the waiters, and the lake has alligators. You remember Leonard Skinner? He got ptomaine poisoning last night after dinner.

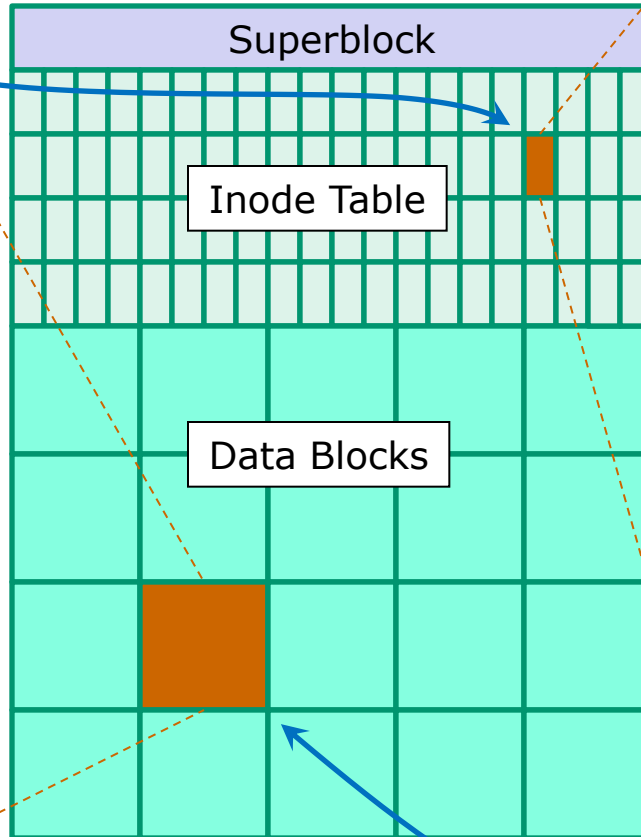
Now I don't want this to scare you, but my bunk mate has malaria. You remember Jeffrey Hardy? Their about to organize a searching party.

Take me home, oh Mother, Father, take me home! I hate Granada.
Don't leave me out in the forest where I might get eaten by a bear! Take me home, I promise that I won't make noise, or mess the house with other boys, oh please don't make me stay -- I've been here one whole day.

Dearest Father, darling Mother, how's my precious little brother? I will come home if you miss me. I will even let Aunt Bertha hug and kiss me!

Wait a minute! It's stopped hailing! Guys are swimming!
Guys are sailing! Playing baseball, gee that's better!
Mother, Father, kindly disregard this letter.

Alan Sherman



ext2 file system

9662	inode number
-	Type
rw-r--r--	Permissions
1	Number of links
simben90	User
cis90	Group
1044	Size
2001-07-20	Modification time
2012-09-17	Access Time
2012-08-01	Change time
Pointer(s) to data blocks	Pointer(s) to data blocks

```
/home/cis90/simmsben $ ls -il letter
9662 -rw-r--r--. 1 simben90 cis90 1044 Jul 20 2001 letter
```

Directories are files too!

- Directories are implemented as files
- The data in a directory includes pairs of filenames and inode numbers (kind of like a phone book)
- Every directory can contain further sub-directories

In other operating systems like Mac and Windows, a directory is often referred to as a "folder" and represented as a office folder icon on the desktop.

Activity

Type these commands in your home directory:

ls -i

ls -il letter

cat letter

Type the inode of your letter file in the chat window



Unix Filename Conventions

UNIX file name conventions

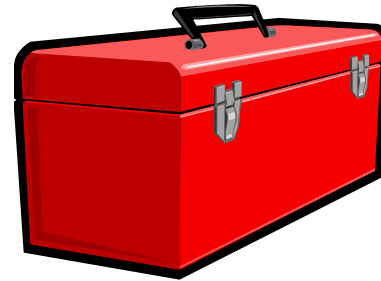
Unix filenames are case sensitive

File names can be any combination of the following:

- Upper and lower case letters: **A-Z** and **a-z**
- Numbers: **0-9**
- Periods, underscores, hyphens: **. _ -**
- Examples: letter, Lab2.1, my_files, my-files

Avoid using the following characters in filenames

- **| ; , ! @ # \$ () < > / \ " ' ` ~ { } [] = + & ^**
<space> <tab>



More commands for your toolbox

Viewing Text Files



Lesson 4 commands for your toolbox

cat

- view a text file

NEW **more**

- view a large text file by scrolling down

NEW **less**

- view a large text file by scrolling down and up

NEW **head**

- view the beginning lines of a text file

NEW **tail**

- view the last lines of a text file

NEW **wc**

- count the lines, words and characters in a text file

NEW **xxd**

- view a binary data file as a hex dump

NEW **cd**

- change to a different directory

ls

- list files

pwd

- show name of current/working directory

file

- show additional file information

type

- show location of a command on path

Viewing **text** files:

- file *useful for identifying if a file is text or binary*
- cat *to print a file*
- more *to scroll down through a file*
- less *to scroll down and up a file*
- head *to print the beginning lines of a file*
- tail *to print the last lines of a file*
- wc *count the words and lines in a text file*

ASCII Text Files

Computers store everything as binary 0's and 1's.

ASCII = American Standard Code for Information Interchange.

ASCII defines binary patterns of 0's and 1's to represent printable text characters.

For example, the letter O is represented by 01001111, the letter z is represented by 01111010.

If a file has data that only contains ASCII text patterns then it is considered a **text file** and "printable".

If some or all of the bit patterns are not ASCII characters then the file is considered a **binary file** and unprintable.

To see all the ASCII characters use the **man ascii** command.

Thanks Hunter! See Hunter's post at <http://oslab.cishawks.net/forum/viewtopic.php?f=88&t=2258&p=8357>

Identifying text files with the file command



#Bangalore

```

/home/cis90/simben $ file letter Poems proposal1 mission uhistory what_am_i
letter:      ASCII English text
Poems:      directory
proposal1:  ASCII English text
mission:    ASCII English text
uhistory:   ASCII mail text
what_am_i:  data
/home/cis90/simben $
    
```

Look for the word "text" in the output to indicate an ASCII text file

If you don't see "text" it's a binary file and unprintable. Note: what_am_i and Poems are not text files

The text viewing commands like cat, more, head, etc. only work on text files. They are not meant to be used to view binary data files or directories.

cat command

used to view a text file

```
/home/cis90/simben $ cat letter  
Hello Mother! Hello Father!
```

A single argument, letter, is given to the cat command to process

```
Here I am at Camp Granada. Things are very entertaining,  
and they say we'll have some fun when it stops raining.
```

```
All the counselors hate the waiters, and the lake has  
alligators. You remember Leonard Skinner? He got  
ptomaine poisoning last night after dinner.
```

< Snipped >

```
Wait a minute! It's stopped hailing! Guys are swimming!  
Guys are sailing! Playing baseball, gee that's better!  
Mother, Father, kindly disregard this letter.
```

Alan Sherman

```
/home/cis90/simben $
```

cat command

viewing multiple text files

```
/home/cis90/simben $ cat spellk letter
```

```
Spell Check
```

```
Eye halve a spelling chequer  
It came with my pea sea  
It plainly marques four my revue  
< snipped >  
Eye have run this poem threw it  
I am shore your pleased two no  
Its letter perfect awl the weigh  
My chequer tolled me sew.
```

```
Hello Mother! Hello Father!
```

```
Here I am at Camp Granada. Things are very entertaining,  
and they say we'll have some fun when it stops raining.
```

```
< snipped >
```

```
Wait a minute! It's stopped hailing! Guys are swimming!  
Guys are sailing! Playing baseball, gee that's better!  
Mother, Father, kindly disregard this letter.
```

```
Alan Sherman
```

```
/home/cis90/simben $
```

Multiple arguments, spellk and letter, are passed to the cat command to process

spellk

letter

cat command viewing long text files

- Problem: if you **cat** really long files the text at the beginning is scrolled off and cannot be read.
- For example: `cat /usr/share/doc/bash-3.2/NEWS`

```

roddyduk@opuz:~$ cat /usr/share/doc/bash-3.2/NEWS
1. The shell is somewhat more efficient: it uses a little less memory and
   makes fewer system calls.
4. Changes of interest in the Readline implementation
   a. There is now support for readline 'callback' functions.
   b. There is now support for user-supplied input, redisplay, and
      preparation functions.
   c. Most of the shell-specific code in readline has been generalized or
      removed.
   d. Most of the annoying redisplay bugs have been fixed, notably the
      problems with incremental search and excessive redrawing when special
      characters appear in the prompt string.
   e. There are new library functions and variables available to application
      writers, most having to do with completion and quoting.
   f. The NEWLINE character (^J) is now treated as a search terminator by the
      incremental search functions.
/home/cis90/roddyduk $
  
```

*And virtual terminals
have no scroll bars!*

```

interactive.
1. The shell is somewhat more efficient: it uses a little less memory and
   makes fewer system calls.
4. Changes of interest in the Readline implementation
   a. There is now support for readline 'callback' functions.
   b. There is now support for user-supplied input, redisplay, and terminal
      preparation functions.
   c. Most of the shell-specific code in readline has been generalized or
      removed.
   d. Most of the annoying redisplay bugs have been fixed, notably the problems
      with incremental search and excessive redrawing when special characters
      appear in the prompt string.
   e. There are new library functions and variables available to application
      writers, most having to do with completion and quoting.
   f. The NEWLINE character (^J) is now treated as a search terminator by the
      incremental search functions.
[cisco@localhost cisco]$_
  
```

*Terminal windows (like PuTTY)
have scroll bars but the
number of lines they buffer
can be exceeded.*

more command viewing long text files

- Use the **more** command for scrolling through really long text files
- For example: **more /usr/share/doc/bash-3.2/NEWS**

```

roddyduk@opus:~
This is a terse description of the new features added to bash-3.2 since
the release of bash-3.1. As always, the manual page (doc/bash.1) is
the place to look for complete descriptions.

1. New Features in Bash
a. Changed the parameter pattern replacement pattern at the beginning of the string
   combination doesn't make any sense.
b. When running in 'word expansion' mode, process substitution.
c. Loadable builtins now work on Mac OS X.
d. Shells running in posix mode no longer check for binary files.
e. The code that checks for binary files now checks only for NUL rather than
   and.
f. Quoting the string argument to the string matching, as with the other
--More-- (1%)
  
```

```

[cisco@localhost cisco]$ more /usr/share/doc/bash-2.05b/NEWS
This is a terse description of the new features added to bash-2.05b since
the release of bash-2.05a. As always, the manual page (doc/bash.1) is
the place to look for complete descriptions.

1. New Features in Bash
a. If set, TIMEOUT is the default timeout for the 'read' builtin.
b. 'type' has two new options: '-f' suppresses shell function lookup, and
   '-P' forces a $PATH search.
c. New code to handle multibyte characters.
d. 'select' was changed to be more ksh-compatible, in that the menu is
   reprinted each time through the loop only if REPLY is set to NULL.
   The previous behavior is available as a compile-time option.
e. 'complete -d' and 'complete -o dirnames' now force a slash to be
   appended to names which are symlinks to directories.
f. There is now a bindable edit-and-execute-command readline command,
   like the vi-mode 'v' command, bound to C-xC-e in emacs mode.
--More-- (2%)
  
```

Use the **space bar** to page forward and **q** to quit

more command viewing multiple text files

- The **more** command can take multiple arguments

```
/home/cis90/simben $ more spellk letter
```

```
::::::::::::::::::  
spellk  
::::::::::::::::::  
Spell Check
```

```
Eye halve a spelling chequer  
It came with my pea sea  
< snipped >  
Its letter perfect awl the weigh  
My chequer tolled me sew.
```

```
::::::::::::::::::  
letter  
::::::::::::::::::
```

```
Hello Mother! Hello Father!  
< snipped >  
Guys are sailing! Playing baseball, gee that's better!  
Mother, Father, kindly disregard this letter.
```

Notice with multiple files as arguments, each file has a header to separate it from the other files

Alan Sherman

```
/home/cis90/simben $
```

less command

viewing long text files



- Use the **less** command to scroll forward and backward through really long text files. (just like the man command works)
- For example: **less /usr/share/doc/bash-3.2/NEWS**

"less is more" 😊

```

k. The `gnu_errfmt` option is enabled automatically if the shell is running
   in an emacs terminal window.

l. New configuration option: --single-help-strings. Causes long help text
   to be written as a single string; intended to ease.

m. The COMP_WORDBREAKS variable now causes the list of completions to
   be emptied when the variable is unset.

n. An unquoted expansion of $* when $IFS is empty now causes the
   parameters to be concatenated if the expansion does not contain
   splitting.

o. Bash now inherits $_ from the environment if it is not set.

p. New shell option: nocasematch. If non-zero, shell option names are
   case when used by `case` and `[[]` commands.

q. The `printf` builtin takes a new option: -v var. The format string
   to be placed into var instead of on stdout.

r. By default, the shell no longer reports processes.

:
  
```

```

k. If a numeric argument is supplied to one of the bash globbing completion
   functions, a `*' is appended to the word before expansion is attempted.

l. The bash globbing completion functions now allow completions to be listed
   with double tabs or if `show-all-if-ambiguous` is set.

m. New `-o nospace` option for `complete` and `compgen` builtins; suppresses
   readline's appending a space to the completed word.

n. New `here-string` redirection operator: <<< word.

o. When displaying variables, function attributes and definitions are shown
   separately, allowing them to be re-used as input (attempting to re-use
   the old output would result in syntax errors).

p. There is a new configuration option `--enable-mem-scramble`, controls
   bash malloc behavior of writing garbage characters into memory at
   allocation and free time.

q. The `complete` and `compgen` builtins now have a new `-s/-A service`
   option to complete on names from /etc/services.

r. `read` has a new `-u fd` option to read from a specified file descriptor.
  
```

Use the **pg up/dn** and up/down arrows to move through text file. Use **q** to quit. For multiple arguments use **:n** and **:p** to move between multiple text files. See the man page for many more options like searching.



head command

view the first lines in a text file

- Use the **head** command to show the first several lines of a file.
- Use the **-n <number>** option to control the number of lines printed.

```
/home/cis90/simben $ head proposal1 Print the first lines of the file proposal1
A Plan for the Improvement of English Spelling
  by Mark Twain
For example, in Year 1 that useless letter "c" would be dropped to be replased
either by "k" or "s", and likewise "x" would no longer be part of the alphabet.
The only kase in which "c" would be retained would be the "ch" formation, which
will be dealt with later. Year 2 might reform "w" spelling, so that "which" and
"one" would take the same konsonant, wile Year 3 might well abolish "y"
replasing it with "i" and Iear 4 might fiks the "g/j" anomali wonse and for all.
Jenerally, then, the improvement would kontinue iear bai iear with Iear 5 doing
awai with useless double konsonants, and Iears 6-12 or so modifaing vowlz and
/home/cis90/simben $
```

```
/home/cis90/simben $ head -n 3 proposal1 Print the first 3 lines of the file proposal1
A Plan for the Improvement of English Spelling
  by Mark Twain
For example, in Year 1 that useless letter "c" would be dropped to be replased
/home/cis90/simben $
```

head command

view the first lines of multiple text files

```
/home/cis90/simben $ head -n2 mission letter spellk log
```

Print the first 2 lines of each of these files

```
==> mission <==
```

```
Mission * Purpose * Values
```

```
==> letter <==
```

```
Hello Mother! Hello Father!
```

Note the small banners containing the filename which separates each file.

```
==> spellk <==
```

```
Spell Check
```

The second line of the first three files are blank.

```
==> log <==
```

```
lab01 was submitted on Wed Feb 8 16:23:35 PST 2012
```

```
lab01 was submitted on Wed Feb 8 16:58:20 PST 2012
```


tail command

view the last lines in a text file

- Use the **tail** command to show the last several lines of a file.
- Use the **-n <number>** option to control the number of lines printed.

```
/home/cis90/simben $ tail mission Print the tail end of the file  
environment which aids students in their pursuit of transfer,  
career preparation, personal fulfillment, job advancement, and  
retraining goals.
```

```
Our core values are academic freedom, critical and independent  
thinking, and respect for all people and cultures. Our commitment  
is to encourage excellence, offer a balanced curriculum, promote  
teaching methods for diverse learning styles, and involve and  
enrich our community.
```

```
/home/cis90/simben $ tail -n3 mission Print the last 3 lines of the file  
teaching methods for diverse learning styles, and involve and  
enrich our community.
```



wc command

count words and lines in a text file

```
/home/cis90/simben $ wc letter
28 182 1044 letter
```

#bytes → 1044
#words → 182
#lines → 28

```
/home/cis90/simben $ wc -l letter
```

28 letter

Use the -l option to count just the number of lines

```
/home/cis90/simben $ wc -w letter
```

182 letter

Use the -w option to count just the number of words

```
/home/cis90/simben $ wc letter mission proposal1
28 182 1044 letter
18 107 759 mission
16 196 1074 proposal1
62 485 2877 total
```

The wc command can take multiple arguments

Class Exercise Viewing Text Files

- Print the first 3 lines of the log file

head -n3 log

- Count the number of words in small_town

wc -w small_town

- Print the proposal1 file

cat proposal1



What happens if you use tac instead of cat? (tac is cat spelled backwards)

Viewing binary files

Viewing **binary** files:

- `file` *useful for identifying whether a file is text or binary*
- `xxd` *show the contents of a binary file as a "hex dump"*

Identifying Binary Files

binary files

```
/home/cis90/simben $ file /bin/uname what_am_i spellk bin/enlightenment
/bin/uname:          ELF 32-bit LSB executable, Intel 80386, version 1
(SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.18,
stripped
what_am_i:          data
spellk:             ASCII English text
bin/enlightenment: POSIX shell script text executable
```

text files

If the output of the file command does not contain "text" then the file is most likely a binary file



Binary Files

Binary files should not be viewed with cat, more, less, head, tail, etc.

```

/home/cis90/simben $ cat /bin/uname
ELF04`I4(444444>>@ ( A HHH Ptd644Qtd/lib/ld-
linux.so.2GNU (B` (*K G->Kyycg}Ti w)
C52L/9=@xH ^fOI
G<' 6?w C*Y A $) ,K, f" ) ,K H. . . .
./d8/</ / /sii /ii w~w
~w~w~wii
) *+, $(,08 <
< snipped >
uTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTT
YPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPu
TTYPuTTYPuTTYPuTTY
/home/cis90/simben $

```

*Tip: Use the **reset** command to fix terminal if it gets really "sick"*



Binary Files

Use xxd command to view

The file /bin/uname is viewed as a hex dump

E=ASCII 45 at 00000001
L=ASCII 4c at 00000002
F=ASCII 46 at 00000003

```

/home/cis90/simben $ xxd /bin/uname
00000000: 7f45 4c46 0101 0100 0000 0000 0000 0000  .ELF.....
00000010: 0200 0300 0100 0000 308b 0408 3400 0000  .....0...4...
00000020: 6049 0000 0000 0000 3400 2000 0800 2800  `I.....4. ...(.
00000030: 1f00 1e00 0600 0000 3400 0000 3480 0408  .....4...4...
00000040: 3480 0408 0001 0000 0001 0000 0500 0000  4.....
00000050: 0400 0000 0300 0000 3401 0000 3481 0408  .....4...4...
00000060: 3481 0408 1300 0000 1300 0000 0400 0000  4.....
00000070: 0100 0000 0100 0000 0000 0000 0080 0408  .....
< snipped >
0004df0: 0000 0000 0000 0000 d842 0000 6c05 0000  .....B..l...
0004e00: 0000 0000 0000 0000 0400 0000 0100 0000  .....
0004e10: 0100 0000 0300 0000 0000 0000 0000 0000  .....
0004e20: 4448 0000 1901 0000 0000 0000 0000 0000  DH.....
0004e30: 0100 0000 0000 0000
/home/cis90/simben $
    
```

Hexadecimal offsets into the file

The printable "ELF" above is located between hex offsets 00000000 and 00000010 shown on the left column

Class Exercise

Where is the hostname command?

```
type hostname
```

What kind of file is the hostname command?

```
file /bin/hostname
```

Try to cat the hostname command:

```
cat /bin/hostname
```

Do a hex dump of the hostname command:

```
xxd /bin/hostname
```



Basic file types



Understanding a Long Listing

```
/home/cis90/simben $ ls letter
letter
```

The l option on the ls command produces a "long listing" that shows more information

```
/home/cis90/simben $ ls -l letter
-rw-r--r--. 1 simben90 cis90 1044 Jul 20 2001 letter
```

permissions
owner (the username that owns the file)
group
size (in bytes)
modification date
filename

number of hard links

file type

Understanding a Long Listing



The l option on the ls command produces a "long listing" that shows more information

```

/home/cis90/simben $ ls -l
total 132
-rw-rw-r--. 1 simben90 cis90 4008 Sep 11 22:23 archives
-rw-r--r--. 2 simben90 cis90 10576 Jul 20 2001 bigfile
drwxr-xr-x. 2 simben90 cis90 4096 Sep 11 2005 bin
-rw-----. 1 simben90 cis90 1445 Sep 13 15:13 dead.letter
-rw-r--r--. 1 simben90 cis90 0 Jul 20 2001 empty
d-----. 2 simben90 cis90 4096 Feb 1 2002 Hidden
-r-----. 1 simben90 staff 2780 Sep 6 13:47 lab01.graded
-r-----. 1 simben90 staff 1312 Sep 13 12:27 lab02.graded
drwxr-xr-x. 2 simben90 cis90 4096 Feb 17 2001 Lab2.0
drwxr-xr-x. 3 simben90 cis90 4096 Feb 17 2001 Lab2.1
-rw-r--r--. 1 simben90 cis90 1044 Jul 20 2001 letter
< snipped >
-rw-r--r--. 1 simben90 cis90 485 Aug 26 2003 spellk
-rw-r--r--. 1 simben90 cis90 250 Jul 20 2001 text.err
-rw-r--r--. 1 simben90 cis90 231 Jul 20 2001 text.fxd
-rwxr-xr-x. 1 simben90 cis90 509 Jun 6 2002 timecal
-rw-rw-r--. 1 simben90 cis90 20829 Sep 17 18:06 uhistory
-rw-r--r--. 1 simben90 cis90 352 Jul 20 2001 what_am_i
    
```

A "d" indicates a directory

A "-" indicates a regular file

Column 1 of long listings shows basic file types

Directory filenames also appear in blue

Some Common File Types



Column 1 of long listing	Type	How to make one
d	Directory	mkdir
-	Regular <ul style="list-style-type: none"> • Programs • Text • Data (binary) • Many more ... <i>Use the file command to further classify regular files</i>	touch vi >
l	Symbolic link	ln -s
c	Character special device	mknod
b	Block special device	mknod

Every file has a specific type attribute which is stored in the inode.

*File types can be viewed in column 1 of **long listings**.*

The /etc directory (Ubuntu)

The terminal window shows the following output for the /etc directory:

```

-rw-r--r-- 1 root root 342 2008-06-20 11:10 popularity-contest.conf
drwxr-xr-x 4 root root 4096 2008-04-22 13:52 power
drwxr-xr-x 8 root dip 4096 2008-04-22 14:01 ppp
-rw-r--r-- 1 root root 497 2008-04-22 13:49 profile
drwxr-xr-x 2 root root 4096 2008-04-15 01:53 profile.d
-rw-r--r-- 1 root root 2510 2007-12-03 17:04 protocols
drwxr-xr-x 2 root root 4096 2008-04-22 14:03 pulse
drwxr-xr-x 2 root root 4096 2008-04-22 14:03 purple
drwxr-xr-x 2 root root 4096 2008-04-22 13:49 python
drwxr-xr-x 2 root root 4096 2008-04-22 13:49 python2.5
drwxr-xr-x 2 root root 4096 2008-06-20 11:12 rc0.d
drwxr-xr-x 2 root root 4096 2008-04-22 14:07 rc1.d
drwxr-xr-x 2 root root 4096 2008-06-20 11:12 rc2.d
drwxr-xr-x 2 root root 4096 2008-06-20 11:12 rc3.d
drwxr-xr-x 2 root root 4096 2008-06-20 11:12 rc4.d
drwxr-xr-x 2 root root 4096 2008-06-20 11:12 rc5.d
drwxr-xr-x 2 root root 4096 2008-06-20 11:12 rc6.d
-rwxr-xr-x 1 root root 306 2008-04-22 13:49 rc.local
drwxr-xr-x 2 root root 4096 2008-04-22 14:05 rcS.d
drwxr-xr-x 2 root root 4096 2008-04-22 14:03 readahead
drwxr-xr-x 3 root root 4096 2008-04-22 13:53 resolvconf
-rw-r--r-- 1 root root 170 2008-06-24 10:44 resolv.conf
-rwxr-xr-x 1 root root 268 2008-04-04 07:07 rmt
-rw-r--r-- 1 root root 887 2007-12-03 17:04 rpc
drwxr-xr-x 2 root root 4096 2008-06-20 11:15 samba
drwxr-xr-x 3 root root 4096 2008-04-22 13:59 sane.d
drwxr-xr-x 2 root root 4096 2008-04-22 14:05 scim
-rw-r--r-- 1 root root 3663 2007-10-23 12:02 screenrc
  
```

Annotations in the image explain the permissions:

- "-" regular files (black):** points to files like `profile`, `protocols`, `rc.local`, `rcS.d`, `resolv.conf`, `rmt`, `rpc`, `sane.d`, and `screenrc`.
- "d" directories (blue):** points to directories like `power`, `ppp`, `profile.d`, `pulse`, `purple`, `python`, `python2.5`, `rc0.d` through `rc6.d`, and `rcS.d`.
- "-" regular files with x (execute) bit set (green) in cols 4,7, 10:** points to files like `rc.local`, `rmt`, and `screenrc`.
- "-" regular file (black):** points to `popularity-contest.conf`.

A portion of the /bin directory (Ubuntu)



rsimms@ulysses: /bin

```
rsimms@ulysses:/bin$ ls -l s* z*
```

Permissions	Size	Owner	Group	Size	Mod Date	Time	File Name
-rwxr-xr-x	1	root	root	40724	2007-12-04	07:50	sed
l -rwxrwxrwx	1	root	root	15	2008-06-20	11:03	setpci -> /usr/bin/setpci
-rwxr-xr-x	1	root	root	8431	2008-04-22	01:59	setupcon
l -rwxrwxrwx	1	root	root	4	2008-06-20	11:03	sh -> dash
l -rwxrwxrwx	1	root	root	4	2008-06-20	11:03	sh.distrib -> bash
-rwxr-xr-x	1	root	root	24488	2008-04-04	02:42	sleep
-rwxr-xr-x	1	root	root	48932	2008-04-04	02:42	stty
-rws -r-xr-x	1	root	root	25540	2008-04-02	21:08	su
-rwxr-xr-x	1	root	root	22312	2008-04-04	02:42	sync
-rwxr-xr-x	1	root	root	64	2007-11-15	06:49	zcat
-rwxr-xr-x	1	root	root	69	2007-11-15	06:49	zcmp
-rwxr-xr-x	1	root	root	4424	2007-11-15	06:49	zdiff
-rwxr-xr-x	1	root	root	64	2007-11-15	06:49	zegrep
-rwxr-xr-x	1	root	root	64	2007-11-15	06:49	zfgrep
-rwxr-xr-x	1	root	root	2015	2007-11-15	06:49	zforce
-rwx x -r x x	1	root	root	4893	2007-11-15	06:49	zgrep
-rwxr-xr-x	1	root	root	1733	2007-11-15	06:49	zless
-rwxr-xr-x	1	root	root	2416	2007-11-15	06:49	zmore
-r x x -r x x	1	root	root	4952	2007-11-15	06:49	znew

Annotations:

- Long listing of files with names starting with s or z (more on * later)
- "l" symbolic links (light blue)
- "-s" regular file with setuid bit set (red background)
- "-x" regular file with execute bit set (green)

Some special files in the /dev directory (Ubuntu)

A "b"
indicates a
Block
Special
Device

A "c"
indicates a
Character
Special
Device

```

rsimms@ulysses: ~
File Edit View Terminal Tabs Help
rsimms@ulysses:~$ ls -l /dev/sda
brw-rw---- 1 root disk 8, 0 2008-06-24 10:43 /dev/sda
rsimms@ulysses:~$ ls -l /dev/sda1
brw-rw---- 1 root disk 8, 1 2008-06-24 10:44 /dev/sda1
rsimms@ulysses:~$ ls -l /dev/tty1
crw----- 1 root root 4, 1 2008-06-24 10:44 /dev/tty1
rsimms@ulysses:~$ ls -l /dev/pts/0
crw----- 1 rsimms tty 136, 0 2008-06-24 10:53 /dev/pts/0
rsimms@ulysses:~$ clear
    
```

Special files (yellow
with black background)

Hard drives are **block** devices (data is transferred in large chunks for efficiency).

Terminals are **character** devices (data is transferred one character at a time).



Viewing the /boot directory (RH9)

```

root@frida:~
File Edit View Terminal Go Help
[root@frida root]# ls -l /boot
total 5127
-rw-r--r--  1 root  root    5824 Jan 24  2003 boot.b
-rw-r--r--  1 root  root    612 Jan 24  2003 chain.b
-rw-r--r--  1 root  root  44309 Feb 27  2003 config-2.4.20-6
drwxr-xr-x  2 root  root   1024 Jun  5  19:10 grub
-rw-r--r--  1 root  root 254430 Jun  5  18:47 initrd-2.4.20-6.img
-rw-r--r--  1 root  root   473 Jun  5  18:47 kernel.h
drwx-----  2 root  root  12288 Jun  5  11:45 lost+found
-rw-r--r--  1 root  root  23108 Feb 24  2003 message
-rw-r--r--  1 root  root  21282 Feb 24  2003 message.ja
lrwxrwxrwx  1 root  root     20 Jun  5  18:47 module-info -> module-info-2.4.20-6
-rw-r--r--  1 root  root  15436 Feb 27  2003 module-info-2.4.20-6
-rw-r--r--  1 root  root   640 Jan 24  2003 os2_d.b
lrwxrwxrwx  1 root  root     19 Jun  5  18:47 System.map -> System.map-2.4.20-6
-rw-r--r--  1 root  root  520099 Feb 27  2003 System.map-2.4.20-6
-rw-r--r--  1 root  root 3193468 Feb 27  2003 vmlinuz-2.4.20-6
lrwxrwxrwx  1 root  root     16 Jun  5  18:47 vmlinuz -> vmlinuz-2.4.20-6
-rw-r--r--  1 root  root 1122363 Feb 27  2003 vmlinuz-2.4.20-6
[root@frida root]#

```

"-" regular files (black)

"d" directories (blue)

The kernel

Symbolic link to kernel

The kernel (compressed)

Class Exercise

Do a long listing of the /boot directory: **ls -l /boot**

- Is *grub* a directory or a regular file?
- Is *vmlinuz-2.6.32-71.el6.i686* a directory or a regular file?

Write you answers in the chat window



Further classification of files



file command

Provides expanded information about files

- There are many different types of regular files:
 - Programs (binary)
 - Scripts (text)
 - Text files
 - Data files (binary)
- The **file** command attempts to classify files and give you more detailed information on the file contents.

*Tip: Use the **file** command to determine if a file is a text file and can be viewed with **cat**, **more**, **less**, **tail** ... etc commands.*

file command

Examples

Use the **file** command to determine if a regular file is text or binary

letter and /bin/uname are both regular files

```
/home/cis90/simben $ ls -l letter /bin/uname
-rwxr-xr-x. 1 root      root  26004 Dec  7  2011 /bin/uname
-rw-r--r--. 1 simben90 cis90  1044 Jul 20  2001 letter
```

```
/home/cis90/simben $ file letter
letter: ASCII English text
/home/cis90/simben $
```

The data portion of the letter file is text and can be viewed by cat, more, head, etc.

```
/home/cis90/simben $ file /bin/uname
/bin/uname: ELF 32-bit LSB executable, Intel 80386, version 1
(SYSV), for GNU/Linux 2.6.9, dynamically linked (uses shared
libs), for GNU/Linux 2.6.9, stripped
/home/cis90/simben $
```

The data portion of the /bin/uname file is binary and can be viewed with the xxd command



Using file command to further classify files

Long listings show basic file types in column 1
 "-"=regular file
 "d"=directory

```
/home/cis90/depot/filetypes $ ls -l
total 108
-rw-r--r--. 1 rsimms cis90 8983 Aug 1 18:49 Adjective.frm
-rw-r--r--. 1 rsimms cis90 5976 Aug 1 18:49 Adjective.MYD
-rw-r--r--. 1 rsimms cis90 2048 Aug 1 18:49 Adjective.MYI
-rw-r--r--. 1 rsimms cis90 10240 Aug 1 18:49 backup.tar
-rw-r-----. 1 rsimms cis90 191 Aug 1 18:49 bash_profile
-rwxr-----. 1 rsimms cis90 4846 Aug 1 18:49 cprog
-rwxr-----. 1 rsimms cis90 4846 Aug 1 18:49 go-cprog
-rw-r--r--. 1 rsimms cis90 119 Aug 1 18:49 letter
-rw-r-----. 1 rsimms cis90 2968 Aug 1 18:49 mbox
-rw-r--r--. 1 rsimms cis90 34611 Aug 1 18:49 rich-260x216.jpg
-rwxr-xr-x. 1 rsimms cis90 445 Aug 1 18:49 runit
drwxr-xr-x. 2 rsimms cis90 4096 Aug 1 18:40 travel
```

Output from the file command provides additional file classification information

```
/home/cis90/depot/filetypes $ file *
Adjective.frm: MySQL table definition file Version 9
Adjective.MYD: DBase 3 data file (33517822 records)
Adjective.MYI: MySQL MISAM compressed data file Version 1
backup.tar: POSIX tar archive (GNU)
bash_profile: ASCII English text
cprog: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
dynamically linked (uses shared libs), for GNU/Linux 2.2.5, not stripped
go-cprog: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
dynamically linked (uses shared libs), for GNU/Linux 2.2.5, not stripped
letter: ASCII English text
mbox: ASCII mail text
rich-260x216.jpg: JPEG image data, JFIF standard 1.02
runit: POSIX shell script text executable
travel: directory
```

Class Activity

Classify the following these files in your home directory:

- uhistory
- letter
- Poems
- timecal

- Which is a bash script?

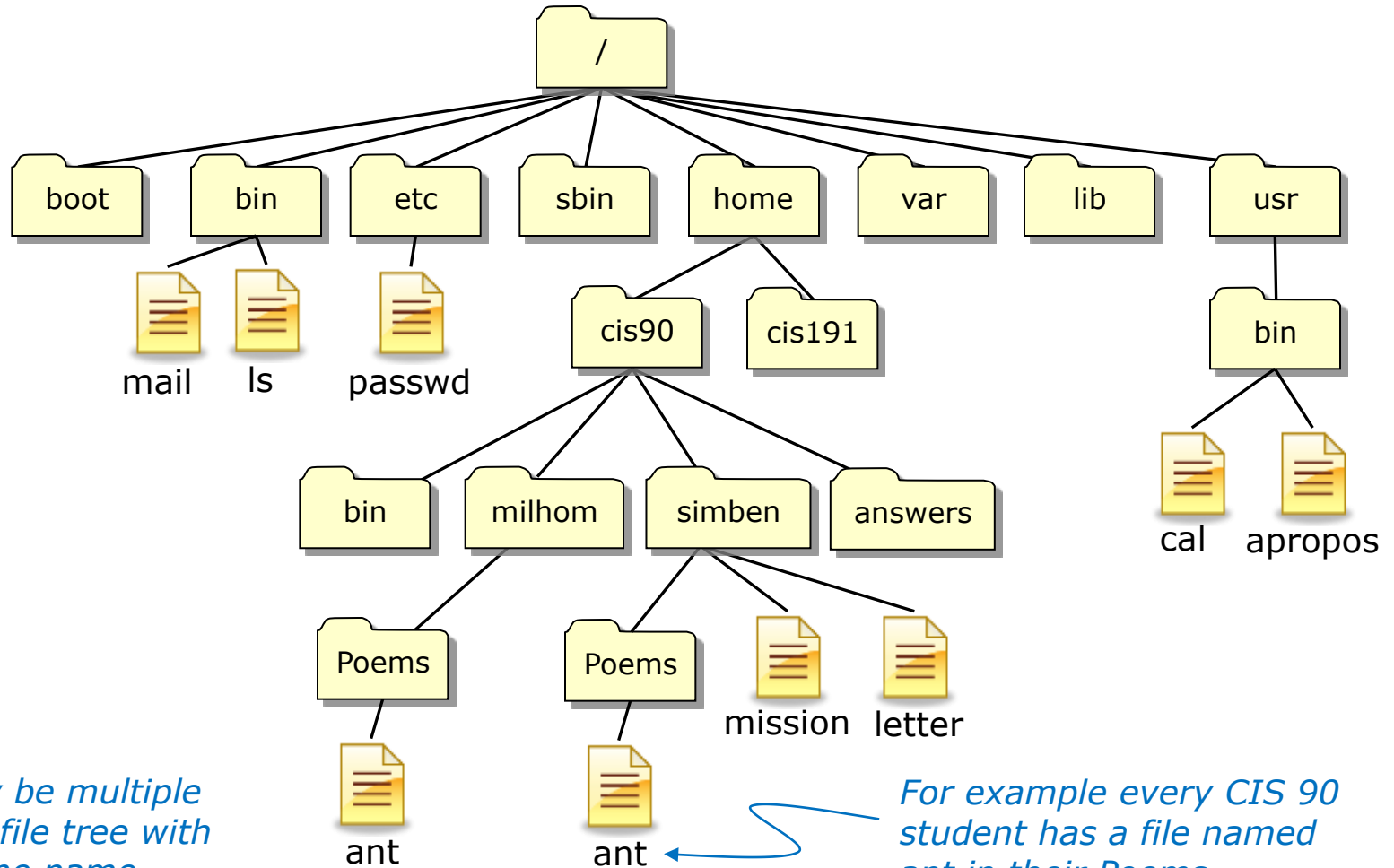
Write your answer in the chat window



Pathnames

The need for pathnames

Question: How can we unambiguously specify any file or directory in the file tree?

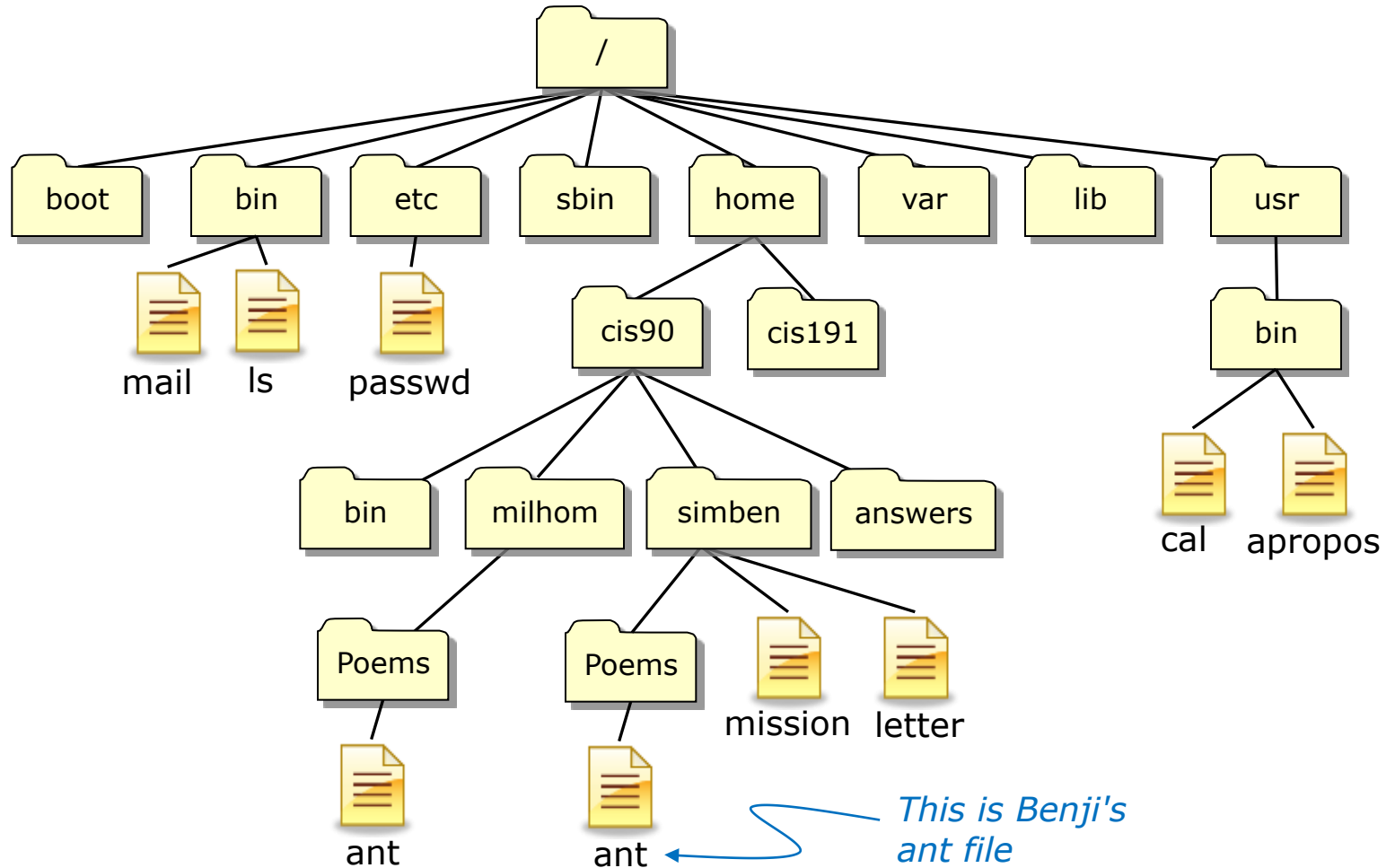


There may be multiple files in the file tree with the same name.

For example every CIS 90 student has a file named ant in their Poems directory

The need for pathnames

Answer: We use **absolute** or **relative** pathnames





Pathnames

What the heck are they?

A pathname is a precise way to specify exactly any file or directory in the file tree.

- An **absolute pathname** specifies the path from the **top of the tree** to the target directory or file.
- A **relative pathname** specifies the path from your **current location** to the target directory or file.

Understanding pathnames is critical because they are used as arguments on all commands that deal with files and directories.



Absolute Pathnames

Absolute Pathnames

An **absolute pathname** specifies the path from the **top of the tree** to the target directory or file.

Examples:

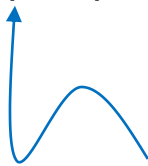
/home/cis90/simben/Poems/ant (file)

/boot (directory)

/usr/bin/cal (file)

/home/cis90/bin/ (directory)

/bin/mail (file)

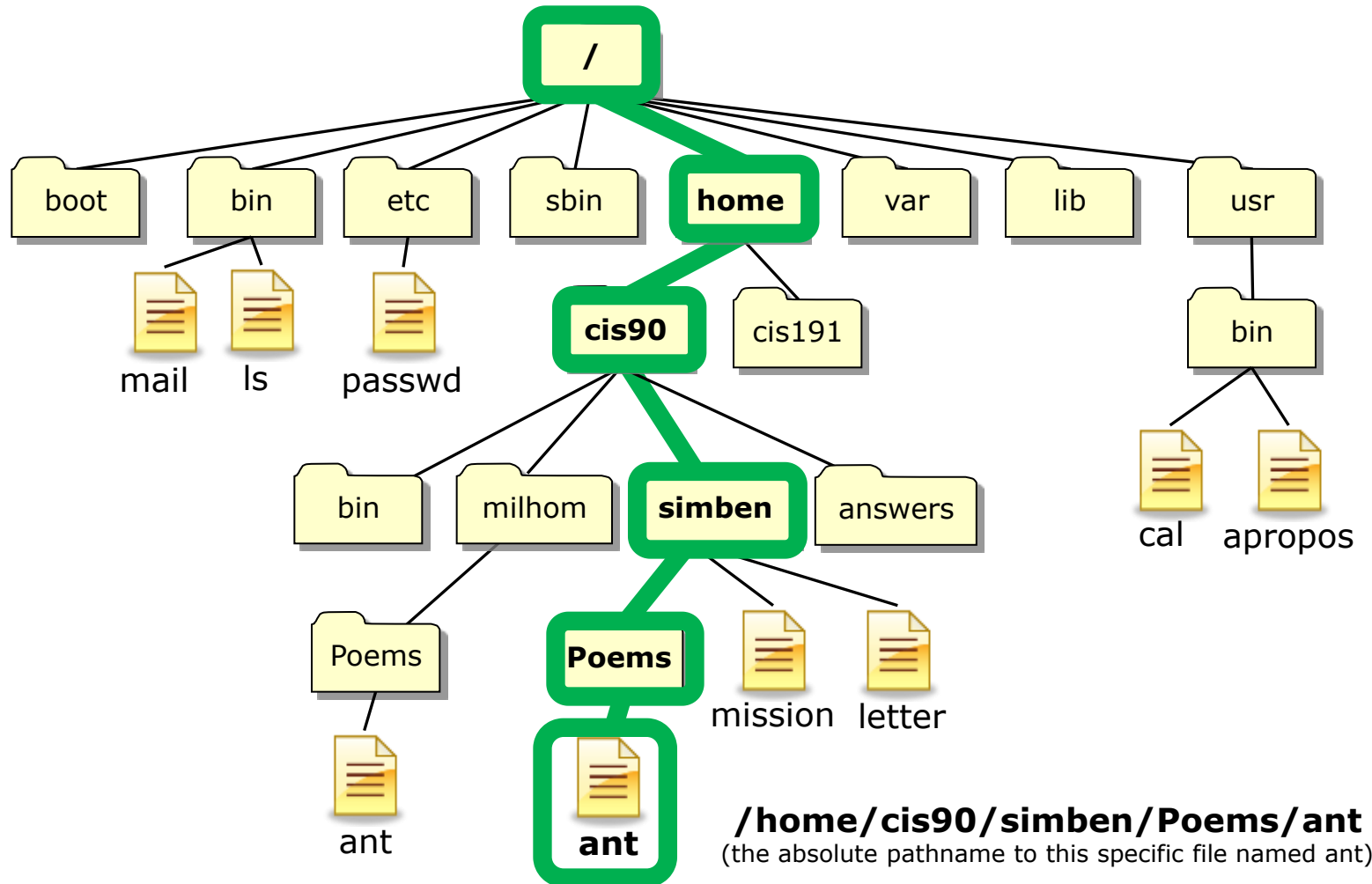


*** *Important* ***

Notice all absolute pathnames start with a / (forward slash) which represents the top of the file tree

Example Absolute Pathname

An **absolute pathname** specifies the path from the **top of the tree** to the target directory or file.

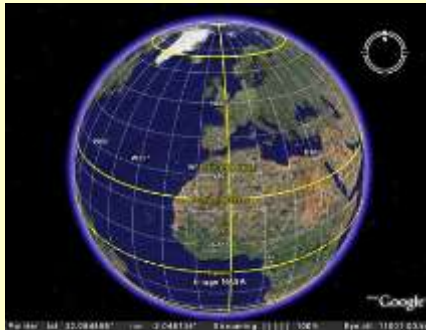


Absolute Pathname Analogy

Where is Watsonville Airport using latitude and longitude?

An analogy ...

<http://www.engineeringtoolbox.com/>



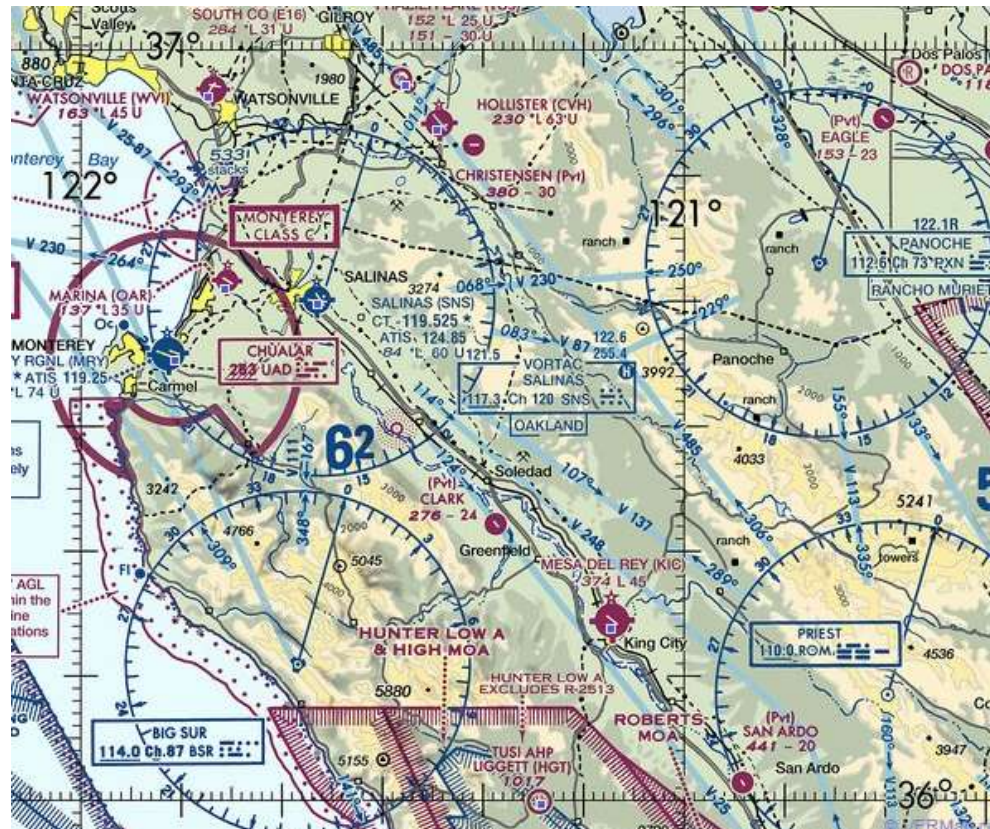
Latitude is measured in degrees north or south of the equator.

Longitude is measured in degrees east or west of the prime meridian.

Watsonville Airport

Latitude: 36-56'09" N

Longitude: 121-47'23" W



Latitude and longitude designate a target destination independent of your current location

Class Activity - absolute pathnames

Show the last two lines of your ant file using an absolute pathname

```
/home/cis90/simben $ tail -n2 /home/cis90/simben/Poems/ant
'till one who seemed the least
of all absorbed my whole of mind.
```

*replace with your
own home
directory name*

Show the last two lines of Homer's ant file using an absolute pathname

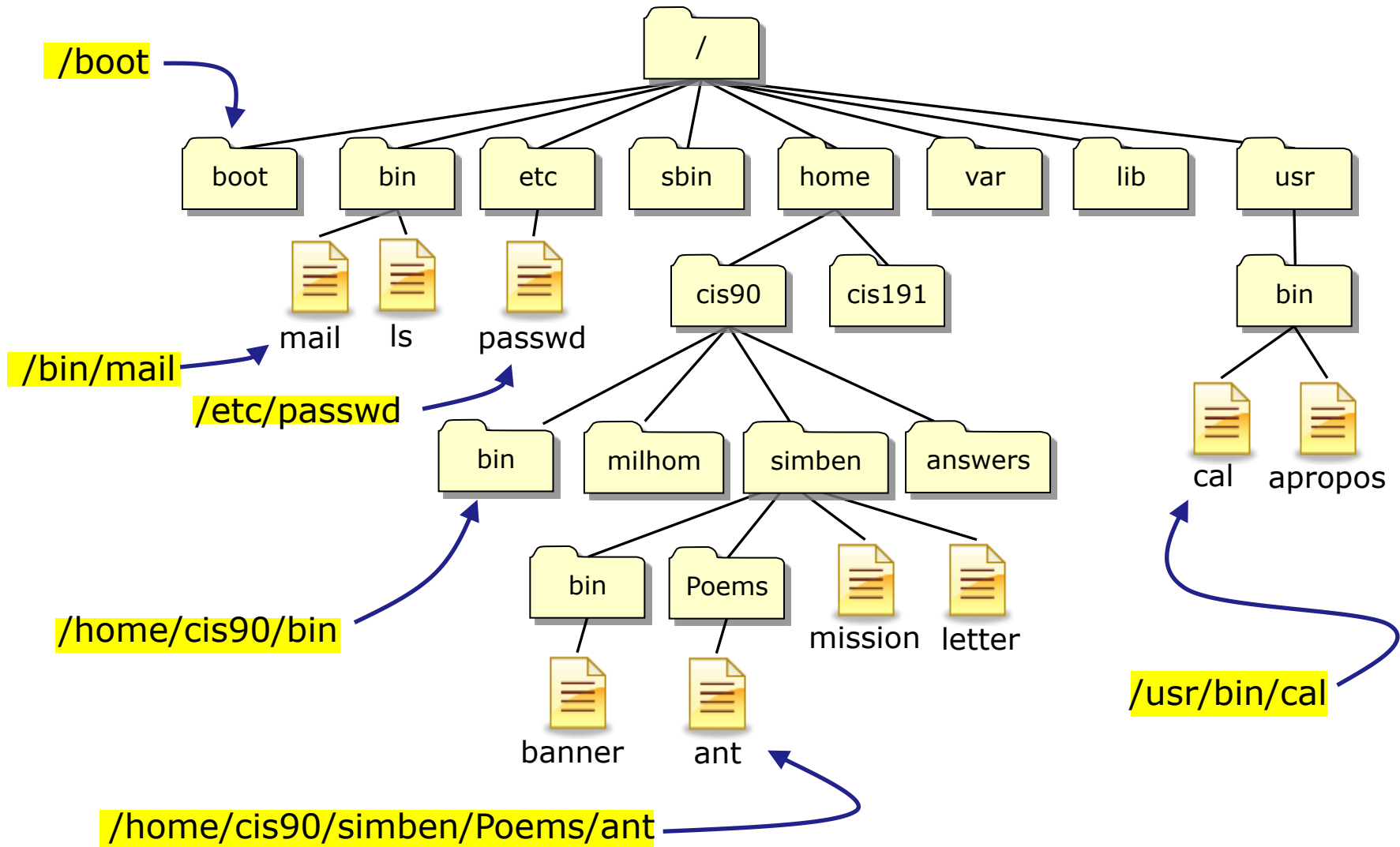
```
/home/cis90/simben $ tail -n2 /home/cis90/milhom/Poems/ant
'till one who seemed the least
of all absorbed my whole of mind.
```

Show the last two lines of your ant file using a variable for part of an absolute pathname

```
/home/cis90/simben $ echo $HOME/Poems/ant
/home/cis90/simben/Poems/ant
/home/cis90/simben $ tail -n2 $HOME/Poems/ant
'till one who seemed the least
of all absorbed my whole of mind.
```

Absolute Pathnames

Some more example absolute pathnames



Absolute Pathnames

Some example absolute pathnames being used as arguments

```
ls /bin /sbin /usr/bin /usr/sbin
```

```
file /usr/bin/cal
```

```
cd /home/cis90/simben/Poems/Shakespeare
```

```
tail -n1 /etc/passwd
```

```
more /home/cis90/simben/bigfile
```

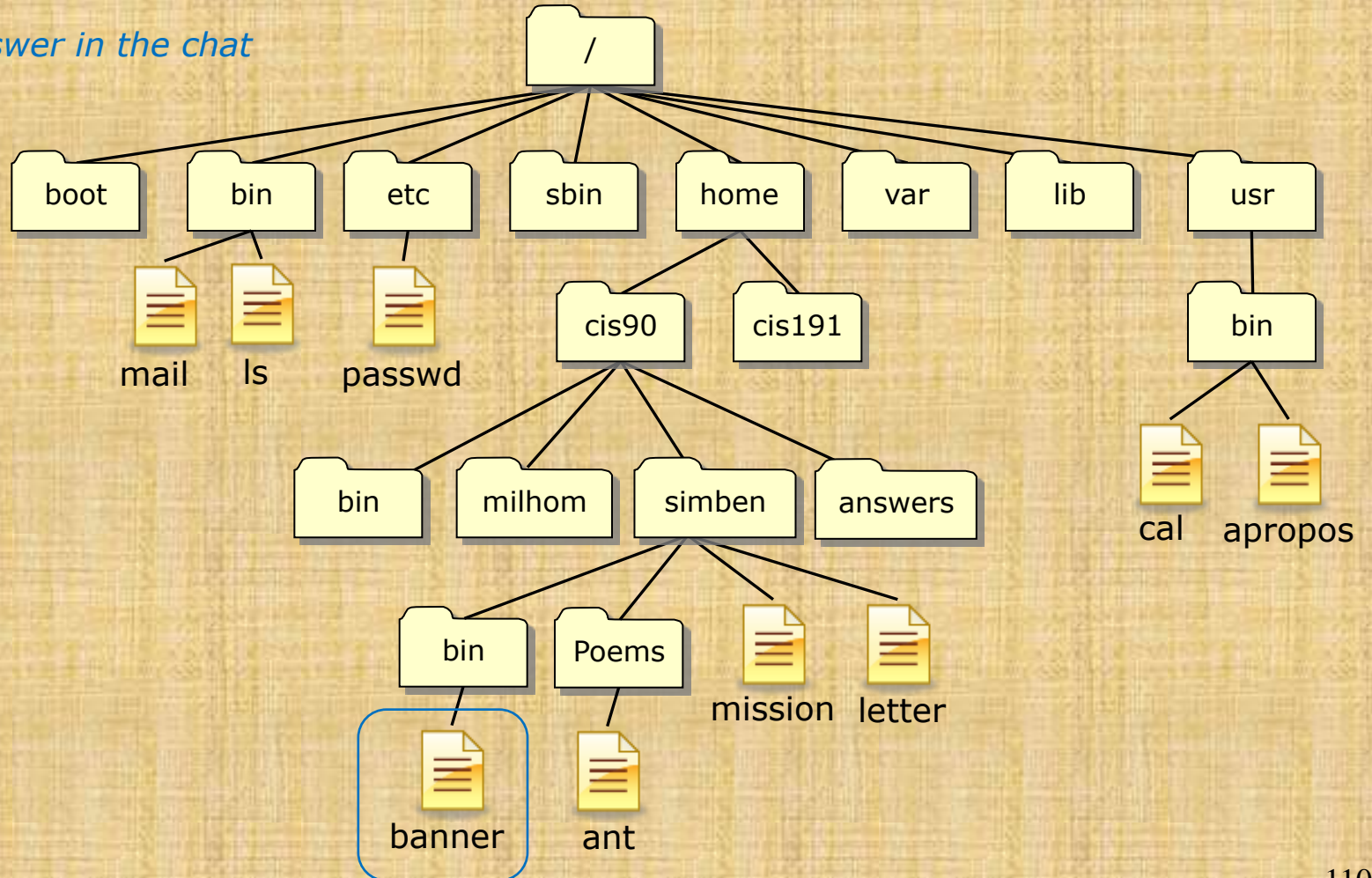
**** Important ****

Notice all absolute pathnames start with a / (forward slash) which represents the top of the file tree

Activity - identify an absolute pathname

Question: *what is the absolute pathname to Benji's banner file?*

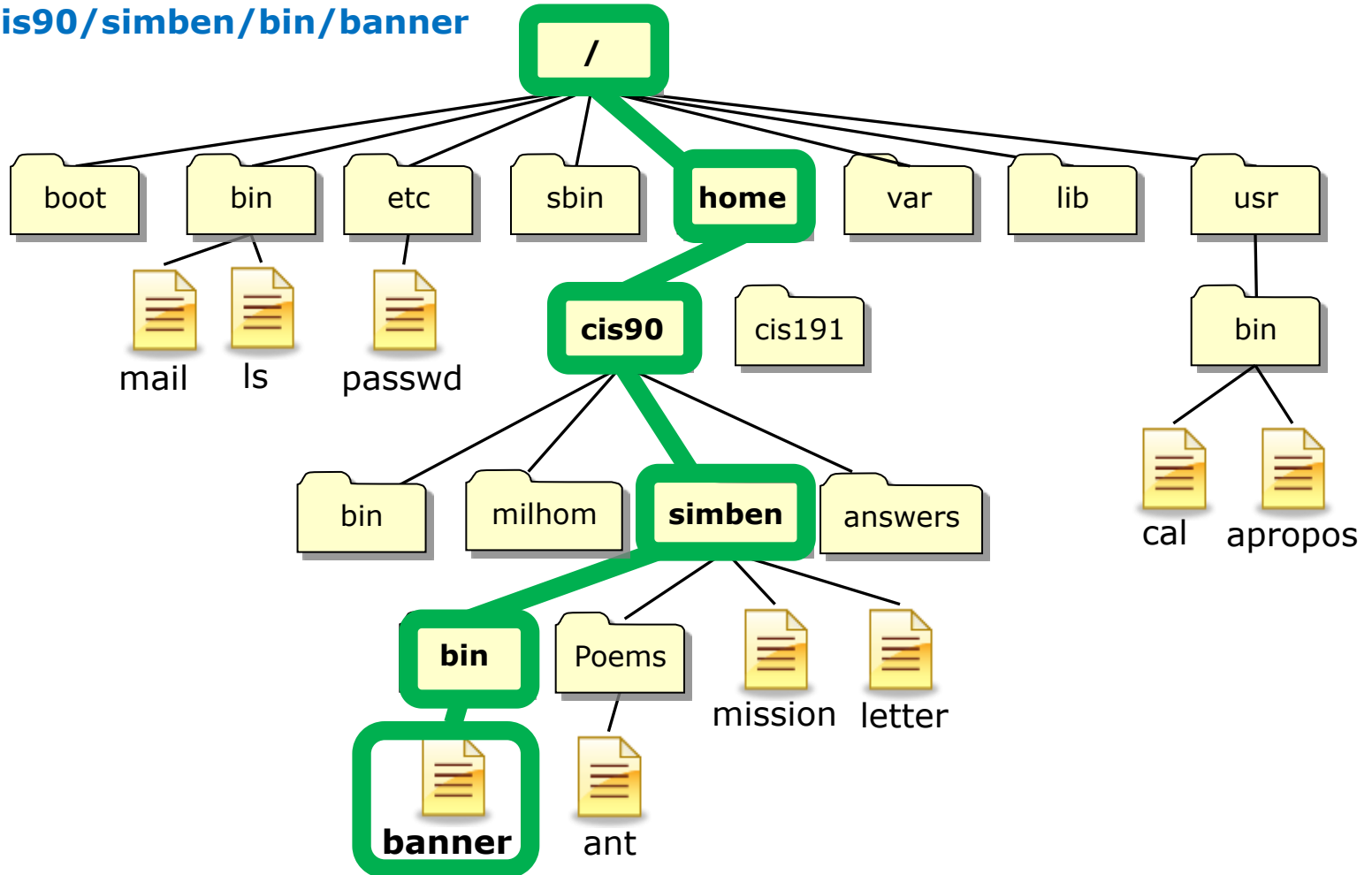
(write your answer in the chat window)



Question: what is the absolute pathname to Benji's banner file?

Answer:

/home/cis90/simben/bin/banner



`/home/cis90/simben/bin/banner`

Translation of this absolute pathname in English:

Start at the top of the tree and descend into the *home* directory, then descend into the *cis90* directory, then descend into the *simben* directory, then descend into the *bin* directory, there you will find the *banner* file.



Relative Pathnames

Relative Pathnames

A **relative pathname** specifies the path from your current directory to the target directory or file.

Examples:

ant	(file)
Poems/Shakespeare/sonnet5	(file)
../mission	(file)
../bin/	(directory)
../../boot/vmlinuz-2.6.18-164.el5	(file)

*** *Important* ***

Note that relative pathnames do NOT start with a /

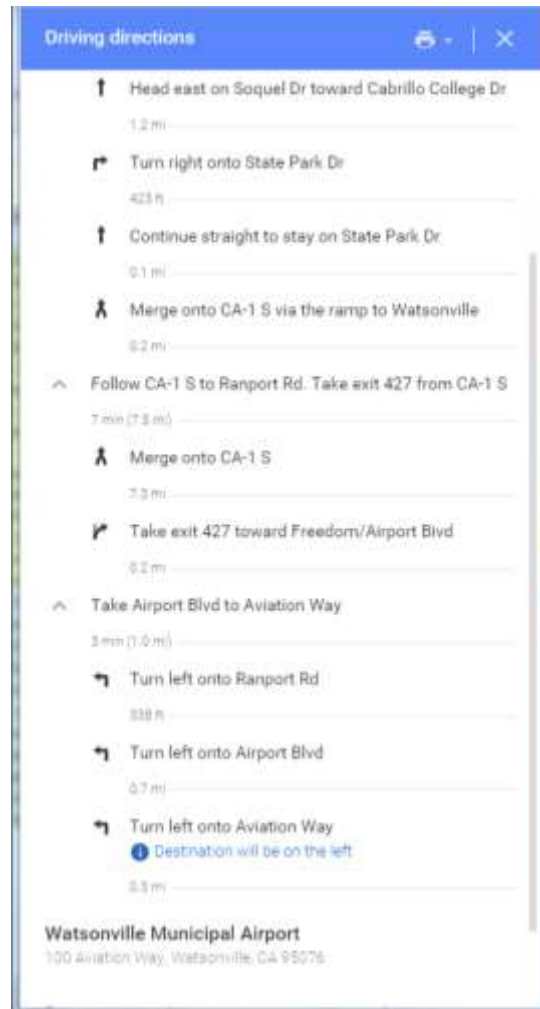
Relative Pathname Analogy

How do I get from Cabrillo College to Watsonville Airport using Google Maps?

An analogy ...



Google Maps show a driving route from your current location to a target destination

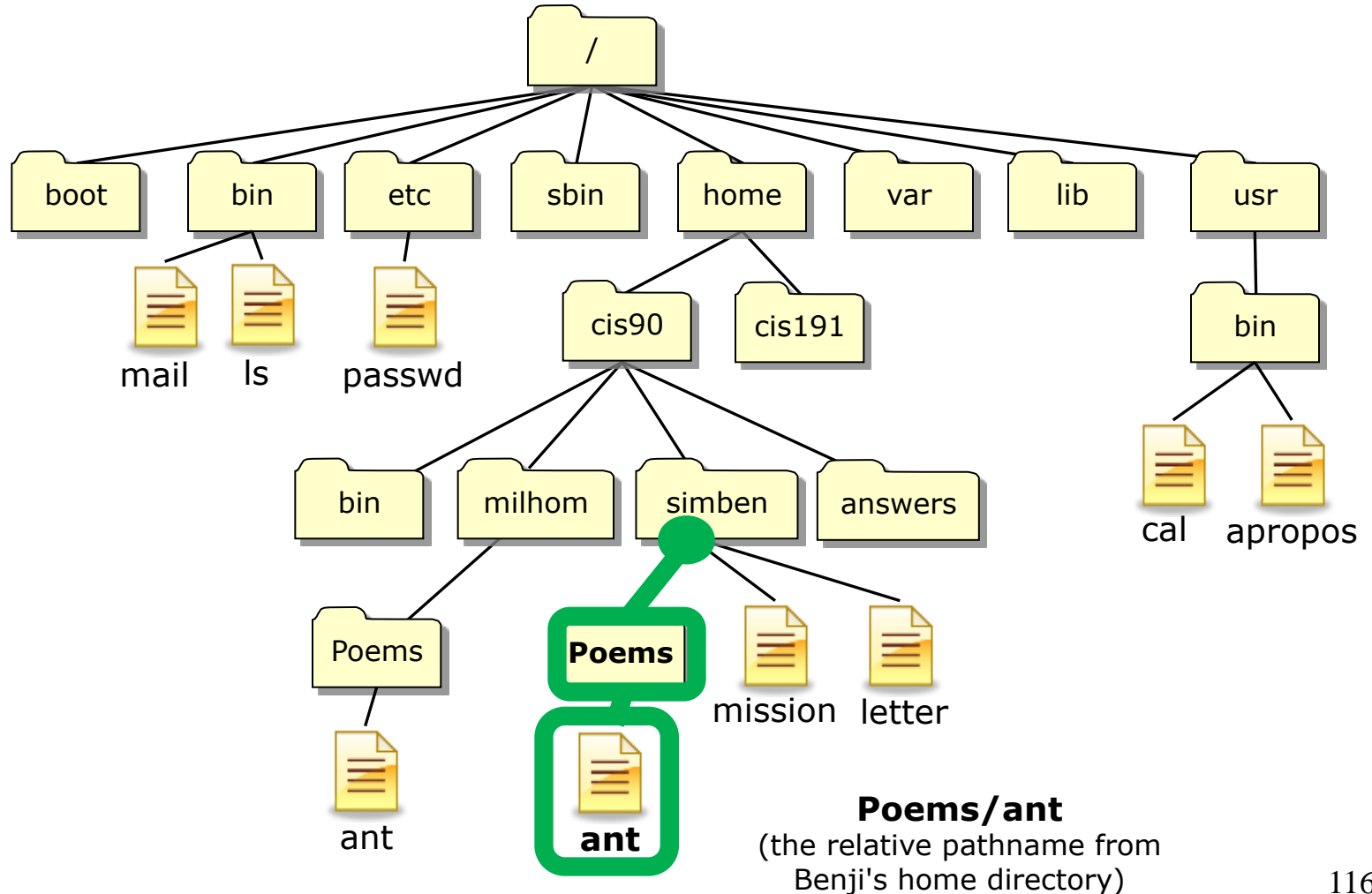


Google Maps instructions to a target destination depend on your starting location.



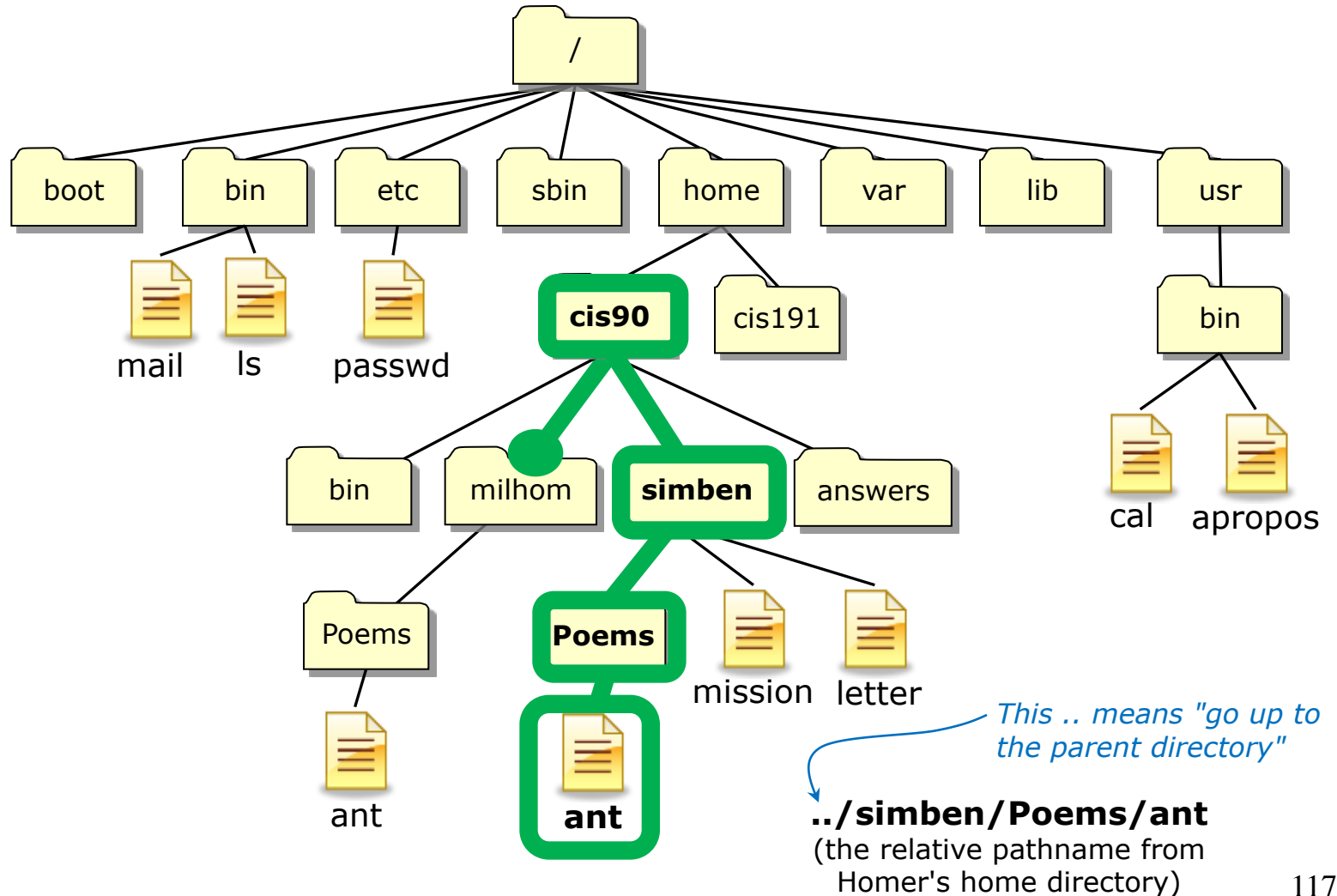
Relative Pathnames

A **relative** pathname specifies a path from our current location in the tree all the way to the specific file.



Relative Pathnames

A **relative** pathname specifies a path from our current location in the tree all the way to the specific file.



Class Activity - relative pathnames

Show the first three lines of your ant file using a relative pathname

```
/home/cis90/simben $ cd
/home/cis90/simben $ head -n3 Poems/ant
    Death of an Ant
```

Go to your home directory if you are not already there

With a magnifying glass

Show the first three lines of Homer's ant file using a relative pathname

```
/home/cis90/simben $ head -n3 ../milhom/Poems/ant
    Death of an Ant
```

.. means to go up one level in the tree to the parent directory of the current working directory

With a magnifying glass

Show the first three lines of your Shakespeare sonnet5 file

```
/home/cis90/simben $ head -n3 Poems/Shakespeare/sonnet5
Those hours that with gentle work did frame
The lovely gaze where every eye doth dwell
Will play the tyrants to the very same,
```

Relative Pathnames

Using relative pathnames as command arguments



#Cabrillo

Examples of using relative pathnames as command arguments:

ls -l ant

file ../../../../bin/mail

cd Poems/Blake

head ../bin/check3

file Poems/Shakespeare/sonnet4

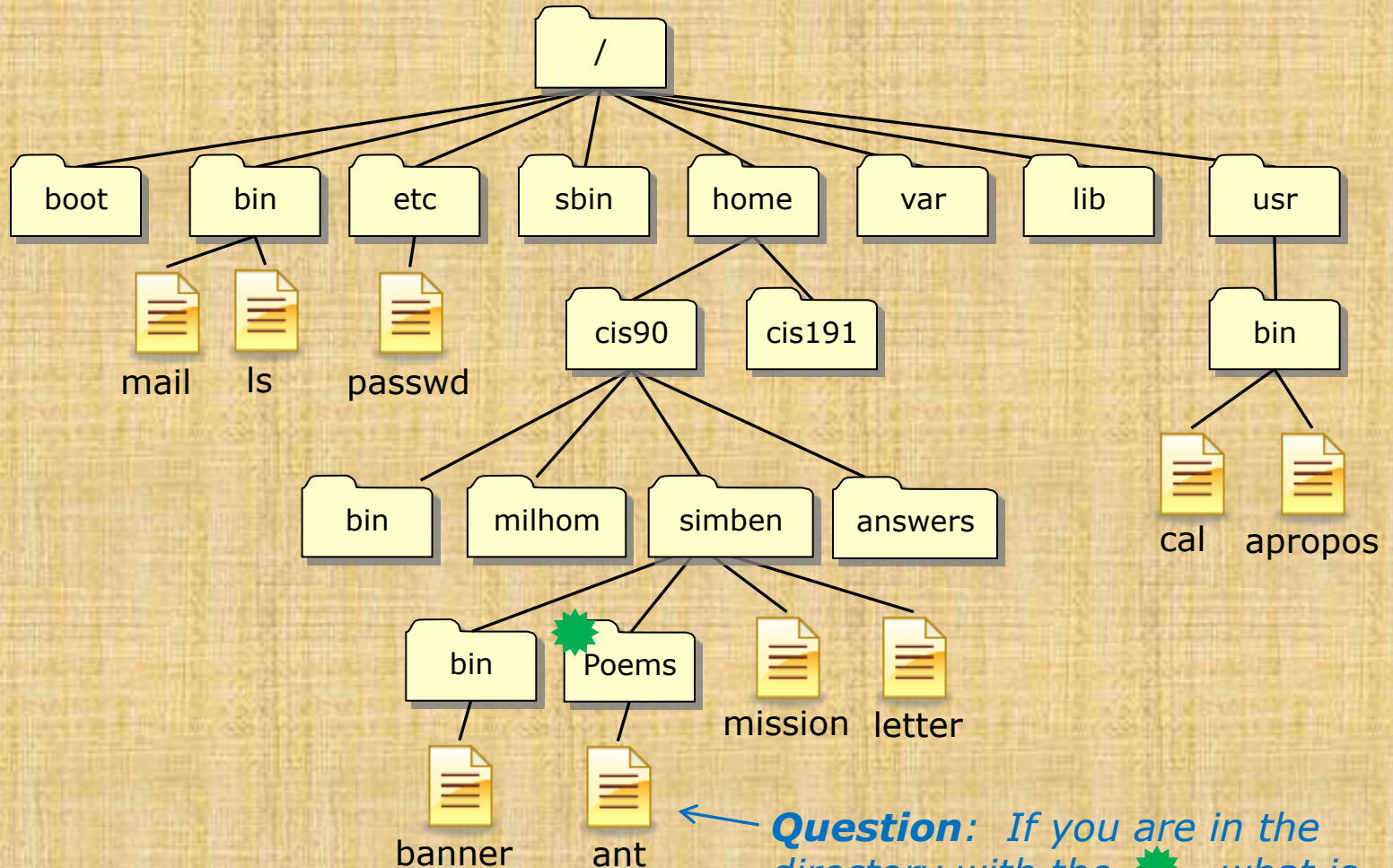
cd Poems/Shakespeare


The .. is used to represent the parent directory

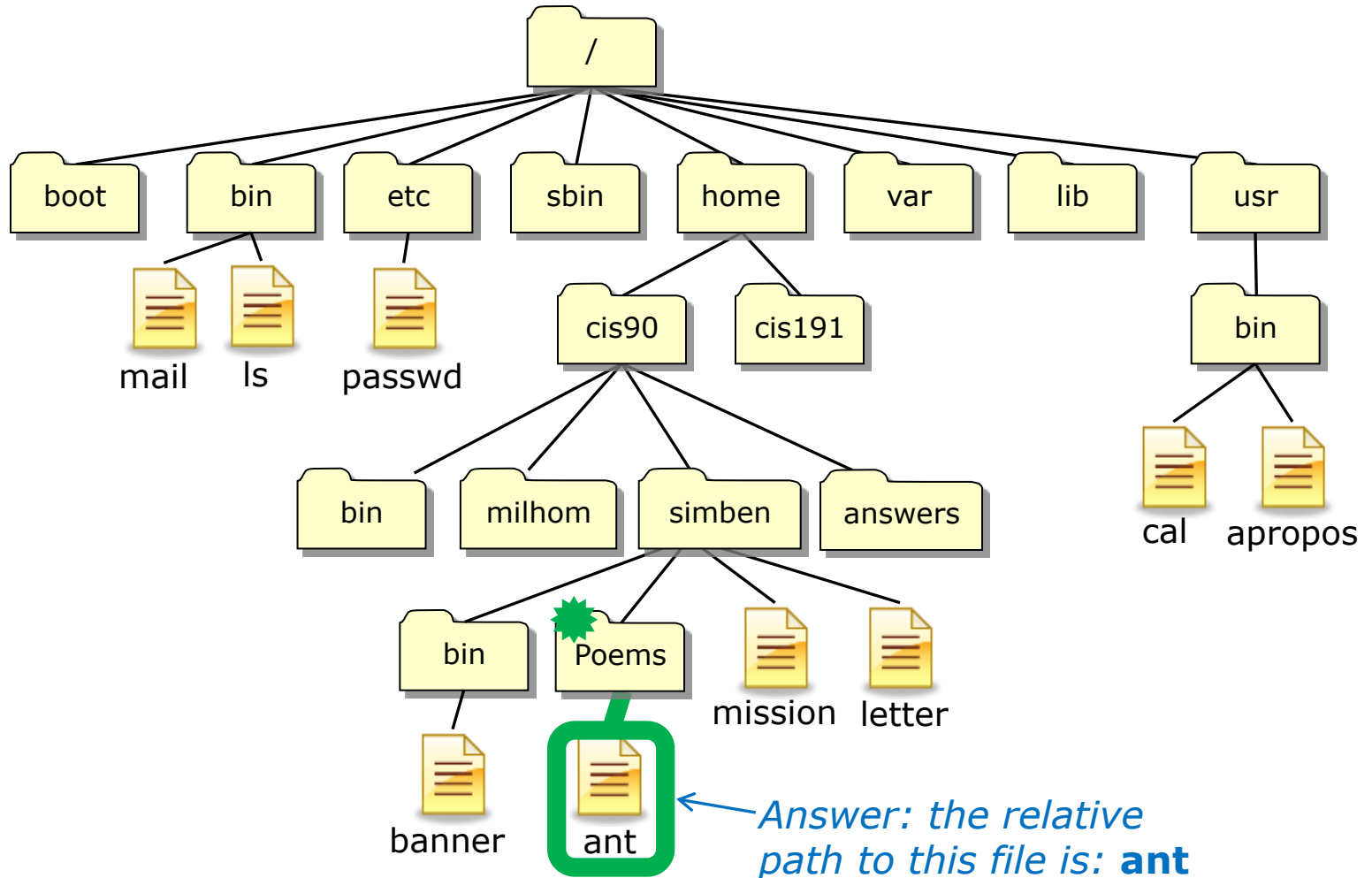
***** Important *****

Notice that these pathnames do NOT start with the /

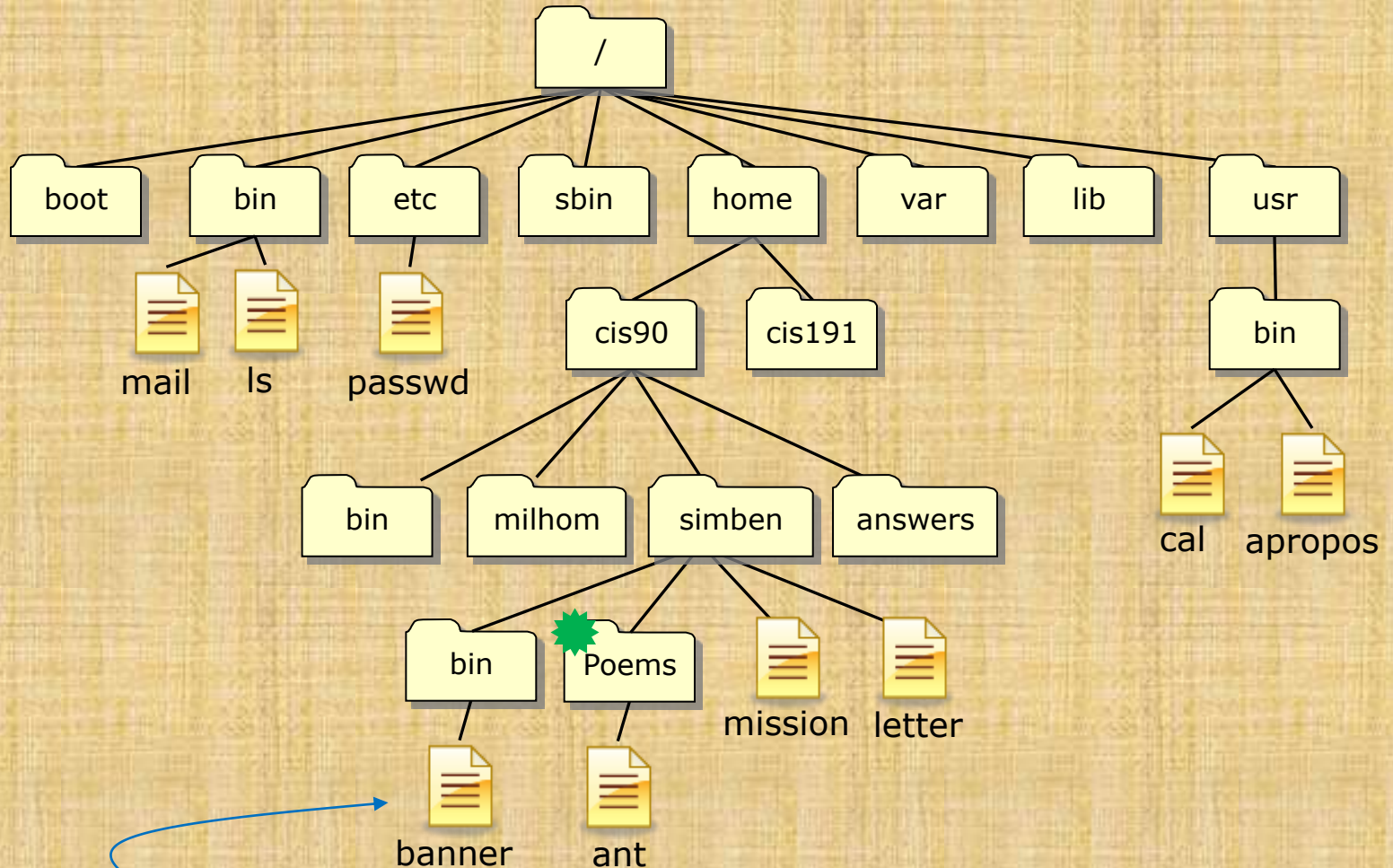
Activity - identify a relative pathname




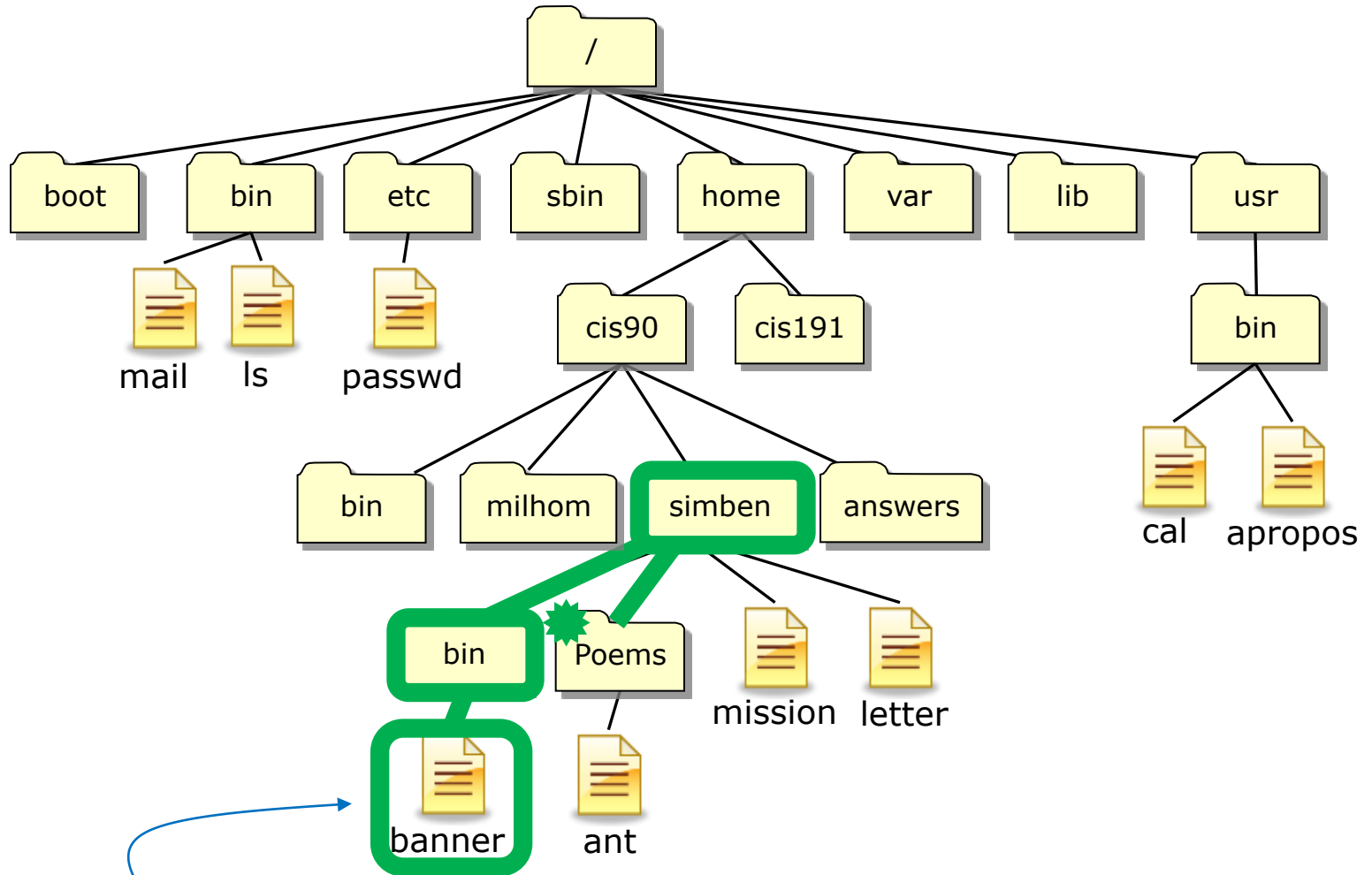
Question: If you are in the directory with the , what is the relative path to this file?



Activity - identify a relative pathname



Question: If you are in the directory with the , what is the relative path to this file?



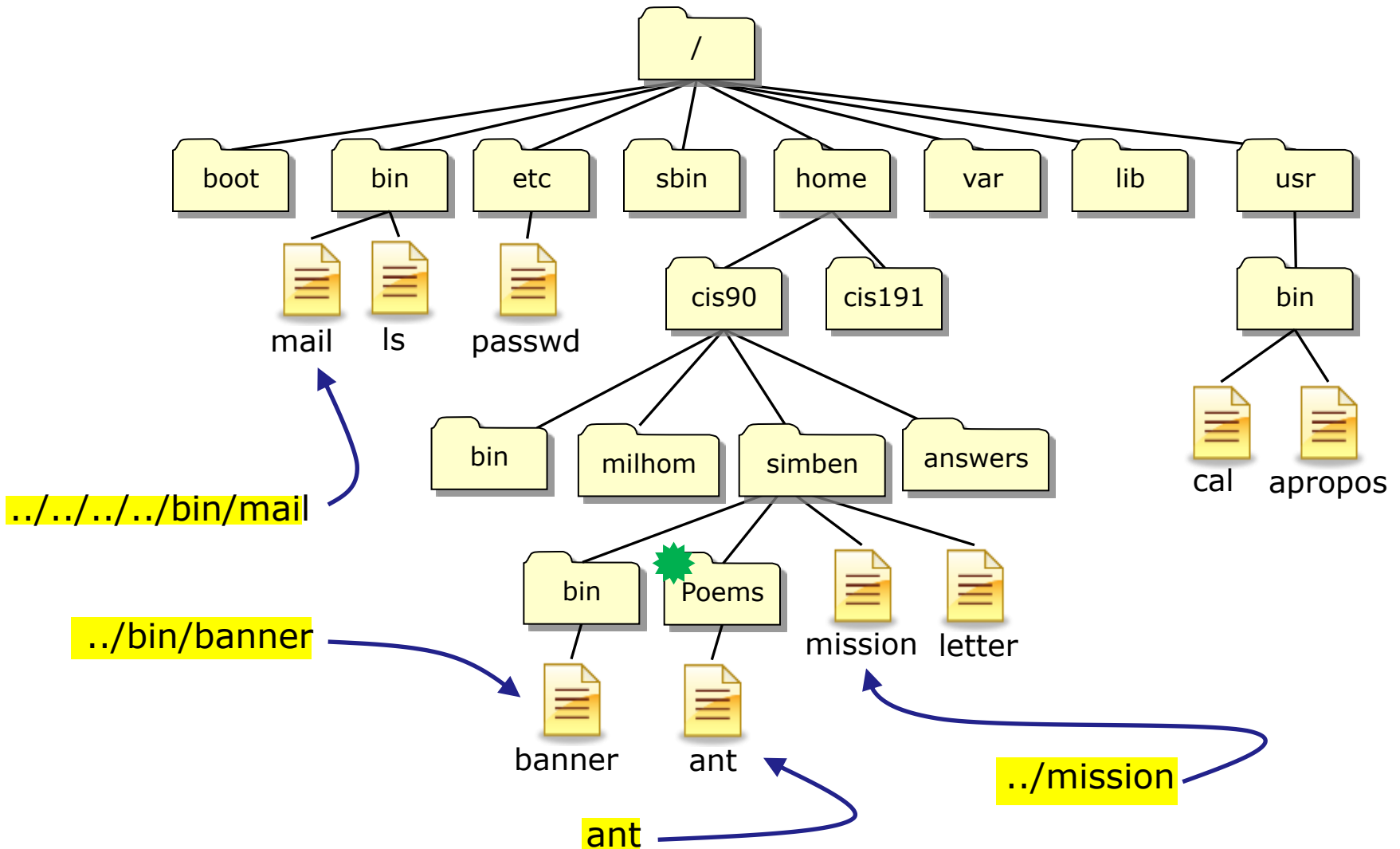
Answer: The relative path to this file is: **../bin/banner**

../bin/banner

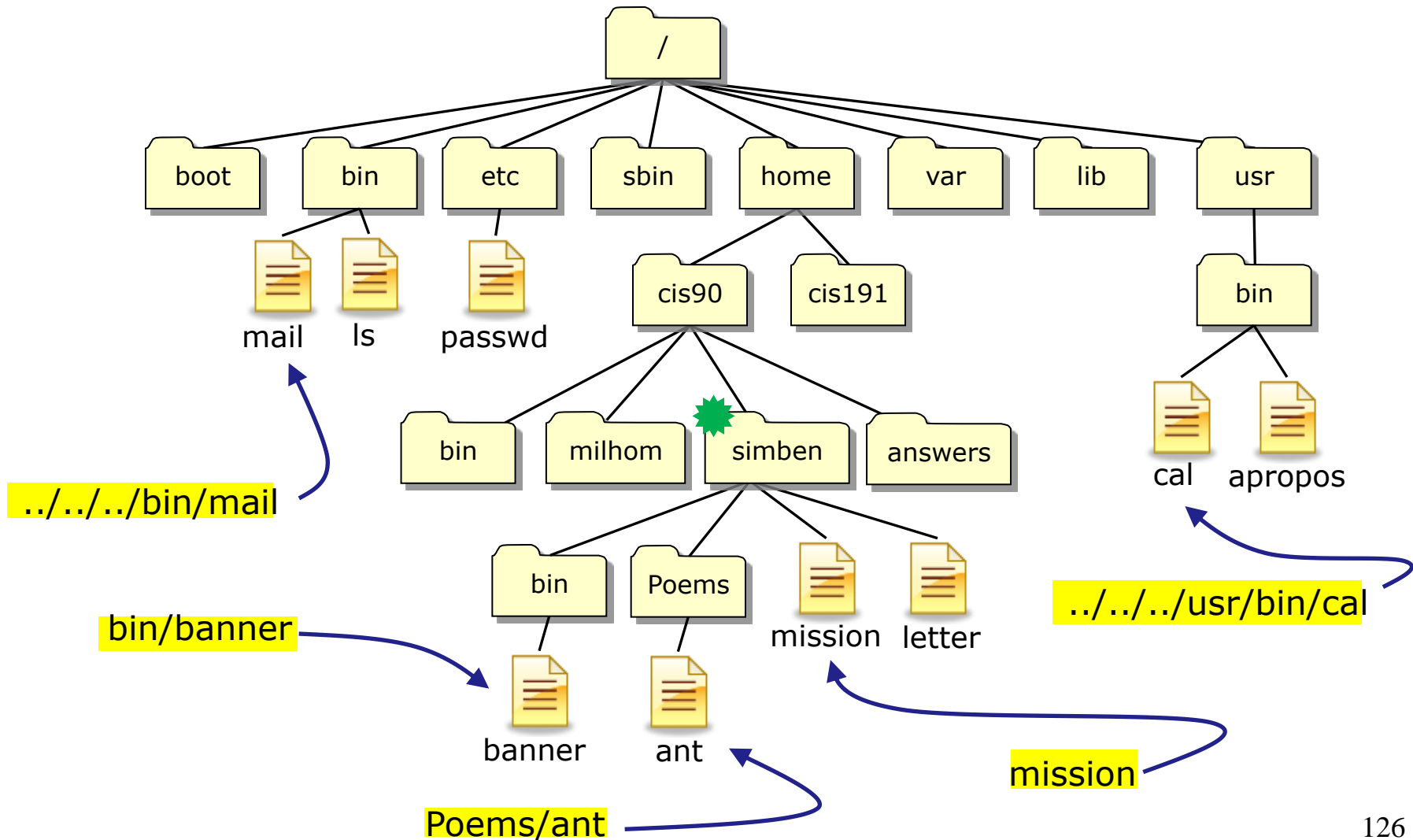
Translation of this relative pathname in English:

Starting in your current directory, go up one level to the parent directory, then descend into the *bin* directory, there you will find the *banner* file.

Some example relative pathnames (from the directory marked with a )



Some example relative pathnames (from the directory marked with a )



Class Exercise

From your home directory:

- List the */etc/passwd* file using a relative pathname

```
/home/cis90/simben $ ls -l ../../../etc/passwd  
-rw-r--r--. 1 root root 10162 Feb 18 09:26 ../../../../etc/passwd
```

- List the */etc/passwd* file using a absolute pathname

```
/home/cis90/simben $ ls -l /etc/passwd  
-rw-r--r--. 1 root root 10162 Feb 18 09:26 /etc/passwd
```

Sometimes it's easier to specify a filename using an absolute pathname



Heads up on a future test question

Question:

What is the absolute pathname of /etc/passwd?

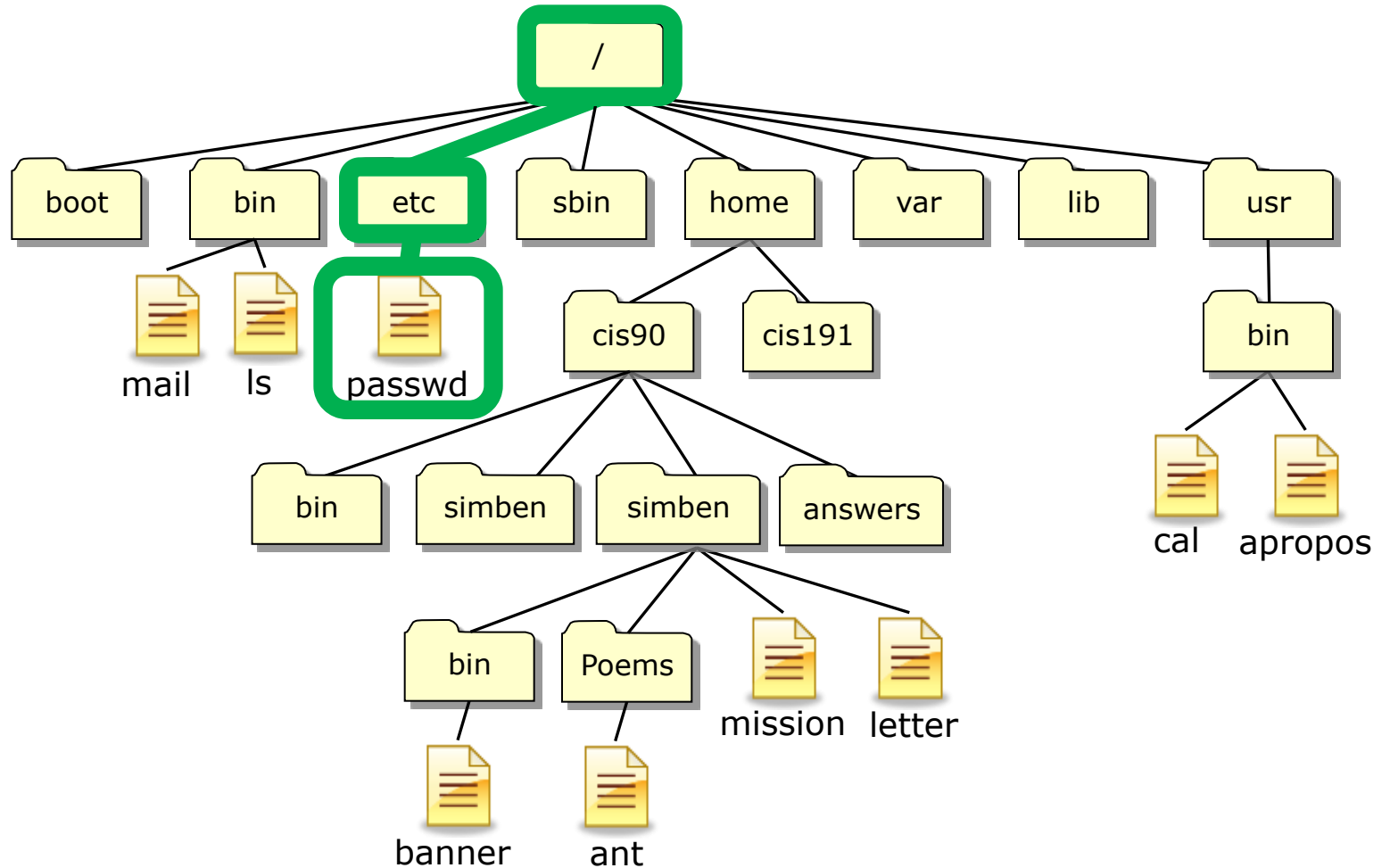
Answer:

/etc/passwd

What is the color of Washington's white horse?

Question: What is the absolute pathname of /etc/passwd?

Answer: /etc/passwd





/

■

■ ■

~

/ . . ~

/ by itself is the root or "slash" directory, the top of the tree, not to be confused with the root user's home directory (/root)

/ at the beginning of a pathname indicates an absolute path

/ at the end of a filename indicates it is a directory

.. is always your current **parent** directory

. is always your current directory ("here")

~ is always your home directory

Note:

. and .. are hidden files since they start with a "."

Hidden files don't show up in ls listings unless the -a option is used

Example Sequence using / .. and ~

1. Change to your Poems/Blake directory using a relative pathname

```
|/home/cis90/simben $ cd Poems/Blake/
|/home/cis90/simben/Poems/Blake $
```

2. List the directories in the / directory using an absolute pathname

```
|/home/cis90/simben/Poems/Blake $ ls /
bin    dev    home  lost+found  misc  net  proc  sbin    srv  tftpboot  u    var
boot  etc    lib   media      mnt   opt  root  selinux  sys  tmp      usr
```

3. List the directories in your current parent directory using ..

```
|/home/cis90/simben/Poems/Blake $ ls ..
ant  Blake  nursery  Shakespeare  twister  Yeats
```

4. List the directories in your current directory using .

```
|/home/cis90/simben/Poems/Blake $ ls .
jerusalem  tiger
```

5. List the files in your home directory using ~

```
|/home/cis90/simben/Poems/Blake $ ls ~
1976          empty          Lab2.0  Miscellaneous  proposal3  text.fxd
android       Hidden         Lab2.1  mission        scott      timecal
bigfile       lab01.graded  letter  Poems          small_town  uhistory
bin           lab01-submitted  log     proposal1     spellk     what_am_i
dead.letter   lab02.graded  mbox    proposal2     text.err
```



Shell tips

(review)

bash shell tip

tab completes

- It can be tedious typing in long pathnames.
- Since bash knows the names of the files you only have to type just enough characters to uniquely specify a name and then the tab key can be pressed to complete them.
- Example: the black characters were typed by the user, the green ones were typed by bash:

```
ls /home/cis90/simben/Poems/Shakespeare/
```





bash shell tip

command history and editing

- It can be tedious re-typing a long command to fix a typo.
- Since bash knows the commands you have previously entered, just use the up and down arrows to re-type a previous command.
- When the command you want appears, use the home, right or left arrow keys to go where you want to make the correction. New text can be inserted and old text deleted or backspaced over.
- Example: The ls command was mis-typed as la:

```
/home/cis90/simmsben $ la /home/cis90/simmsben/Poems/Shakespeare/  
-bash: la: command not found
```

  then fix typo

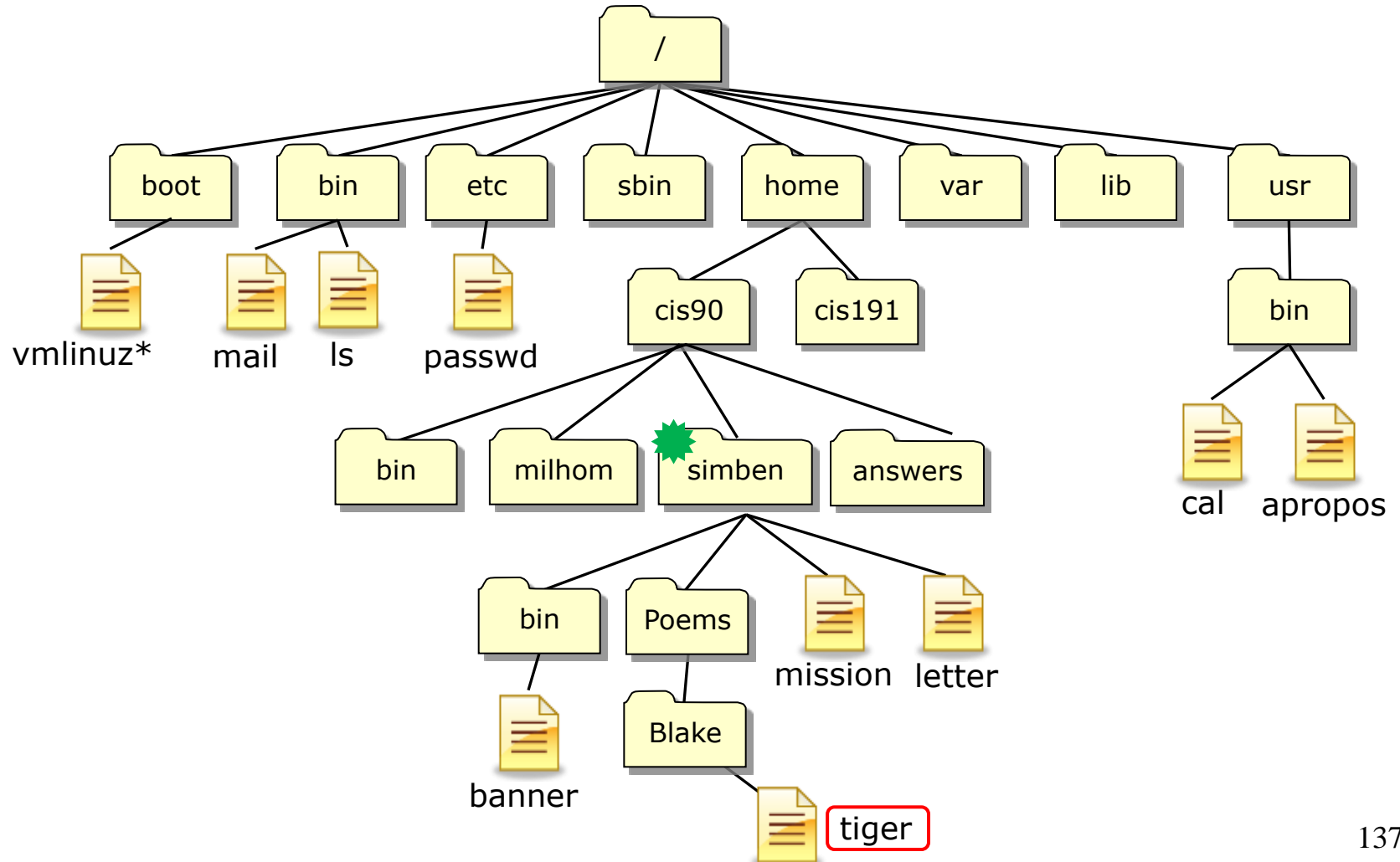
```
/home/cis90/simmsben $ ls /home/cis90/simmsben/Poems/Shakespeare/  
sonnet1    sonnet11   sonnet17   sonnet26   sonnet35   sonnet5    sonnet9  
sonnet10   sonnet15   sonnet2    sonnet3    sonnet4    sonnet7  
/home/cis90/simmsben $
```



Using pathnames as arguments

Task: cat the tiger file from your home directory

How can we do this?



Task: cat the tiger file from your home directory

Option 1: "Navigate" to the directory then cat the file

```
/home/cis90/simben $ cd      start in our home directory
```

```
/home/cis90/simben $ ls      see what's there
```

```
bigfile      Hidden      log          proposal1   text.err
bin          lab01.graded mbox         proposal2   text.fxd
countargs    Lab2.0      Miscellaneous proposal3    timecal
dead.letter  Lab2.1      mission      small_town  uhistory
empty        letter      Poems        spellk      what_am_i
```

```
/home/cis90/simben $ cd Poems/ descend into the Poems directory
```

```
/home/cis90/simben/Poems $ ls see what's there
```

```
ant Blake nursery Shakespeare twister Yeats
```

```
/home/cis90/simben/Poems $ cd Blake/ descend into the Blake directory
```

```
/home/cis90/simben/Poems/Blake $ ls see what's there
```

```
jerusalem tiger
```

```
/home/cis90/simben/Poems/Blake $ cat tiger
```

```
Tiger, Tiger burning bright
In the forest of the night,
What immortal hand or eye
Dare frame thy fearful symmetry?
```


Task: cat the tiger file from your home directory

Option 2: Use a relative pathname

```
/home/cis90/simben $ cat Poems/Blake/tiger  
Tiger, Tiger burning bright  
In the forest of the night,  
What immortal hand or eye  
Dare frame thy fearful symmetry?  
/home/cis90/simben $
```

Task: cat the tiger file from your home directory
Option 3: Use an absolute pathname

```
/home/cis90/simben $ cat /home/cis90/simben/Poems/Blake/tiger  
Tiger, Tiger burning bright  
In the forest of the night,  
What immortal hand or eye  
Dare frame thy fearful symmetry?  
/home/cis90/simben $
```

Task: cat the tiger file from your home directory

Option 4: communicating with the shell using ESP

```
/home/cis90/simben $ cat tiger  
cat: tiger: No such file or directory  
/home/cis90/simben $
```

ESP is not an option!

There is no tiger file in the /home/cis90/simben directory.

There are over 40 tiger files on Opus.

If you don't give the shell a correct pathname that unambiguously specifies the location of a file in the file tree you should expect this error.

Don't expect the shell to read your mind as to which file in the file tree you are thinking about!



Task: cat the tiger file from your home directory

Navigating to the directory then catting the file

```
/home/cis90/simben $ cd Poems/; cd Blake; cat tiger; cd
Tiger, Tiger burning bright
In the forest of the night,
What immortal hand or eye
Dare frame thy fearful symmetry?
```

Using a relative pathname

```
/home/cis90/simben $ cat Poems/Blake/tiger
Tiger, Tiger burning bright
In the forest of the night,
What immortal hand or eye
Dare frame thy fearful symmetry?
```

*This is the option I would
choose (fewest keystrokes)*

Using an absolute pathname

```
/home/cis90/simben $ cat /home/cis90/simben/Poems/Blake/tiger
Tiger, Tiger burning bright
In the forest of the night,
What immortal hand or eye
Dare frame thy fearful symmetry?
```

Using ESP method

```
/home/cis90/simben $ cat tiger
cat: tiger: No such file or directory
```



cd command
(your legs)

cd command

change directory

- Syntax: **cd** [*directory*]
- Changes the current working directory to the directory specified.
- Use **cd** with no arguments to return to your home directory.

*Note, users always start in their home directory after logging in.
Every user's home directory is configured in the /etc/passwd file.*

- The *directory* can be:
 - An absolute pathname, e.g. **cd /home/cis90/simben/Poems/Yeats**
 - A relative pathname, e.g. **cd Poems/Yeats**
 - A `..` for the parent of the current working directory, e.g. **cd ..**
- Note, **cd** is a Bash builtin command (part of the shell itself)
`/home/cis90/simben $ type cd`
`cd is a shell builtin`

The .. directory

To move up the tree use: **cd ..**

.. is a hidden file located in every single directory and it is hard linked to the absolute pathname of the parent directory



cd command

change directory example

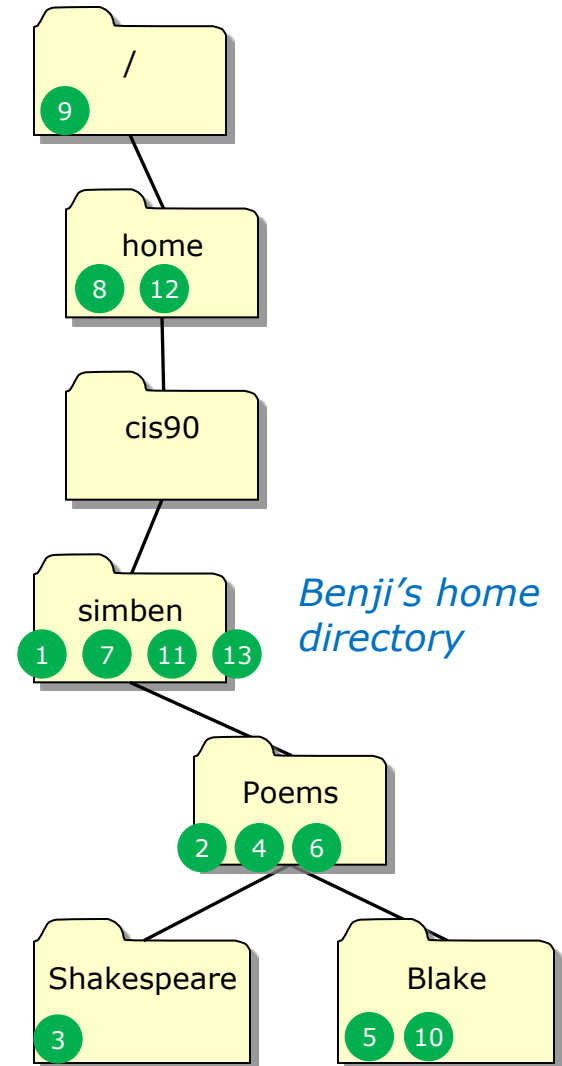
```
/home/cis90/simben $ echo $HOME
```

```
/home/cis90/simben
```

```
/home/cis90/simmsben $ echo $PS1
```

```
$PWD $
```

- 1 /home/cis90/simben \$ **cd Poems/**
- 2 /home/cis90/simben/Poems \$ **cd Shakespeare/**
- 3 /home/cis90/simben/Poems/Shakespeare \$ **cd ..**
- 4 /home/cis90/simben/Poems \$ **cd Blake/**
- 5 /home/cis90/simben/Poems/Blake \$ **cd ..**
- 6 /home/cis90/simben/Poems \$ **cd ..**
- 7 /home/cis90/simben \$ **cd /home**
- 8 /home \$ **cd ..**
- 9 / \$ **cd /home/cis90/simben/Poems/Blake/**
- 10 /home/cis90/simben/Poems/Blake \$ **cd**
- 11 /home/cis90/simben \$ **cd ../..**
- 12 /home \$ **cd**
- 13 /home/cis90/simben \$





pwd command (your GPS)

pwd command

print working directory

- The **pwd** command is your "GPS" to show your current location on the UNIX file tree. Especially with more typical prompts!
- The **pwd** command is equivalent to displaying the value of the PWD environment variable

```
[rsimms@opus net]$ pwd
/lib/modules/2.6.18-164.el5/kernel/drivers/net
```

This is a UNIX command

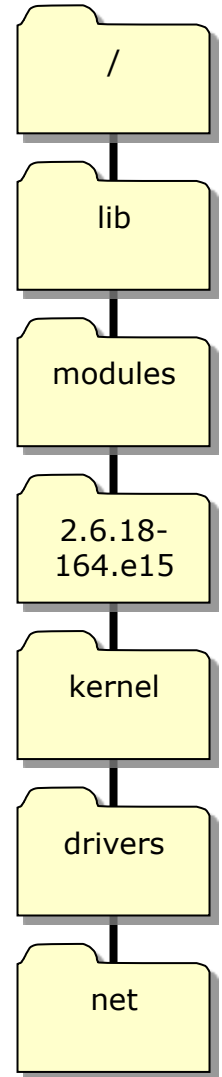
```
[rsimms@opus net]$ echo $PWD
/lib/modules/2.6.18-164.el5/kernel/drivers/net
```

This is a UNIX command

This is shell environment variable (used as an argument to the echo command)

Note: The default shell prompt CIS 90 students utilizes the PWD variable to always show the current working directory.

i.e. When CIS 90 students login this command: PS1='\$PWD \$ ' is automatically done as part of setting up their shell environment.



pwd command

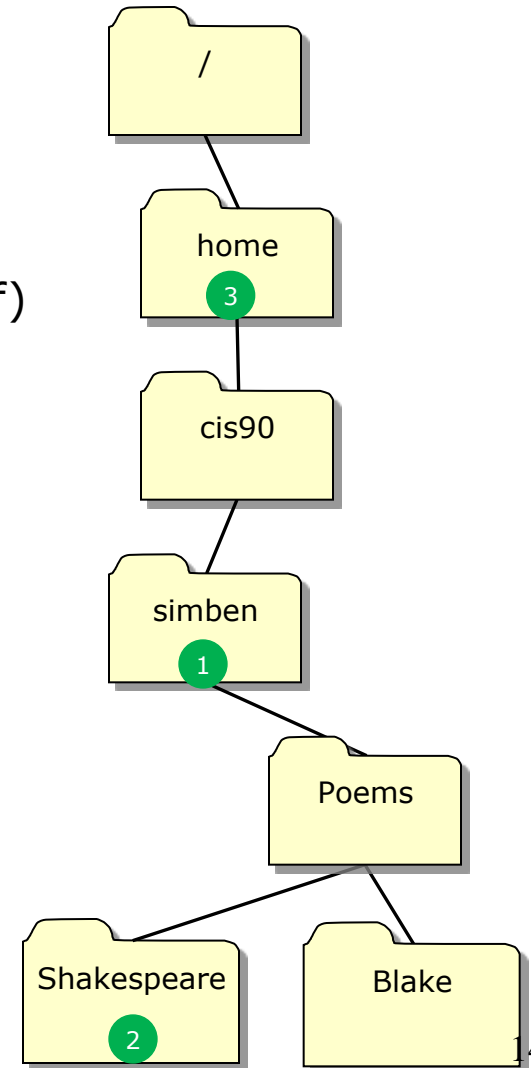
print working directory

Note: The shell prompt has been configured for CIS 90 students to always show the current working directory. This example shows the pwd command with a more typical prompt.

- Syntax: **pwd**
- Prints the current working directory.
- pwd is a BASH builtin command (part of the shell itself)
/home/cis90/simben \$ **type pwd**
pwd is a shell builtin

```

/home/cis90/simben $ PS1='[\u@\h \W]\$ '
1 [simben90@opus ~]$ pwd
/home/cis90/simben
[simben90@opus ~]$ cd Poems/Shakespeare/
2 [simben90@opus Shakespeare]$ pwd
/home/cis90/simben/Poems/Shakespeare
[simben90@opus Shakespeare]$ cd /home/
3 [simben90@opus home]$ pwd
/home
/home/cis90/simben $ PS1='$PWD $ '
/home/cis90/simben $
  
```





ls command
(your eyes)

ls command

Using files vs directories as arguments

```
/home/cis90/simben $ ls
```

```
bigfile  Lab2.0          mission      proposal3   text.fxd
bin      Lab2.1              Poems        small_town  timecal
empty    letter              proposal1    spellk      what_am_i
Hidden   Miscellaneous        proposal2    text.err
```

With no arguments specified, all files in the current directory will be listed

```
/home/cis90/simben $ ls bigfile
```

```
bigfile
```

*With a **filename** specified as an argument, just that file will be listed*

```
/home/cis90/simben $ ls Poems/
```

```
ant  Blake  nursery  Shakespeare  twister  Yeats
```

*With a **directory** specified as an argument, the contents of the directory will be listed*

ls command

specifying multiple directories

*The **ls** command can take multiple arguments*

When a file is specified, just the filename is listed

```
/home/cis90/simben $ ls Poems/ bin/ letter
letter
```

regular file

directories

When a directory is specified, the contents of the directory are listed

```
bin/:
app banner enlightenment hi I treed tryme zoom

Poems/:
ant Blake nursery Shakespeare twister Yeats
```


Is command



- Syntax: **ls [options] [directory]...**

Option	Description
-a	Show all files, even the hidden ones with names starting with "."
-i	Show inode numbers
-d	Show the directory itself rather than the contents of the directory
-l	Long listing (lots of inode information)
-F	Show file types (directory/, program*, link@, socket=)
-S	Sort by size
-t	Sort by date
-R	Recursive (show all sub-directories)

- The *directory* argument can be:
 - An absolute pathname, e.g. **cd /home/cis90/milhom/Poems/**
 - A relative pathname, e.g. **cd Poems**
- If no directory is specified, the current working directory is used.
- More than one directory can be specified
- Use **man ls** to see more information.

ls command

List Files

FYI ...

- **ls** is in /bin and has been aliased to use color on terminal output

```
[simmsben@opus ~]$ type -a ls  
ls is aliased to `ls --color=tty`  
ls is /bin/ls
```

Using the type command to show where a command resides on the path

Note: the `--color=tty` is an option on the **ls** command. Options that are fully spelled usually use two dashes -- instead of 1

We will learn about aliases later in the course

ls command example *with no options*

```

/home/cis90/simmsben $ ls
bigfile  Hidden  letter      Poems      proposal3  text.err  what_am_i
bin      Lab2.0  Miscellaneous proposal1  small_town text.fxd
empty    Lab2.1  mission     proposal2  spellk     timecal
  
```

Regular files in black

Directories in blue

Executables (programs or scripts) in green

*Using the **ls** command with no arguments will list the files in the current directory*

Is command example *with the -F option*

```
/home/cis90/simmsben $ ls -F
bigfile  Hidden/  letter    Poems/    proposal3  text.err  what_am_i
bin/     Lab2.0/  Miscellaneou/  proposal1  small_town  text.fxd
empty    Lab2.1/  mission   proposal2  spellk      timecal*
```

Regular files have no suffix

Directories end with /

*Executables
(programs or scripts)
end with **

*Use the **-F** option to show file types with symbols rather than color (helpful if you are color blind)*

ls command example *with the -a option*



#Gronovis

/home/cis90/simmsben \$ **cd** *cd with no arguments takes you to your home directory*

```
/home/cis90/simmsben $ ls -a
.          .bashrc  Hidden   Miscellaneous  proposal1  text.err
..         bigfile  Lab2.0   mission        proposal2  text.fxd
.bash_history  bin      Lab2.1   .mozilla       proposal3  timecal
.bash_logout  .emacs   .lessht  .plan          small_town what_am_i
.bash_profile empty    letter   Poems          spellk     .zshrc
/home/cis90/simmsben $
```

*Use the -a option to show **hidden files** (files whose names start with a ".")*

.. a hidden file, is the parent directory

. a hidden file, is this the current directory, think of . as meaning "here"

ls command example *with the -S option*



```

/home/cis90/simben $ ls -lS
total 132
-rw-rw-r--. 1 simben90 cis90 21762 Sep 18 15:30 uhistory
-rw-r--r--. 2 simben90 cis90 10576 Jul 20 2001 bigfile
drwxr-xr-x. 2 simben90 cis90 4096 Sep 11 2005 bin
d-----. 2 simben90 cis90 4096 Feb 1 2002 Hidden
drwxr-xr-x. 2 simben90 cis90 4096 Feb 17 2001 Lab2.0
drwxr-xr-x. 3 simben90 cis90 4096 Feb 17 2001 Lab2.1
drwxr-xr-x. 2 simben90 cis90 4096 Sep 11 2005 Miscellaneous
drwxr-xr-x. 5 simben90 cis90 4096 Sep 18 08:49 Poems
-rw-rw-r--. 1 simben90 cis90 4008 Sep 11 22:23 archives
-rw-rw-r--. 1 simben90 cis90 3766 Sep 12 18:53 mbox
-r-----. 1 simben90 staff 2780 Sep 6 13:47 lab01.graded
-rw-r--r--. 1 simben90 cis90 2175 Jul 20 2001 proposal2
-rw-r--r--. 1 simben90 cis90 2054 Sep 14 2003 proposal3
-rw-----. 1 simben90 cis90 1892 Sep 18 15:29 dead.letter
-rw-r--r--. 1 simben90 cis90 1580 Nov 16 2004 small_town
-r-----. 1 simben90 staff 1312 Sep 13 12:27 lab02.graded
-rw-rw-r--. 1 simben90 cis90 1194 Sep 12 15:19 mymessages
-rw-r--r--. 1 simben90 cis90 1074 Aug 26 2003 proposal1
-rw-r--r--. 1 simben90 cis90 1044 Jul 20 2001 letter
-rw-r--r--. 1 simben90 cis90 759 Jun 6 2002 mission
-rwxr-xr-x. 1 simben90 cis90 509 Jun 6 2002 timecal
-rw-r--r--. 1 simben90 cis90 485 Aug 26 2003 spellk
-rw-r--r--. 1 simben90 cis90 352 Jul 20 2001 what_am_i
-rw-r--r--. 1 simben90 cis90 250 Jul 20 2001 text.err
-rw-r--r--. 1 simben90 cis90 231 Jul 20 2001 text.fxd
-rw-r--r--. 1 simben90 cis90 52 Sep 3 10:03 log
-rw-r--r--. 1 simben90 cis90 0 Jul 20 2001 empty
/home/cis90/simben $

```

Note directories all have the same size (4096 bytes)

Use the -S option to sort files by size

ls command example *with the -i option*



/home/cis90/simmsben \$ **cd** *cd with no arguments take you to your home directory*

/home/cis90/simmsben \$ **ls -i**

9171	archives	9351	lab02.graded	12107	mission	12137	spellk
12613	bigfile	12080	Lab2.0	9233	mymessages	12138	text.err
12067	bin	12091	Lab2.1	12109	Poems	12139	text.fxd
9087	dead.letter	9662	letter	12133	proposal1	12140	timecal
12076	empty	14208	log	12134	proposal2	9249	uhistory
12077	Hidden	9142	mbox	12135	proposal3	12141	what_am_i
15725	lab01.graded	12102	Miscellaneous	12136	small_town		

Use the -i option to show the inode associated with a filename

***This command shows exactly what is kept in a directory:
filename & inode pairs (kind of like a phone book)***

ls command with the -lR options

long listing and recursive

```

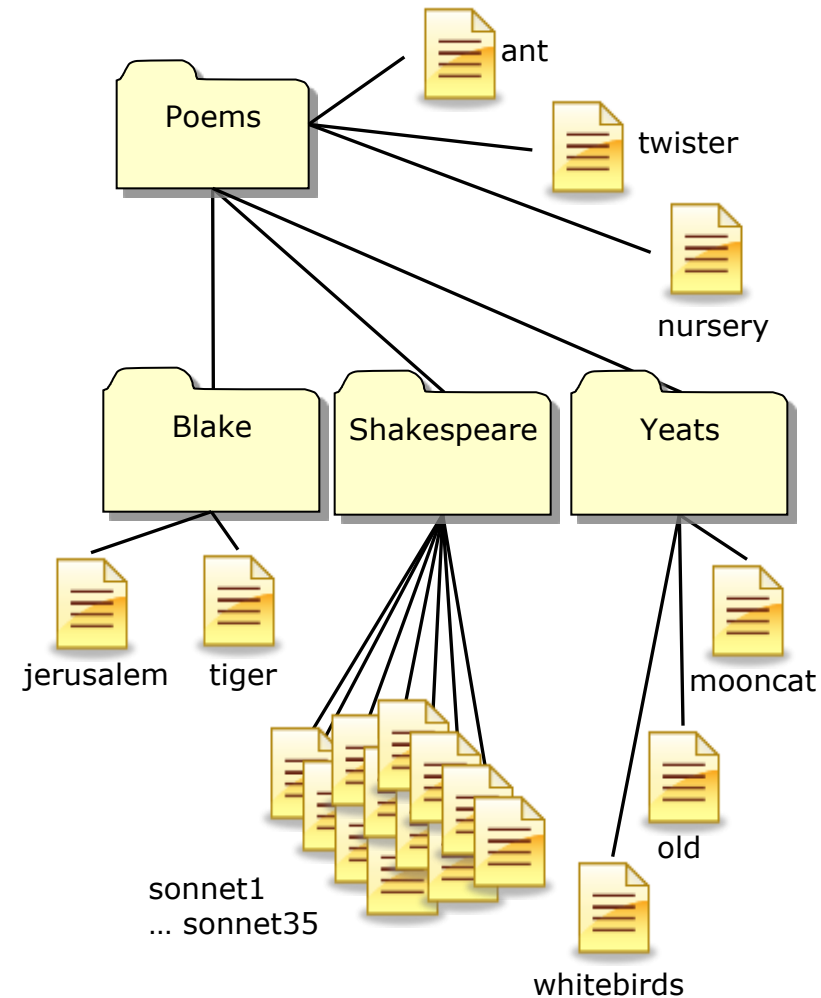
simmsben@opus:~/Poems
[simmsben@opus Poems]$ls -lR
.:
total 48
-rw-r--r-- 1 simmsben cis90 237 Aug 26 2003 ant
drwxr-xr-x 2 simmsben cis90 4096 Jul 20 2001 Blake
-rw-r--r-- 1 simmsben cis90 779 Oct 12 2003 nursery
drwxr-xr-x 2 simmsben cis90 4096 Oct 31 2004 Shakespeare
-rw-r--r-- 1 simmsben cis90 151 Jul 20 2001 twister
drwxr-xr-x 2 simmsben cis90 4096 Jul 20 2001 Yeats

./Blake:
total 16
-rw-r--r-- 1 simmsben cis90 582 Jul 20 2001 jerusalem
-rw-r--r-- 1 simmsben cis90 115 Jul 20 2001 tiger

./Shakespeare:
total 104
-rw-r--r-- 1 simmsben cis90 614 Jul 20 2001 sonnet1
-rw-r--r-- 1 simmsben cis90 620 Jul 20 2001 sonnet10
-rw-r--r-- 1 simmsben cis90 689 Oct 31 2004 sonnet11
-rw-r--r-- 1 simmsben cis90 618 Jul 20 2001 sonnet15
-rw-r--r-- 1 simmsben cis90 647 Jul 20 2001 sonnet17
-rw-r--r-- 1 simmsben cis90 631 Jul 20 2001 sonnet2
-rw-r--r-- 1 simmsben cis90 601 Jul 20 2001 sonnet26
-rw-r--r-- 1 simmsben cis90 615 Jul 20 2001 sonnet3
-rw-r--r-- 1 simmsben cis90 598 Jul 20 2001 sonnet35
-rw-r--r-- 1 simmsben cis90 588 Jul 20 2001 sonnet4
-rw-r--r-- 1 simmsben cis90 622 Jul 20 2001 sonnet5
-rw-r--r-- 1 simmsben cis90 581 Jul 20 2001 sonnet7
-rw-r--r-- 1 simmsben cis90 620 Jul 20 2001 sonnet9

./Yeats:
total 24
-rw-r--r-- 1 simmsben cis90 855 Jul 20 2001 mooncat
-rw-r--r-- 1 simmsben cis90 520 Jul 20 2001 old
-rw-r--r-- 1 simmsben cis90 863 Jul 20 2001 whitebirds
[simmsben@opus Poems]$

```



ls command *with the -d option*



@tahid

```
/home/cis90/simben $ ls bin  
app banner enlightenment hi I treed tryme zoom
```

The contents of the directory are shown

```
/home/cis90/simben $ ls -d bin  
bin
```

The directory itself is shown with the -d option

*Use the **d** option to list the directory itself. Without the **d** the directory contents are listed instead.*

ls command *with the -d option*



```
simben90@opus:~/cis90/simben $ ls -l bin
total 68
-rwxr-xr-x 1 simben90 cis90 220 Apr 22 2004 app
-rwxr-xr-x 1 simben90 cis90 6160 Aug 28 2003 banner
-rwxr-xr-x 1 simben90 cis90 3442 Feb 4 16:36 enlightenment
-rwxr-xr-x 1 simben90 cis90 107 Jul 20 2001 hi
-rwxr-x--x 1 simben90 cis90 375 Oct 20 2003 I
-rwxr-xr-x 1 simben90 cis90 190 Jul 20 2001 treed
-rwxr-xr-x 1 simben90 cis90 174 Mar 4 2004 tryme
-rwxr-xr-x 1 simben90 cis90 74 Jul 20 2001 zoom
~/cis90/simben $
~/cis90/simben $ ls -ld bin
drwxr-xr-x 2 simben90 cis90 4096 Feb 12 16:07 bin
~/cis90/simben $
```

The directory contents are shown

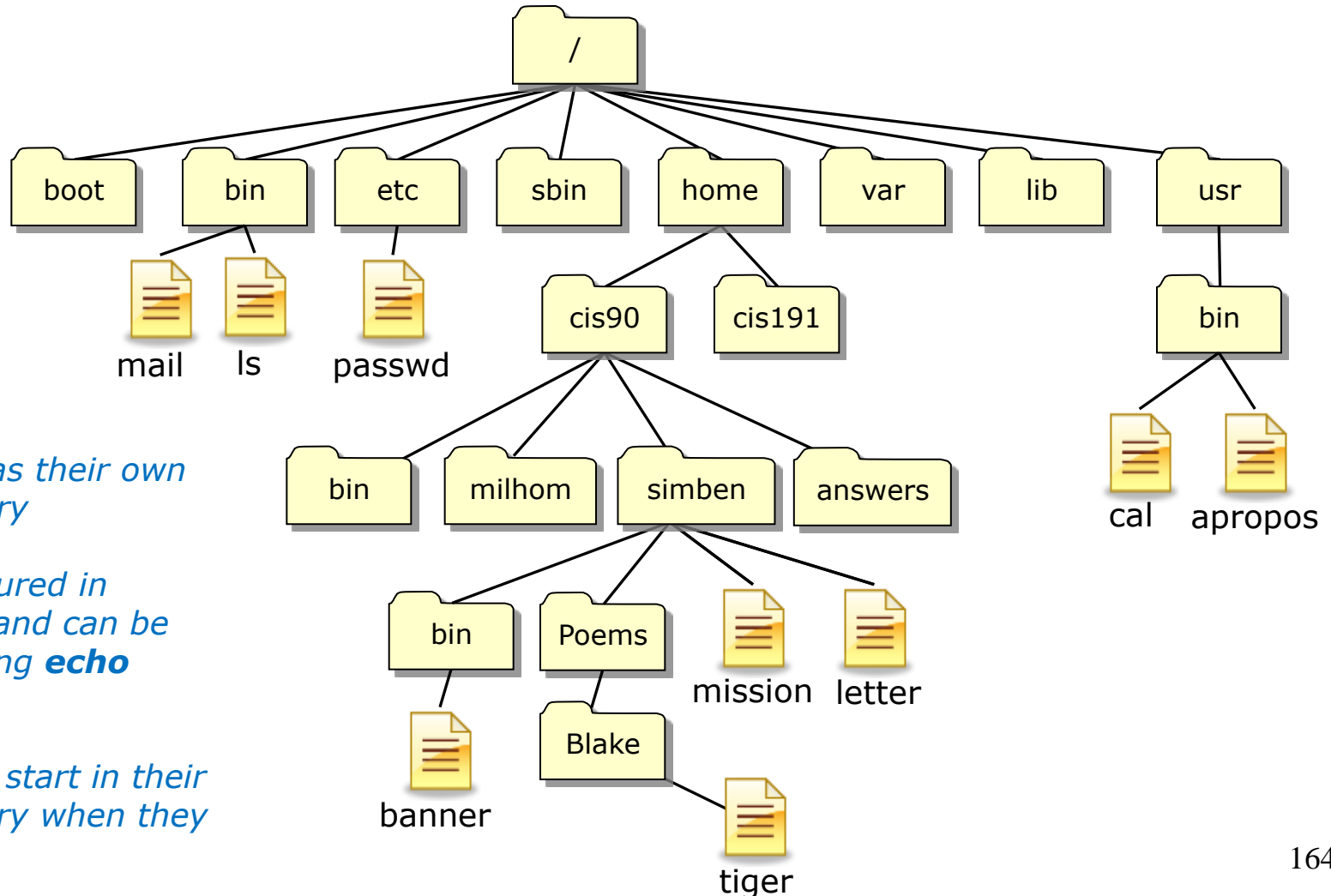
The directory itself is shown with the -d option



Home directories

UNIX File Tree

/ = root of the tree



Every user has their own home directory

This is configured in `/etc/passwd` and can be displayed using **echo \$HOME**

Users always start in their home directory when they login

Class Activity

- 1) Find your entry (use your own logname) in /etc/passwd

```
/home/cis90/simben $ grep simben90 /etc/passwd  
simben90:x:1047:190:Benji Simms:/home/cis90/simben:/bin/bash
```

- 2) Show the contents of the HOME variable

```
/home/cis90/simben $ echo $HOME  
/home/cis90/simben
```

- 3) List the contents of your home directory

```
/home/cis90/simben $ ls /home/cis90/simben  
archives      empty          Lab2.0  Miscellaneous  proposal2  text.err  uhistory.bak  
bigfile       Hidden        Lab2.1  mission        proposal3  text.fxd  what_am_i  
bin           lab01.graded  letter  Poems          small_town timecal  
dead.letter   lab02.graded  log     proposal1     spellk     uhistory
```

Question:

What are some different ways to get the inode number of your home directory?



Question: What are some different ways to get the inode number of your home directory while you are in your home directory?

Answer: At least four ways:

① /home/cis90/simben \$ **ls -id /home/cis90/simben/**
9017 /home/cis90/simben/

Specify the absolute pathname of the home directory

② /home/cis90/simben \$ **ls -id .**
9017 .

Using the . if you are currently in your home directory

③ /home/cis90/simben \$ **ls -id ~**
9017 /home/cis90/simben

The ~ is always an absolute pathname to home directory

④ /home/cis90/simben \$ **ls -i /home/cis90** *Using contents of the parent directory*

13658	answers	12656	depot	9342	keljos	9605	mosmic	9559	specod
9062	beakie	9154	fahmic	9348	lefnic	9460	patcar	9635	thinic
12625	bin	9277	fitcon	9354	lehreb	9484	perste	9573	tilbuz
9074	calmic	9647	genmar	9374	lemrob	9653	ramenr	9579	vasjor
9087	casenr	11282	guest	9389	malmil	9535	ramjua	9629	vivrut
9100	casric	9283	gutemi	9641	matjon	9032	rodduk	9611	weljon
6782	cis	9297	hictre	9131	mccpat	9544	rudtro	9585	weltim
9137	daweli	9312	hormat	9023	milhom	9017	simben		

Note the use of the -d option on ls to focus on the directory itself rather than the directory contents

Filename expansion with *

The "*" metacharacter

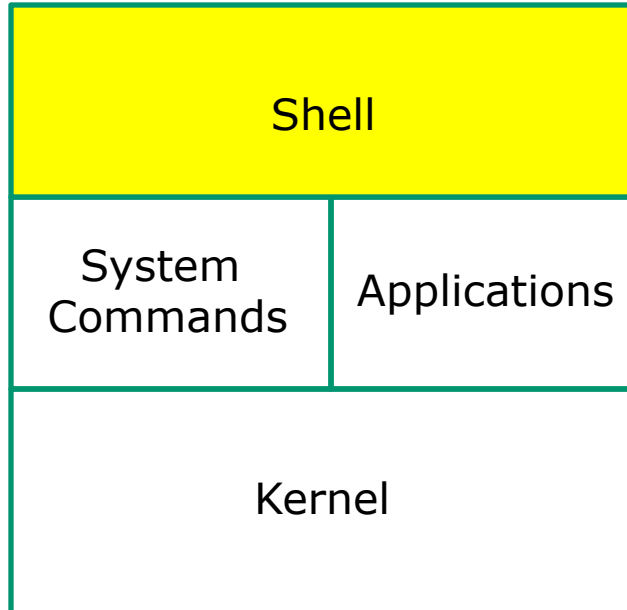


*The * is expanded by the shell and replaced with the names of all files and directories in the current directory*

```
/home/cis90/simben $ file *
archives:      ASCII mail text
bigfile:      ISO-8859 English text, with overstriking
bin:          directory
dead.letter:  ASCII text
empty:        empty
Hidden:       directory
lab01.graded: ASCII English text
lab02.graded: ASCII English text
Lab2.0:       directory
Lab2.1:       directory
letter:       ASCII English text
log:          ASCII text
Miscellaneous: directory
mission:      ASCII English text
Poems:        directory
proposal1:    ASCII English text
proposal2:    ASCII English text
proposal3:    ASCII English text
small_town:  ASCII English text
spellk:      ASCII English text
text.err:    ASCII text
text.fxd:    ASCII text
timecal:     Bourne-Again shell script text executable
uhistory:    ASCII mail text
uhistory.bak: ASCII mail text
what_am_i:   data
```



Life of the Shell



1) Prompt

2) **Parse**

*Metacharacters, like the *, are processed and expanded during the Parse step*

3) Search

(before the selected command is even run)

4) Execute

5) Nap

6) Repeat



*

filename expansion metacharacter

- The * is a shell metacharacter
- During the **parse step** the shell expands * and replaces it with matching filenames in the current directory or as part of any pathnames specified as arguments.
- The commands loaded by the shell never see the *, instead then see the expanded filenames.
- The * will only match non-hidden filenames when used by itself.

*

filename expansion metacharacter

```
/home/cis90/simben/Poems/Yeats $ ls
mooncat  old  whitebirds
```

```
/home/cis90/simben/Poems/Yeats $ file mooncat old whitebirds
mooncat:      ASCII English text
old:          ASCII English text
whitebirds:   ASCII English text
```

*user manually
types in each
filename in
directory*

```
/home/cis90/simben/Poems/Yeats $ file *
mooncat:      ASCII English text
old:          ASCII English text
whitebirds:   ASCII English text
```

*User let's the shell
do the work instead*

*In the second example, the shell, during the parse step, expands the * and replaces it with mooncat old whitebirds.*

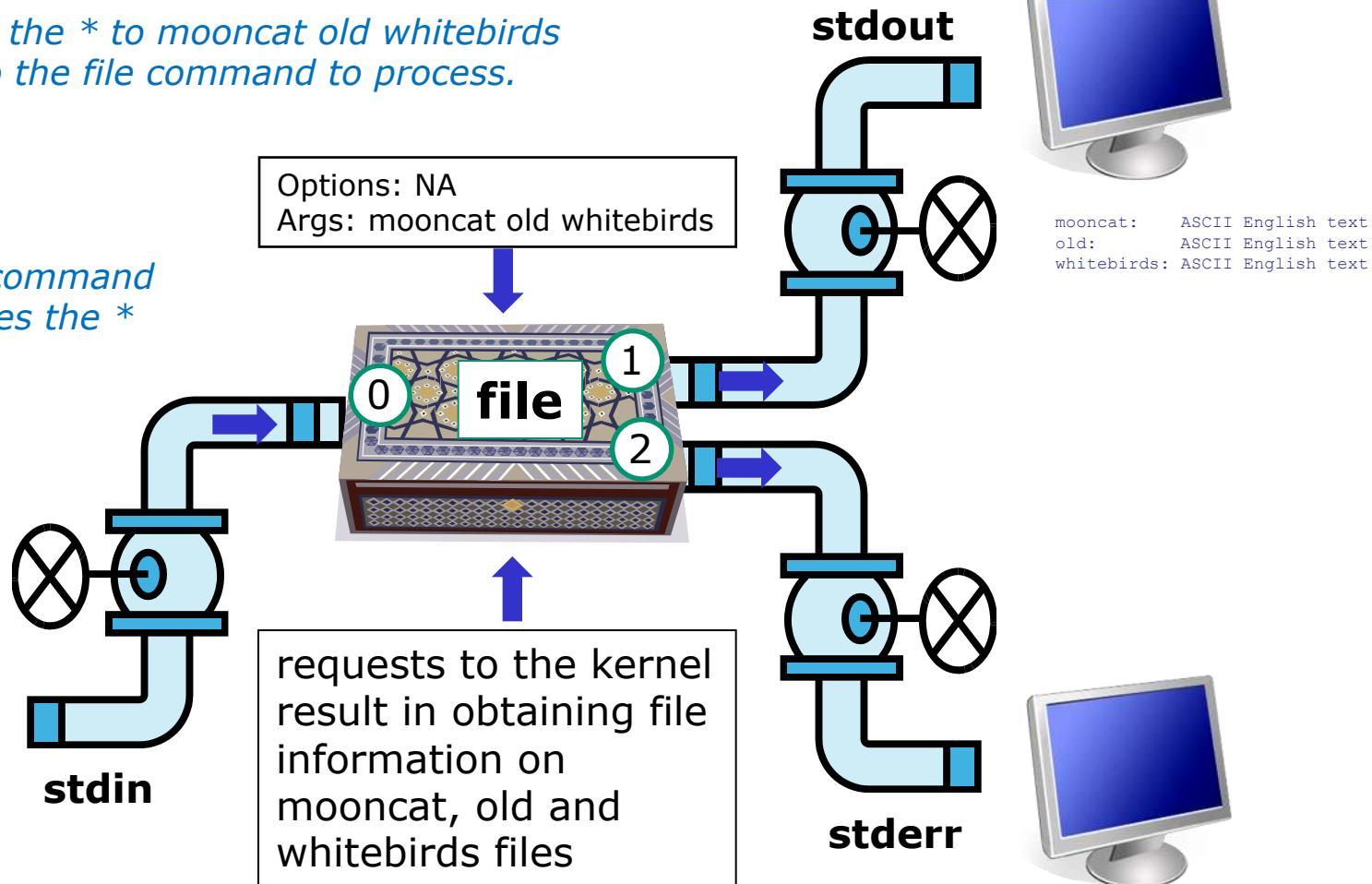
*The **file** command never sees the "*"*

Example program to process: file command

```
/home/cis90/simben/Poems/Yeats $ file *
```

*The shell expands the * to mooncat old whitebirds which is passed to the file command to process.*

*The file command never sees the **



***** metacharacter
used as a *prefix* character

```
/home/cis90/simben $ ls
bigfile  Lab2.0          mission  proposal3  text.fxd
bin      Lab2.1             Poems    small_town  timecal
empty    letter             proposal1 spellk      what_am_i
Hidden   Miscellaneous      proposal2 text.err
```

```
/home/cis90/simben $ ls *.err
text.err
```

***.err** matches all file names **ending** with ".err"

*Shell operation question: Does the **ls** command see the "*" typed by the user?*

***** metacharacter
used as an *infix* character

```
/home/cis90/simben $ ls
bigfile  Lab2.0          mission      proposal3   text.fxd
bin      Lab2.1              Poems       small_town  timecal
empty    letter              proposal1   spellk      what_am_i
Hidden   Miscellaneous        proposal2   text.err
```

```
/home/cis90/simben $ ls *am*
what_am_i
```

am matches all file names **containing** "am"

Answer to the question on previous slide: NO! The shell replaced the ".err" with the string "text.err" and that's what the **ls** command received as an argument.*

***** metacharacter
used as a *postfix* character

```
/home/cis90/simben $ ls
bigfile  Lab2.0          mission      proposal3   text.fxd
bin      Lab2.1              Poems       small_town  timecal
empty    letter              proposal1   spellk      what_am_i
Hidden   Miscellaneous        proposal2   text.err
```

```
/home/cis90/simben $ ls p*
proposal1 proposal2 proposal3
```

p* matches all file names ***starting*** with a "p"

Class Activity

List all poems in the CIS 90 student home directories whose filename contains "cat"

Type the name of these files in the chat window

```
ls ../Poems/*cat*
```



The path to enlightenment

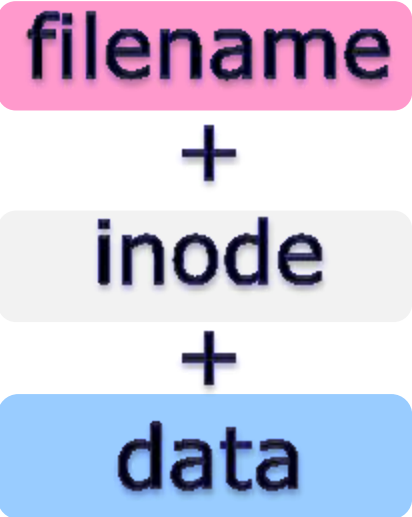
UNIX Files

The three elements of a file

```
/home/cis90/simben/Poems $ ls  
ant Blake nursery Shakespeare twister Yeats
```

```
/home/cis90/simben/Poems $ ls -li twister  
102625 -rw-r--r-- 1 simben90 cis90 151 Jul 20 2001 twister
```

```
/home/cis90/simben/Poems $ cat twister  
A tutor who tooted the flute,  
tried to tutor two tooters to toot.  
Said the two to the tutor,  
"is it harder to toot? Or to  
tutor two tooters to toot?"
```



Class Exercise Enlightenment

- **cd** to your home directory on Opus
- Run the enlightenment program: **enlightenment**
- Write down each magic word as you learn them.

Assignment





Lab 4: The UNIX File System

The goal of this lab is to become proficient with system commands for viewing the directories and different file types that make up a UNIX file system.

Objectives

1. Find and skim Lesson 4 slides: <http://simms-teach.com/cis90calendar.php>
2. Check the forum for news on this lab: <http://cislab.cis.cabrillo.edu/forum/>
3. For additional assistance come to the CIS Lab: <http://webhawks.org/~cislab/>

Prerequisites (not graded)

Log on to the Open server so that you have a terminal line shell at your service. Be sure you are in your home directory to start this lab. You do not need to record or submit your answers for this session.

- o Display a listing of the files in your home directory. Are they in any special order?
- o Use `ls -ls` to display the contents of directory `./bin`.
- o Which filenames do not follow the UNIX file naming conventions?
- o Use `ls -l *` to determine what kind of files you have in your home directory.
- o Do all directories begin with an uppercase letter?
- o Use the `ls -l` command to analyze the following files: `binaries`, `programs`, `binmail`, and `empty`.
- o Display the contents of the `binmail` file on your screen.
- o Display the contents of the `binmail` file on your screen. Is it more appropriate to use `cat` or `more`?
- o List the filenames stored under the root (`/`) directory. Are these files or subdirectories or both?
- o Determine the absolute pathname of your home directory. Use this pathname as an argument to the `ls` command. What are two other ways of getting the same listing?
- o List the contents of your `binmail` directory using a relative pathname.
- o Do the same thing using an absolute pathname, i.e. beginning with a slash (`/`).
- o Display the contents of the `binmail` file stored under the `angelina` directory, which is under the `binmail` directory.
- o Use a single `more` or `less` command to browse all of the files stored under the `binmail` subdirectory. Why did you choose the command you used? How can you tell where one file ends and the next begins?
- o Use the `more` and `less` commands to look at the raw and formatted text files of `binmail`.
- o Use the `ls` command to see what is stored in the `/bin` directory.

Lab 4

If you get stuck, please ask questions on the forum or ask one of the lab assistants in the CIS Lab.



Wrap up

Commands:

cat	Print a file on the screen
cd	Change directory
file	Classify a file
head	View first several lines of a file
less	Scroll up and down long files
ls	List files
more	Scroll down long files
pwd	Print working directory
reset	Use to reset terminal window
tail	View last several lines of a file
wc	Count the words, lines or characters in a file
xxd	Hex dump of a binary file

New Files and Directories:

/	Root of the file tree
/home	Opus home directories
/home/cis90	CIS 90 class home directories
/home/cis90/ <i>username</i>	The home directory for CIS 90 student <i>username</i> (<i>without the 90</i>)
/etc/passwd	



Next Class

Assignment: Check Calendar Page on web site to see what is due next week.

Lab 4

Quiz questions for next class:

- 1) What are two commands you can use to read through long text files?
- 2) How do you distinguish between relative and absolute pathnames?
- 3) What are the three elements of a UNIX file?



Backup

Parsing & Command Syntax

Shell prints
this to prompt
user to enter a
command

Shell parses this command line



Examples

Options modify the
behavior of the command

Arguments are what the
command works upon

Redirection is
covered later in
the course

```

/home/cis90/simben $
/home/cis90/simben $ ls
/home/cis90/simben $ ls -l
/home/cis90/simben $ ls -l -t
/home/cis90/simben $ ls -li Poems/
/home/cis90/simben $ ls -a Poems/ bin/
/home/cis90/simben $ ls -d Poems/ bin/ > mylist
    
```

Spaces (blanks) are used to separate the command,
options and arguments. Additional blanks are ignored.



Lab 2

Lab 2 Results - S2

2. The type command takes another command as an argument and shows whether that command is on the path and if so where it resides. Type each of the following commands and notice where the commands supplied as arguments are located.

```
type man  
type uname  
type tryme  
type echo  
type type  
type bogus  
type man uname type
```

Lab 2 Results - S2

```
/home/cis90/simben $ type man
```

```
man is /usr/bin/man
```

*The **man** command is in the /usr/bin directory*

```
/home/cis90/simben $ type uname
```

```
uname is /bin/uname
```

*The **uname** command is in the /bin directory*

```
/home/cis90/simben $ type tryme
```

```
tryme is /home/cis90/simben/bin/tryme
```

*The **tryme** command is in the bin/ directory of your home directory*

*Use the **type** command to find where on the path a command is located*

Lab 2 Results - S2

```
/home/cis90/simben $ type echo  
echo is a shell builtin
```

```
/home/cis90/simben $ type type  
type is a shell builtin
```

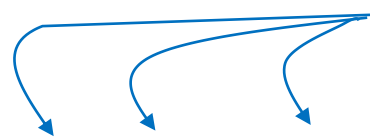
```
/home/cis90/simben $ type bogus  
-bash: type: bogus: not found
```

*The **echo** and **type** commands are built into the bash shell*

There was no command named bogus on the path

Lab 2 Results - S2

```
/home/cis90/simben $ type man uname type  
man is /usr/bin/man  
uname is /bin/uname  
type is a shell builtin
```

 3 arguments

*The type command can
take multiple arguments*

Lab 2 Results - S3

3. Use the echo command to show the value of several shell variables.

```
echo $HOME
```

```
echo $TERM
```

```
echo $LOGNAME
```

```
echo $PS1
```

```
echo $SHELL
```

```
echo $PATH
```

```
echo $TERM $HOME $LOGNAME
```

```
echo $LOGNAME
```

```
echo LOGNAME
```

```
echo $BOGUS
```

```
echo I am $LOGNAME and I like the $SHELL shell
```

Lab 2 Results - S3

```
/home/cis90/simben $ echo $HOME  
/home/cis90/simben
```

The HOME variable contains the absolute pathname of your home directory

```
/home/cis90/simben $ echo $TERM  
xterm
```

The TERM variable contains the type of the terminal you are using

```
/home/cis90/simben $ echo $LOGNAME  
simben90
```

The LOGNAME variable contains the your username

Lab 2 Results - S3

The PS1 variable contains the your primary prompt string definition.

```
/home/cis90/simben $ echo $PS1  
$PWD $
```

The SHELL variable contains the name of the shell being used.

```
/home/cis90/simben $ echo $SHELL  
/bin/bash
```

The PATH variable contains the directories the shell will search for commands you wish to run.

```
/home/cis90/simben $ echo $PATH  
/usr/lib/qt-  
3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin  
:/sbin:/home/cis90/simben/../bin:/home/cis90/simben/bin:.
```

Lab 2 Results - S3

```
/home/cis90/simben $ echo $TERM $HOME $LOGNAME  
xterm /home/cis90/simben simben90
```

You can specify multiple variables at a time (as multiple arguments) on the echo command

```
/home/cis90/simben $ echo $LOGNAME  
simben90
```

A "\$" in front of a variable name instructs the shell to use the value rather than the name of the variable

```
/home/cis90/simben $ echo LOGNAME  
LOGNAME
```

```
/home/cis90/simben $ echo $BOGUS
```

Undefined variables have a null value. "Null" means no value.

```
/home/cis90/simben $
```

echo prints a blank line without any arguments

Lab 2 Results - S3

```
/home/cis90/simben $ echo I am $LOGNAME and I like the $SHELL shell  
I am simben90 and I like the /bin/bash shell
```

This is an example of the echo command taking both text and variables as arguments.

Notice how the shell uses the value rather than the name of a variable when a \$ metacharacter is used.

Lab 2 Results - S7

7. How many arguments do each of the following command lines have?

echo one two threefour


echo "My TERM type is " \$TERM

echo one.two.three

Lab 2 Results - S7

```
/home/cis90/simben $ echo one two threefour  
one two threefour  
(3 arguments)
```

Notice how the shell ignores additional unquoted blanks



```
/home/cis90/simben $ echo "My TERM type is " $TERM  
My TERM type is xterm  
(2 arguments)
```

```
/home/cis90/simben $ echo one.two.three  
one.two.three  
(1 argument)
```

Lab 2 Results - S8

8. What is the difference in output between the following two commands? Note, the \$ and > are part of the prompt, you don't need to type them.


```
$ echo red 'white  
> and blue'
```

and

```
$ echo red white \  
> and blue
```


Note: the [enter] key is pressed immediately after the last character of each line

Lab 2 Results - S8

```
/home/cis90/simben $ echo red 'white'   
> and blue'  
red white  
and blue
```

*The unclosed single quote prevents the **<newline>** from signaling the end of the command.*

*The **<newline>** gets passed to the echo command which outputs two lines.*

```
/home/cis90/simben $ echo red white \  
> and blue  
red white and blue
```

*The **<newline>** is escaped in this example. The shell ignores it and continues to prompt the user for the rest of the command.*

*The escaped **<newline>** is NOT passed to the echo command which outputs only a single line.*

*Pressing the Enter (or Return on Macs) key generates an invisible **<newline>** metacharacter.*

*The **<newline>** signals the shell to stop prompting and process the command line.*

Lab 2 Results - S8

Note: Primary prompt is determined by the value of PS1

```
/home/cis90/simben $ echo $PS1  
$PWD $
```

The value of the PWD environment variable is your current working directory

```
/home/cis90/simben $ echo red 'white' 
```

```
> and blue'
```

```
red white  
and blue
```

Note: Secondary prompt is determined by the value of PS2

```
/home/cis90/simben $ echo $PS2  
>
```


Lab 2 Results - S9

9. Use the shell metacharacter ";" to write out a one line command that will clear the screen, print out the date and the current month's calendar.

\$ _____

Lab 2 Results - S9

```
/home/cis90/simben $ clear; date; cal
```



```
simben90@opus:~  
Sun Feb 26 18:13:42 PST 2012  
February 2012  
Su Mo Tu We Th Fr Sa  
          1  2  3  4  
 5  6  7  8  9 10 11  
12 13 14 15 16 17 18  
19 20 21 22 23 24 25  
26 27 28 29  
  
/home/cis90/simben $
```

The ; metacharacter allows multiple commands on one line

Lab 2 Results - S11

11. Use a single `uname` command with the necessary options to display ONLY the network node hostname, the kernel release number and the operating system. Your command should produce the following output:

```
oslab.cishawks.net 2.6.32-220.23.1.el6.i686 GNU/Linux
```

Hint: Use the `man uname` command, use `q` to quit.

Lab 2 Results - S11

Output from **man uname**

-s, --kernel-name
print the kernel name

-n, --nodename
print the network node hostname

-r, --kernel-release
print the kernel release

-v, --kernel-version
print the kernel version

-m, --mac
pr

-p, --pro
pr

-i, --har
pr

-o, --operating-system
print the operating system

```

/home/cis90/simben $ uname -orn
oslab.cabrillo.edu 2.6.32-220.23.1.el6.i686 GNU/Linux
or
/home/cis90/simben $ uname -o -r -n
oslab.cabrillo.edu 2.6.32-220.23.1.el6.i686 GNU/Linux
    
```

Use the man page to determine the necessary options for the:
operating system, kernel release numbers and network node hostname

Use up and down arrows to scroll man page

Use q to quit the man page

FYI - a tangent on the GNU Public License (GPL)



Under the GPL, Free = Freedom to view and modify source code



Richard Stallman started the GNU project in 1983 to create a free UNIX-like OS. He Founded the Free Software Foundation in 1985. In 1989 he wrote the first version of the GNU General Public License

```
/home/cis90/simben $ uname -orn
opus.cabrillo.edu 2.6.18-164.el5 GNU/Linux
```

└──────────┘
└──────────┘
└──────────┘
node hostname
kernel release
OS

*Dan M. didn't like the order the **uname** command printed the information so he downloaded the source code, modified it, recompiled it. He now has his own version of the **uname** command!*

```
cis90@eko-04:~/dan/coreutils-7.4/src$ ./uname -orn
GNU/Linux 2.6.32-27-generic eko-04
```

└──┘
└──────────┘
└──────────┘
OS
kernel release
node hostname

This is one of the really cool things about Linux and the GNU General Public License ... if you don't like something you can change it!

See: <http://oslab.cabrillo.edu/forum/viewtopic.php?f=31&t=683&p=2632>

Lab 2 Results - S16

16. What is the **whatis** command? Use the command with the argument, bc

How does this compare to using the man command with -f option?

man -f bc

Lab 2 Results - S16

Use the **whatis** or **man** command to determine what the **whatis** command does.

```
/home/cis90/simben $ whatis whatis
```

```
whatis (1) - search the whatis database for complete words
```

```
/home/cis90/simben $ man whatis
```

Output from **man whatis**

```
simmsben@opus:~
whatis(1) whatis(1)
NAME
    whatis - search the whatis database for complete words.
SYNOPSIS
    whatis keyword ...
DESCRIPTION
    whatis searches a set of database files containing short descriptions
    of system commands for keywords and displays the result on the
    standard output. Only complete word matches are displayed.

    The whatis database is created using the command /usr/sbin/make-
    whatis.
AUTHOR
    John W. Eaton was the original author of man. Zeyd M. Ben-Halim
    released man 1.2, and Andries Brouwer followed up with versions 1.3
    thru 1.5p. Federico Lucifredi <flucifredi@acm.org> is the current
```

Lab 2 Results - S16

Use the **whatis** to find out about the **BC** command

```
/home/cis90/simben $ whatis bc  
bc                (1)  - An arbitrary precision calculator language  
bc                (1p) - arbitrary-precision arithmetic language  
bc                (rpm) - GNU's bc (a numeric processing language)  
and dc (a calculator).
```

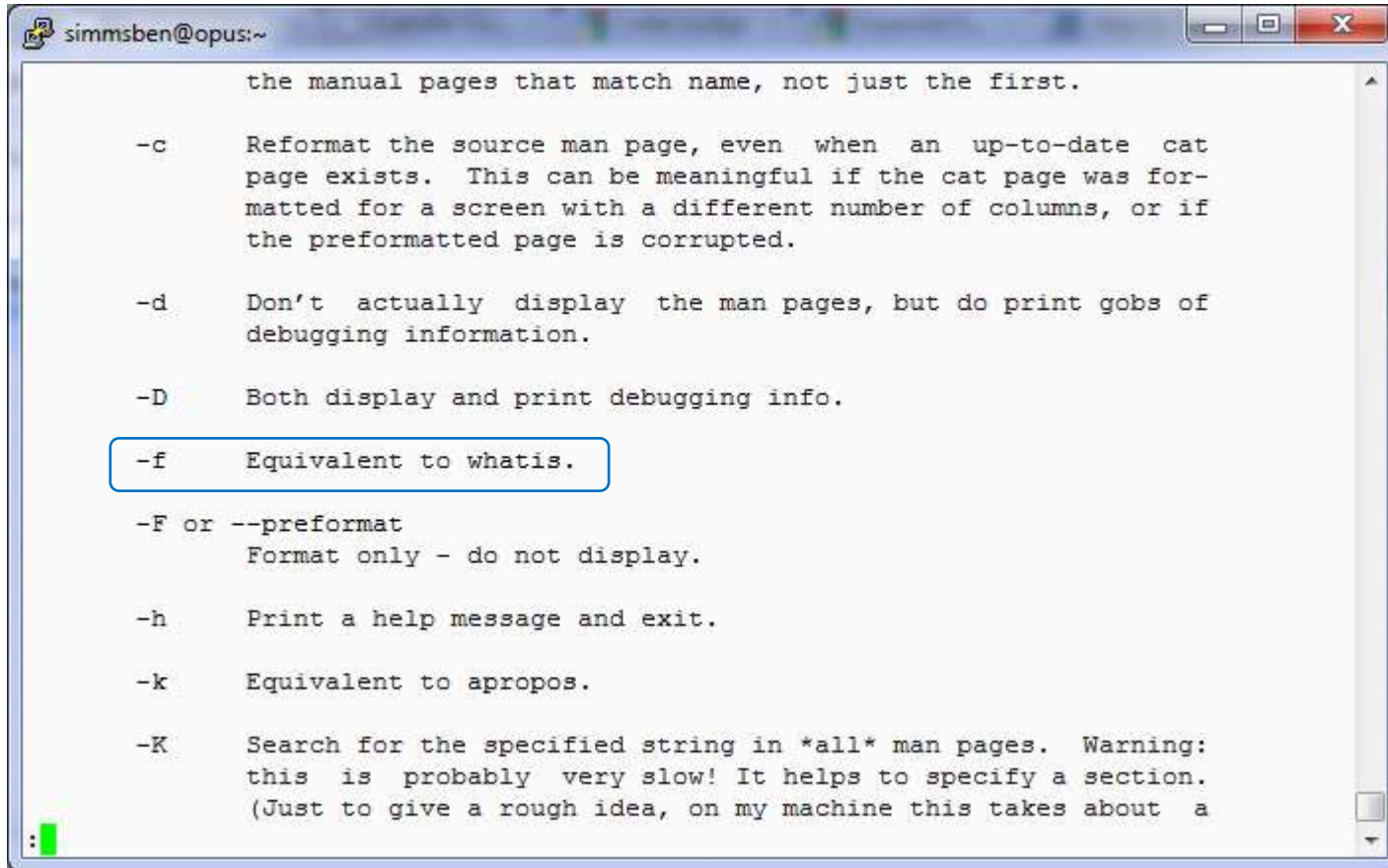
Compare output with **man -f** command

```
/home/cis90/simben $ man -f bc  
bc                (1)  - An arbitrary precision calculator language  
bc                (1p) - arbitrary-precision arithmetic language  
bc                (rpm) - GNU's bc (a numeric processing language)  
and dc (a calculator).  
/home/cis90/simben $
```

They are equivalent

Lab 2 Results - S16

Output from **man man**



```
simmsben@opus:~  
the manual pages that match name, not just the first.  
  
-c Reformat the source man page, even when an up-to-date cat  
page exists. This can be meaningful if the cat page was for-  
matted for a screen with a different number of columns, or if  
the preformatted page is corrupted.  
  
-d Don't actually display the man pages, but do print gobs of  
debugging information.  
  
-D Both display and print debugging info.  
  
-f Equivalent to whatis.  
  
-F or --preformat  
Format only - do not display.  
  
-h Print a help message and exit.  
  
-k Equivalent to apropos.  
  
-K Search for the specified string in *all* man pages. Warning:  
this is probably very slow! It helps to specify a section.  
(Just to give a rough idea, on my machine this takes about a
```

***man man** will display the manual page for the man command and its documented there that the -f option is "Equivalent to whatis"*

Lab 2 Results - S17

17. Is tryme a UNIX command? How do you know?

Lab 2 Results - S17

```
/home/cis90/simben $ tryme
```

```
My name is "tryme"
```

```
I am pleased to make your acquaintance, Benji Simms
```

```
/tmp
```

```
/home/cis90/simben $ whatis tryme
```

```
tryme: nothing appropriate
```

```
/home/cis90/simben $ man tryme
```

```
No manual entry for tryme
```

*UNIX commands are documented with man pages and have entries in the whatis database. **tryme** does not appear in either one so is not a UNIX command*

Lab 2 Results - S17

```
/home/cis90/simben $ type tryme  
tryme is /home/cis90/simben/bin/tryme
```

***type** shows **tryme** resides in the bin/ directory of Benji's home directory*

```
/home/cis90/simben $ file /home/cis90/simben/bin/tryme  
/home/cis90/simben/bin/tryme: Bourne-Again shell script text executable
```

***file** shows **tryme** is a bash shell script*

```
/home/cis90/simben $ cat /home/cis90/simben/bin/tryme  
#!/bin/bash
```

***cat** shows the actual **tryme** script itself*

```
hello()  
{  
    cd /tmp  
}  
PATH=/bin  
echo My name is \"`basename $0`\"  
IFS=:  
set `grep $LOGNAME /etc/passwd`  
echo I am pleased to make your acquaintance, $5  
hello  
pwd
```

Lab 2 Results - S18

18. Use the manual pages, and the **who** command, to find out the number of users logged on.

Lab 2 Results - S18

Output from **man who**

```

--lookup
    attempt to canonicalize hostnames via DNS

-m      only hostname and user associated with stdin

-p, --process
    print active processes spawned by init

-q, --count
    all login names and number of users logged on

-r, --runlevel
    print current runlevel

```

*The man page for **who** shows the **q** option will count the users logged in. Use up and down arrows to scroll.*

```

[rsimms@opus ~]$ who -q
helrog90 jimme190 rsimms saljac193 vascar193
# users=5

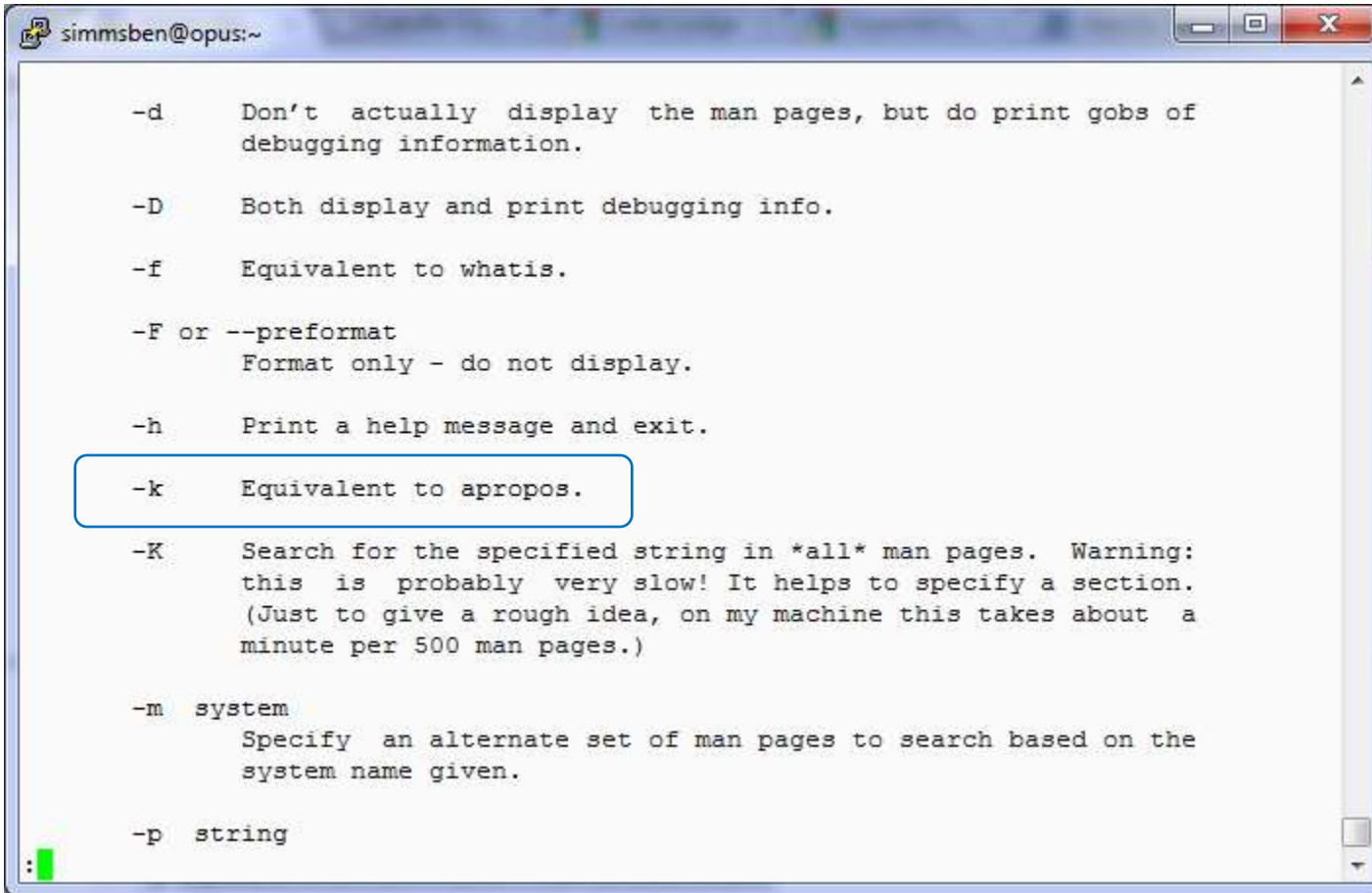
```

Lab 2 Results - S19

19. Run the command: **man -k boot** Use the manual pages to find out what the -k option does. What command is **man -k** equivalent to? Run the equivalent command and verify.

Lab 2 Results - S19

Output from **man man**



```
simmsben@opus:~  
  
-d      Don't actually display the man pages, but do print gobs of  
        debugging information.  
  
-D      Both display and print debugging info.  
  
-f      Equivalent to whatis.  
  
-F or --preformat  
        Format only - do not display.  
  
-h      Print a help message and exit.  
  
-k      Equivalent to apropos.  
  
-K      Search for the specified string in *all* man pages.  Warning:  
        this is probably very slow!  It helps to specify a section.  
        (Just to give a rough idea, on my machine this takes about a  
        minute per 500 man pages.)  
  
-m system  
        Specify an alternate set of man pages to search based on the  
        system name given.  
  
-p string
```

*Use **man**
man to read
the manual
page for the
man
command*

*the **apropos** command is equivalent to the **man -k** command*

Lab 2 Results - S19

Output from **apropos boot**

```

simmsben@opus:~
/home/cis9001/simmsben $ apropos boot
ExtUtils::Mkbootstrap (3pm) - make a bootstrap file for use by DynaLoader
boot-scripts [boot] (7) - General description of boot sequence
bootparam (7) - Introduction to boot time parameters of the Linux kernel
firstboot (rpm) - Initial system configuration utility
firstboot-tui (rpm) - A text interface for firstboot
grub (rpm) - GRUB - the Grand Unified Boot Loader.
initrd (4) - boot loader initialized RAM disk
kexec (8) - directly boot into a new kernel
mbchk (1) - check the format of a Multiboot kernel
mkbootdisk (8) - creates a stand-alone boot floppy for the running system
mkbootdisk (rpm) - Creates a boot floppy disk for booting a system.
perlboot (1) - Beginner(aqs Object-Oriented Tutorial
pxeboot (8) - Network Booting Operating Systems Configuration Utility
pxeos (8) - PXEBoot Operating System description Configuration Utili
ty
reboot (2) - reboot or enable/disable Ctrl-Alt-Del
reboot [halt] (8) - stop the system
rhgb (rpm) - Red Hat Graphical Boot
sys-unconfig (8) - shell script to reconfigure the system upon next boot
syslinux (rpm) - Simple kernel loader which boots from a FAT filesystem
system-config-netboot (8) - Network Booting Configuration Utility
system-config-netboot (rpm) - network booting/install configuration utility (GUI)
system-config-netboot-cmd (rpm) - network booting/install configuration utility
/home/cis9001/simmsben $

```

Output from **man -k boot**

```

simmsben@opus:~
/home/cis9001/simmsben $ man -k boot
ExtUtils::Mkbootstrap (3pm) - make a bootstrap file for use by DynaLoader
boot-scripts [boot] (7) - General description of boot sequence
bootparam (7) - Introduction to boot time parameters of the Linux kernel
firstboot (rpm) - Initial system configuration utility
firstboot-tui (rpm) - A text interface for firstboot
grub (rpm) - GRUB - the Grand Unified Boot Loader.
initrd (4) - boot loader initialized RAM disk
kexec (8) - directly boot into a new kernel
mbchk (1) - check the format of a Multiboot kernel
mkbootdisk (8) - creates a stand-alone boot floppy for the running system
mkbootdisk (rpm) - Creates a boot floppy disk for booting a system.
perlboot (1) - Beginner(aqs Object-Oriented Tutorial
pxeboot (8) - Network Booting Operating Systems Configuration Utility
pxeos (8) - PXEBoot Operating System description Configuration Utili
ty
reboot (2) - reboot or enable/disable Ctrl-Alt-Del
reboot [halt] (8) - stop the system
rhgb (rpm) - Red Hat Graphical Boot
sys-unconfig (8) - shell script to reconfigure the system upon next boot
syslinux (rpm) - Simple kernel loader which boots from a FAT filesystem
system-config-netboot (8) - Network Booting Configuration Utility
system-config-netboot (rpm) - network booting/install configuration utility (GUI)
system-config-netboot-cmd (rpm) - network booting/install configuration utility
/home/cis9001/simmsben $

```

*the **apropos** command is equivalent to the **man -k** command*

Lab 2 Results - Q1 Name a UNIX command that gets its input only from the command line?

Lab 2 Results - Q1 Name a UNIX command that gets its input only from the command line?

```
/home/cis90/simmen $ echo hello world  
hello world
```

```
/home/cis90/simben $ banner hello world  
# # ##### # # #####  
# # # # # # #  
# # # # # # #  
##### ##### # # # #  
# # # # # # # # # #  
# # # # # # # # # #  
# # ##### # # # #  
# # # # # # # # # #  
# # # # # # # # # #  
## ## ##### # # #####
```

The **echo** and **banner** commands are examples of commands that get their input from the command line



Lab 2 Results - Q2 Name an interactive command that reads its input from the keyboard?

Lab 2 Results - Q2 Name an interactive command that reads its input from the keyboard?

```
/home/cis90/simmsben $ bc
bc 1.06
Copyright 1991-1994, 1997, 1998, 2000 Free
Software Foundation, Inc.
This is free software with ABSOLUTELY NO
WARRANTY.
For details type `warranty'.
2+2
4
500-200+3
303
sqrt(64)
8
quit
```

```
/home/cis90/simmsben $ passwd
Changing password for user simmsben.
Changing password for simmsben
(current) UNIX password:
New UNIX password:
BAD PASSWORD: is too similar to the old
one
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated
successfully.
```

*The **bc** (binary calculator) and **passwd** commands are examples of interactive commands that read their input from the keyboard*

Lab 2 Results - Q3 Name a UNIX command that gets its input from the Operating System?

Lab 2 Results - Q3 Name a UNIX command that gets its input from the Operating System?

```
/home/cis90/simmen $ who
dycktim pts/1      2010-09-07 17:07 (nosmo-nat.cabrillo.edu)
root    :0          2009-12-18 17:30
velasoli pts/2      2010-09-07 17:08 (adsl-35-201-114-102.dsl.net)
guest90 pts/3      2010-09-07 16:56 (nosmo-nat.cabrillo.edu)
rsimms  pts/4      2010-09-07 15:54 (dsl-45-78-13-81.dhcp.com)
guest90 pts/5      2010-09-07 16:59 (nosmo-nat.cabrillo.edu)
watsohar pts/6      2010-09-07 17:03 (nosmo-nat.cabrillo.edu)
swansgre pts/7      2010-09-07 17:10 (nosmo-nat.cabrillo.edu)
guest90 pts/8      2010-09-07 17:10 (nosmo-nat.cabrillo.edu)
abbenste pts/9      2010-09-07 17:11 (nosmo-nat.cabrillo.edu)
```

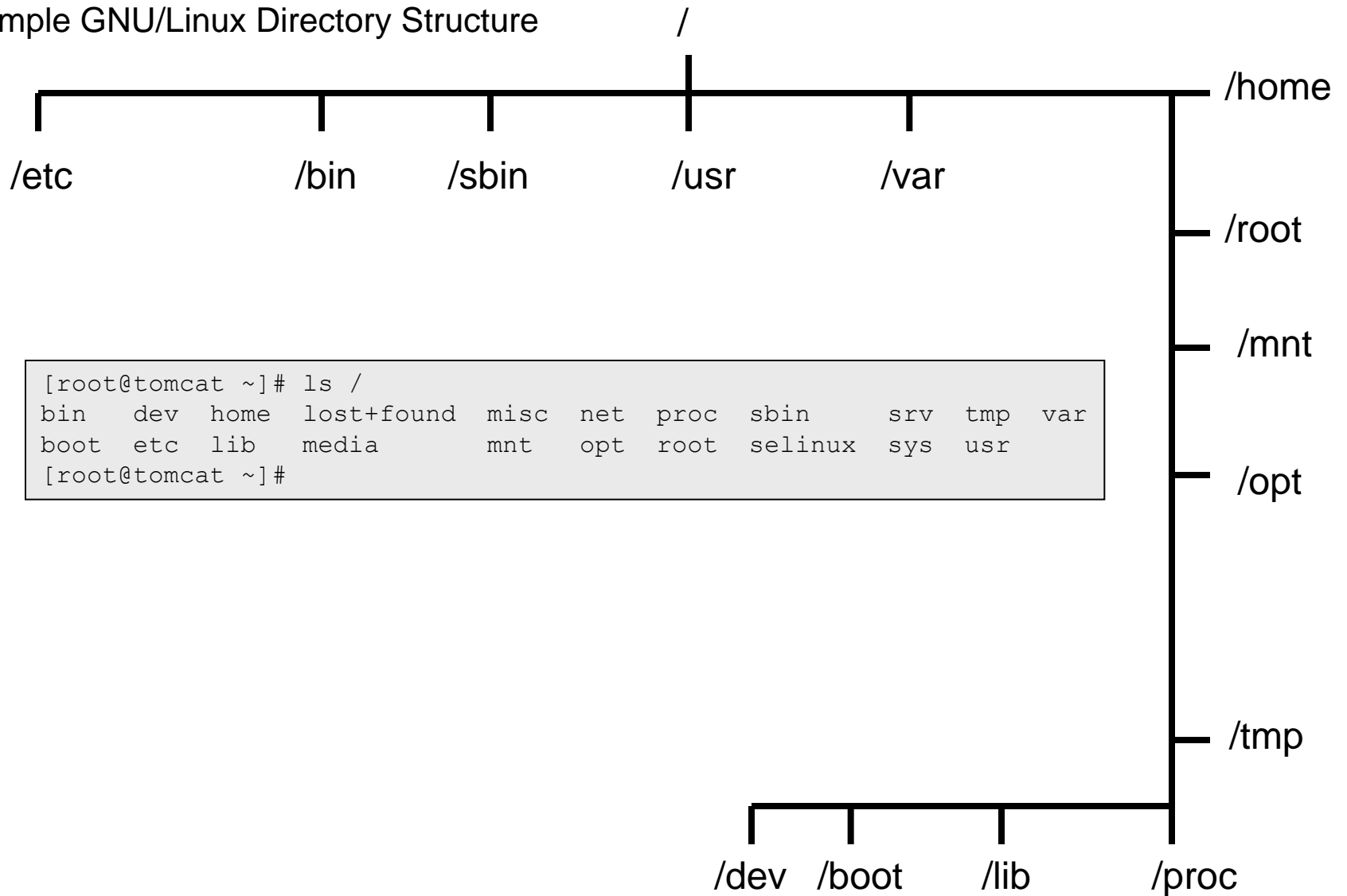
```
/home/cis90/simben $ uname
Linux
```

*The **who** and **uname** commands are examples of commands that get their input from the Operating System*



CIS Directories and Files

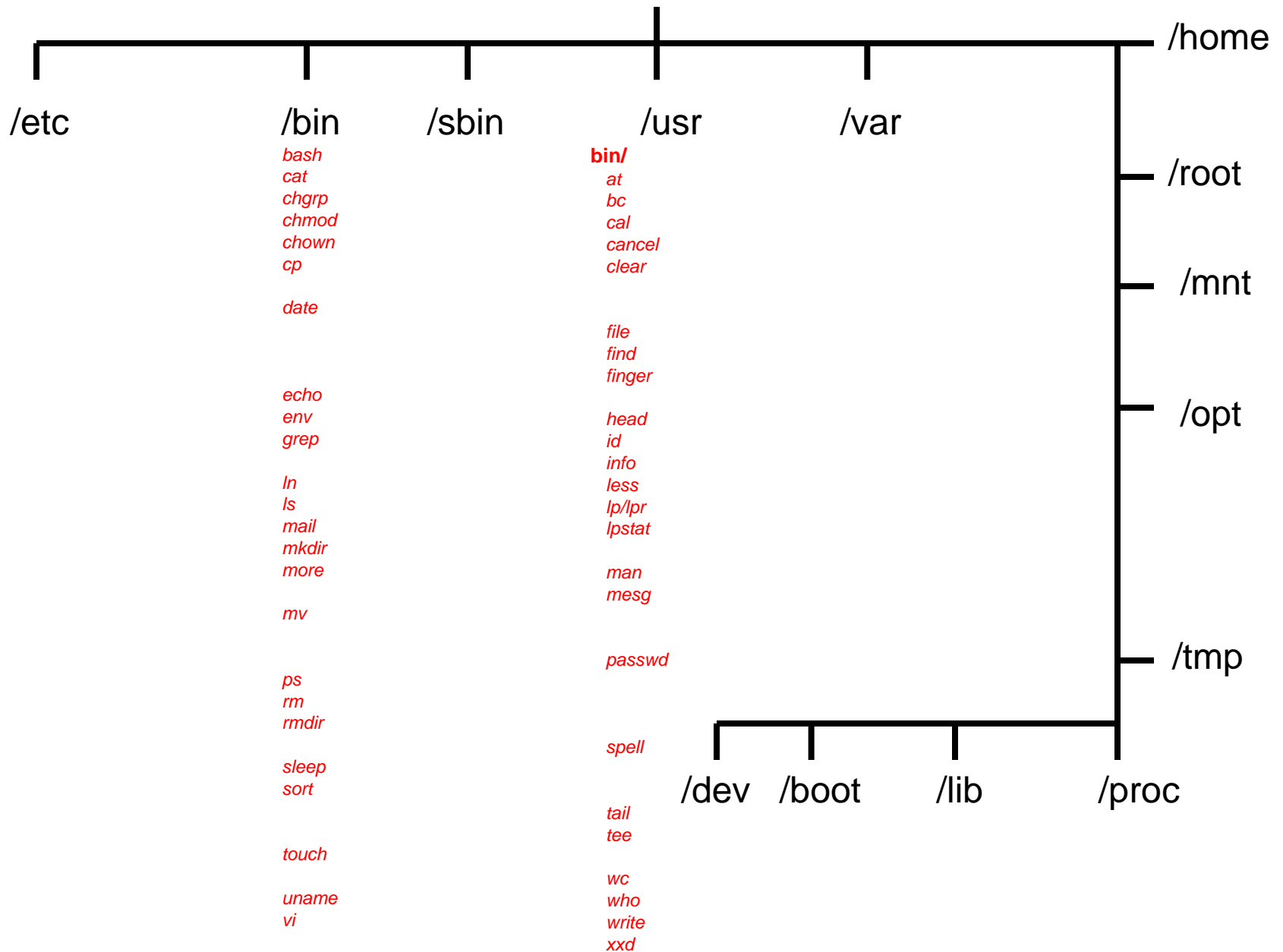
Example GNU/Linux Directory Structure



```
[root@tomcat ~]# ls /
bin  dev  home  lost+found  misc  net  proc  sbin  srv  tmp  var
boot etc  lib   media      mnt   opt  root  selinux  sys  usr
[root@tomcat ~]#
```

Example GNU/Linux Directory Structure

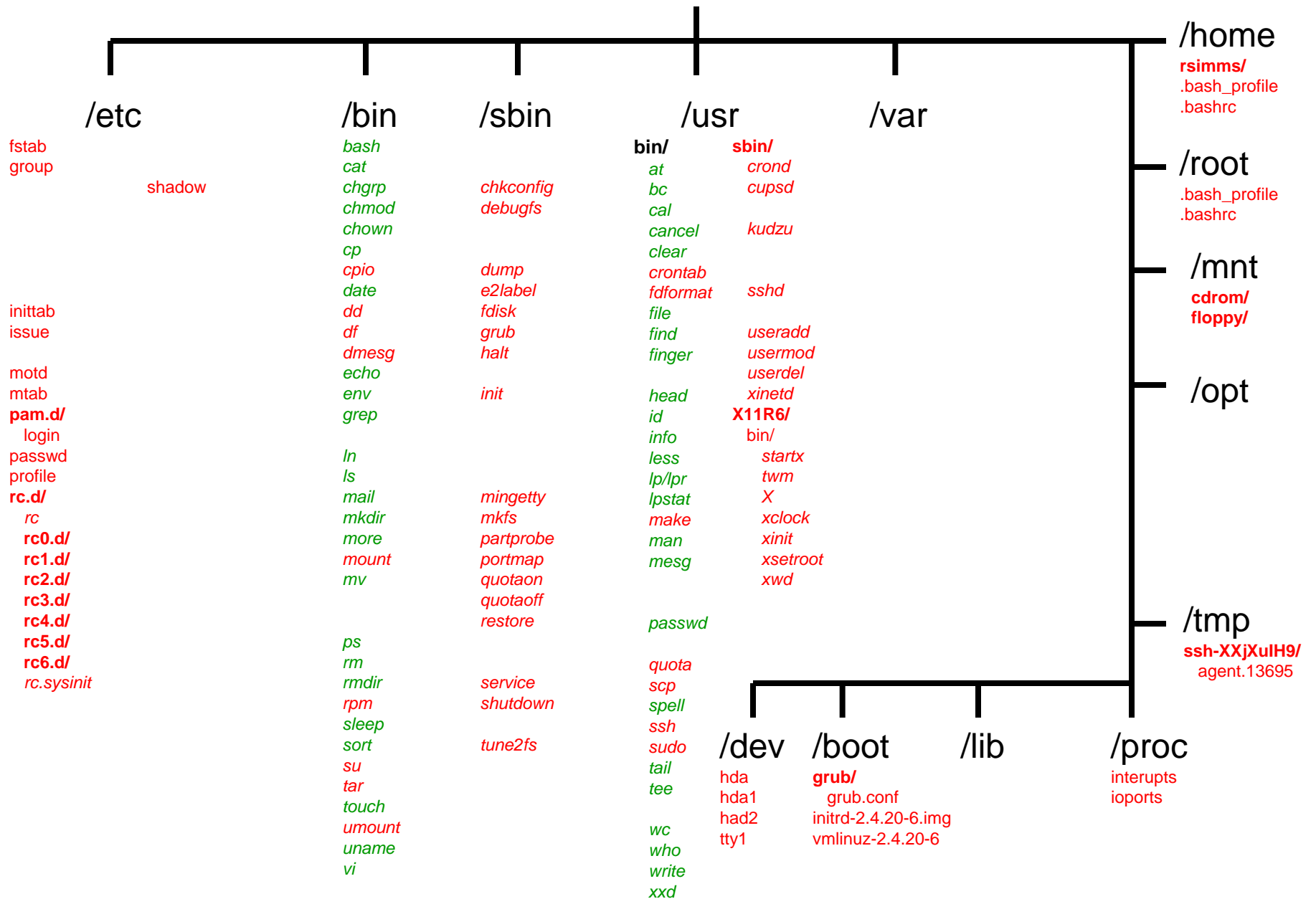
CIS 90 files, directories, commands



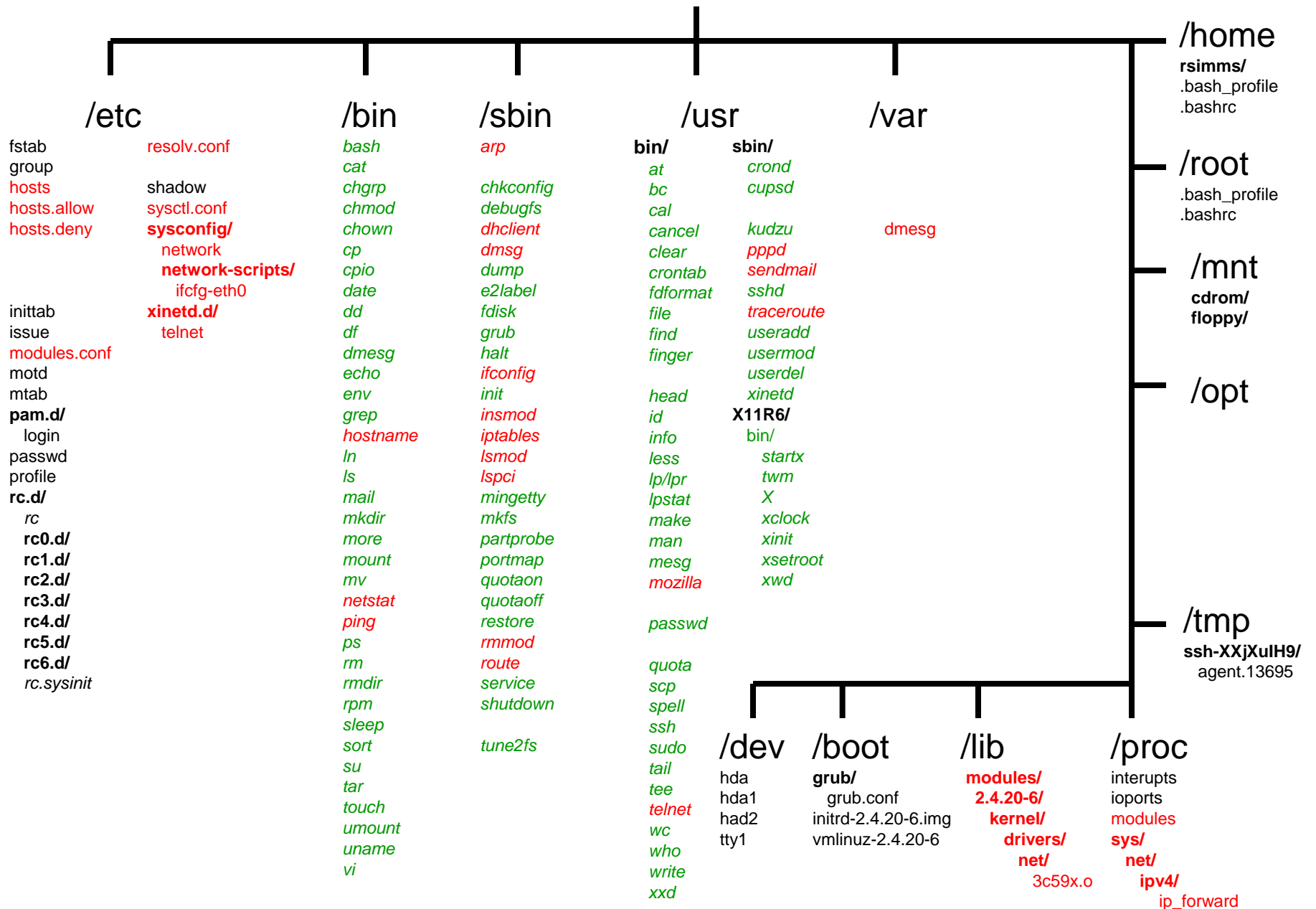
Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset

Example GNU/Linux Directory Structure

CIS 191 files, directories, commands



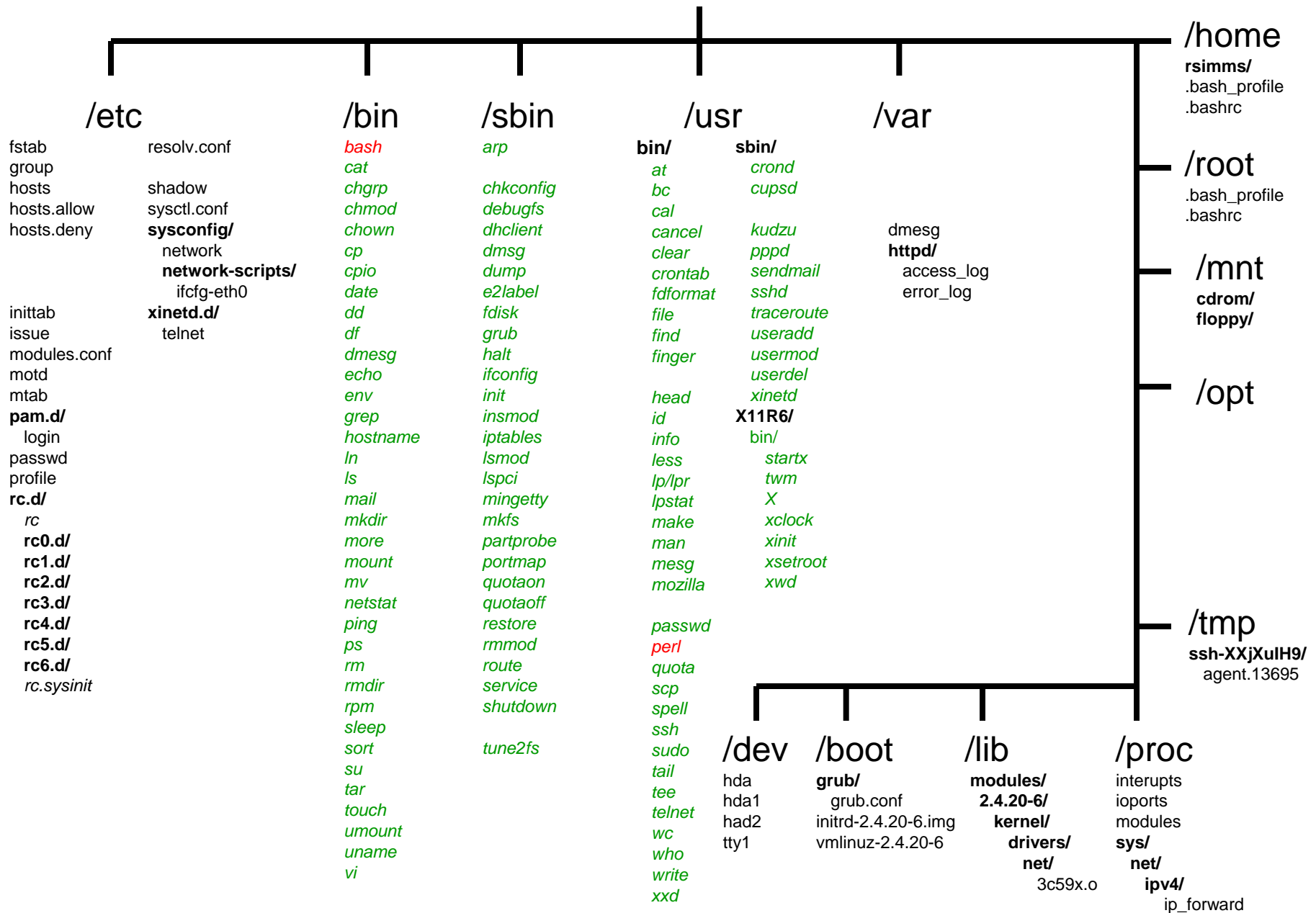
Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset



Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset

Example GNU/Linux Directory Structure

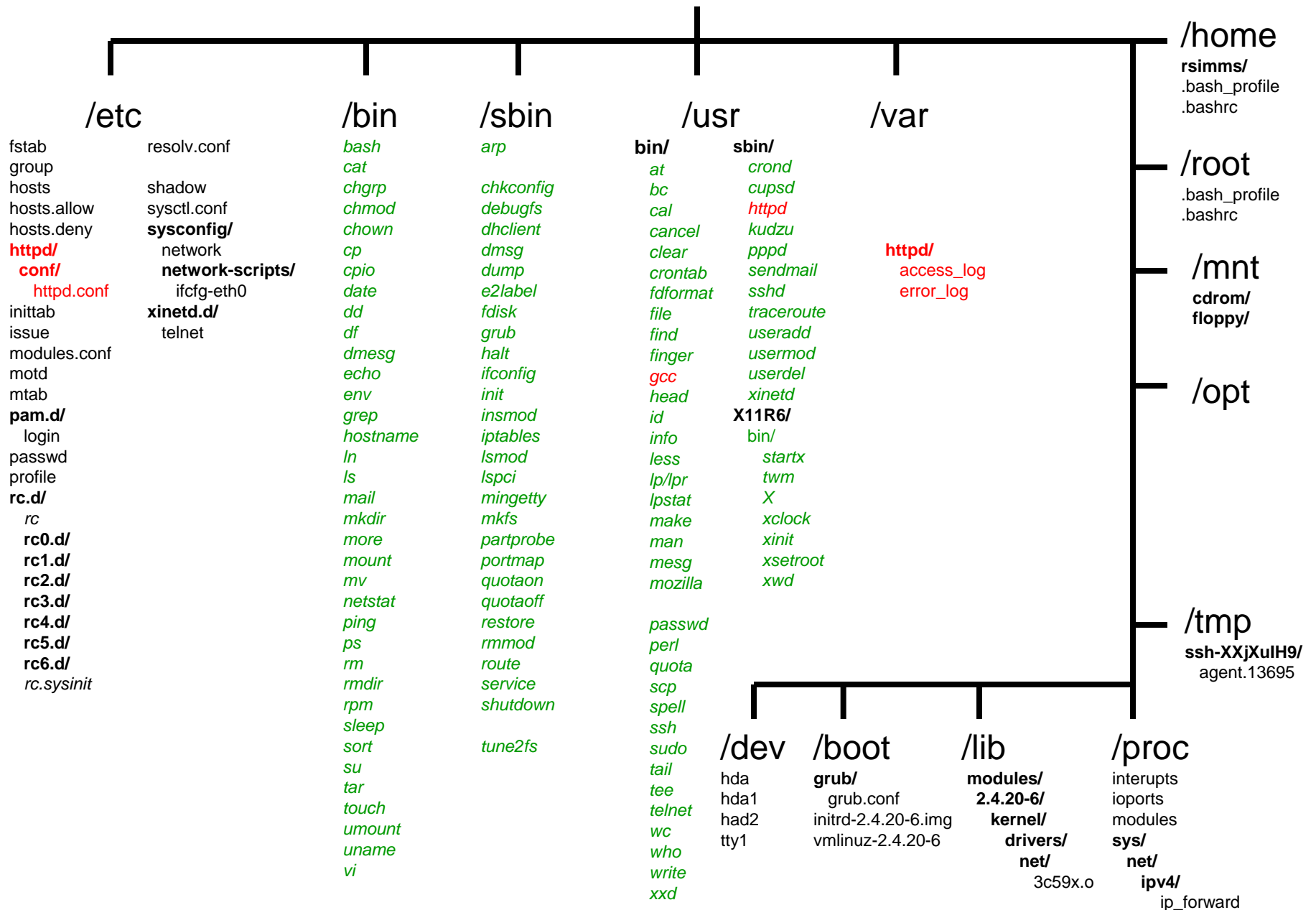
CIS 98 files, directories, commands



Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset shell keywords = if, then, else, case, for, while

Example GNU/Linux Directory Structure

CIS 164 files, directories, commands

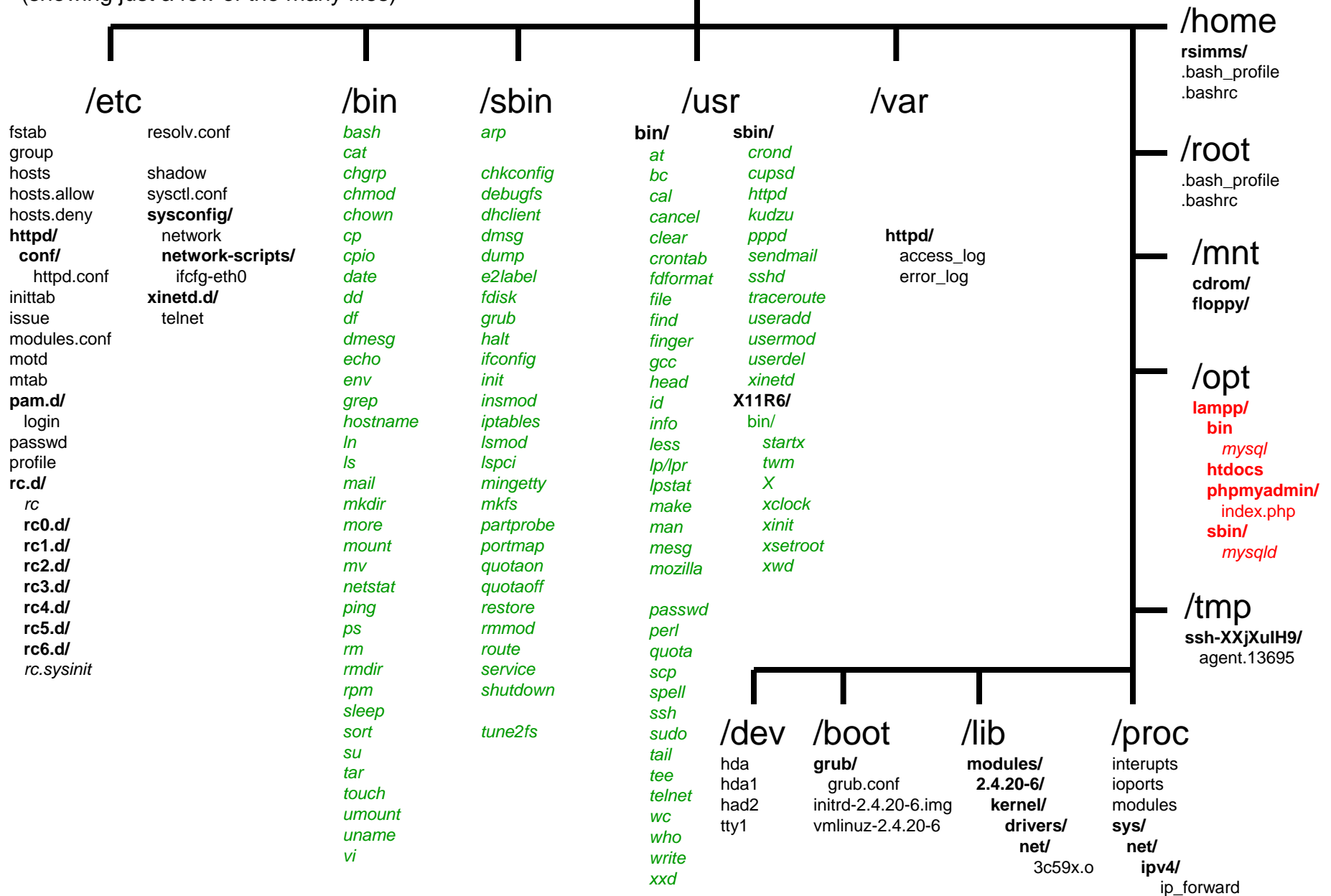


Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset shell keywords = if, then, else, case, for, while

Example GNU/Linux Directory Structure

(showing just a few of the many files)

CIS 165PH files, directories, commands

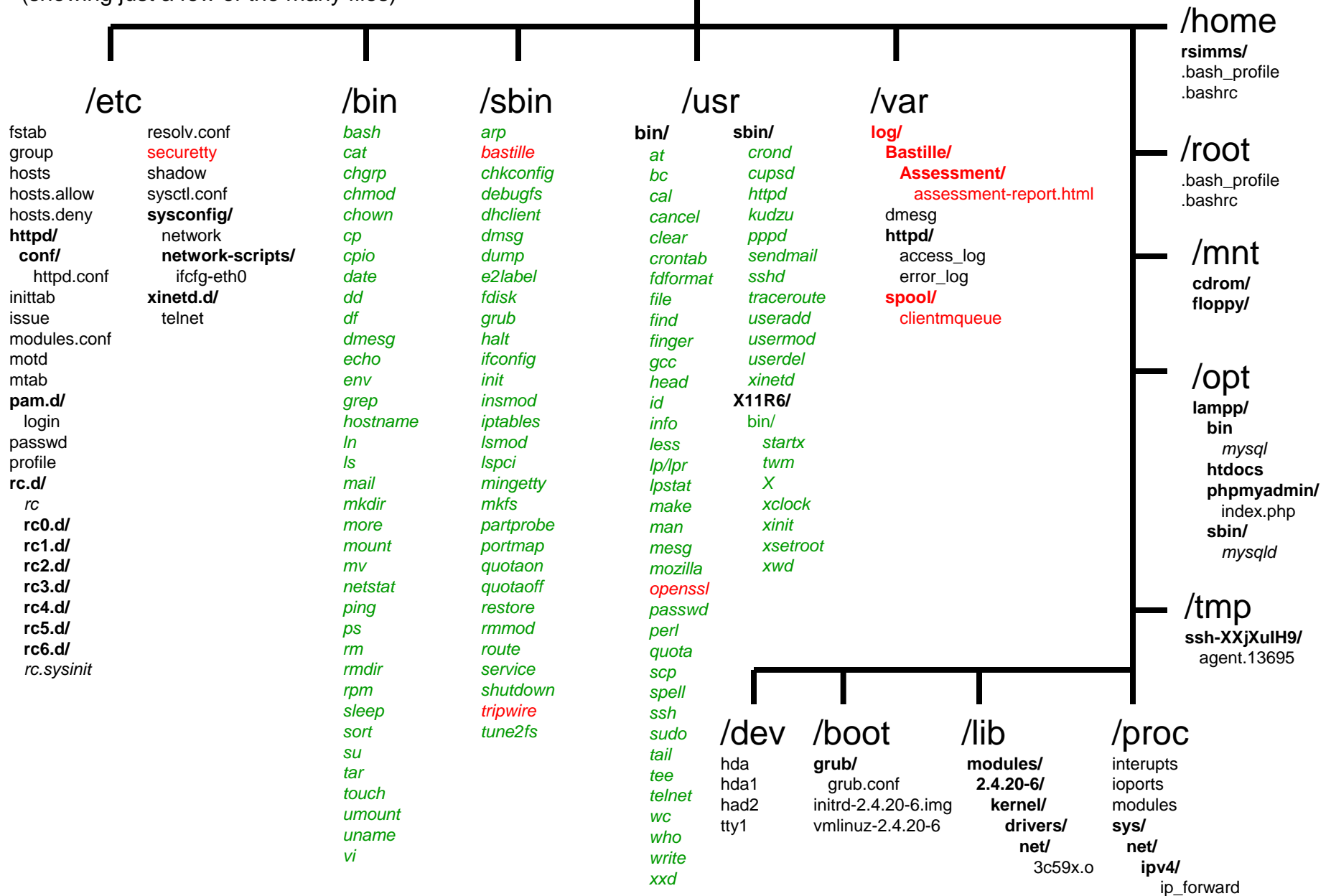


Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset shell keywords = if, then, else, case, for, while

Example GNU/Linux Directory Structure

(showing just a few of the many files)

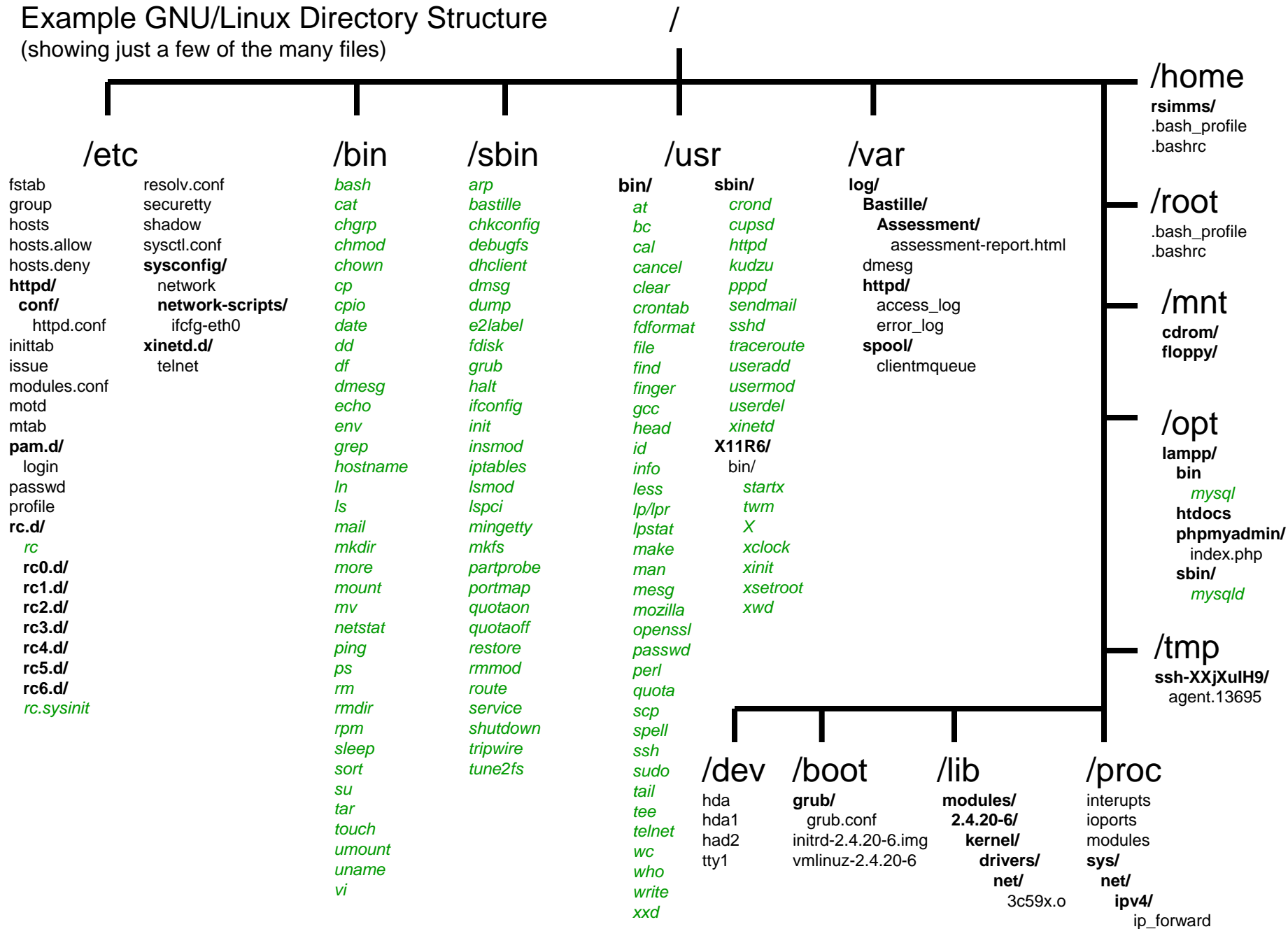
CIS 193 files, directories, commands



Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset shell keywords = if, then, else, case, for, while

Example GNU/Linux Directory Structure

(showing just a few of the many files)



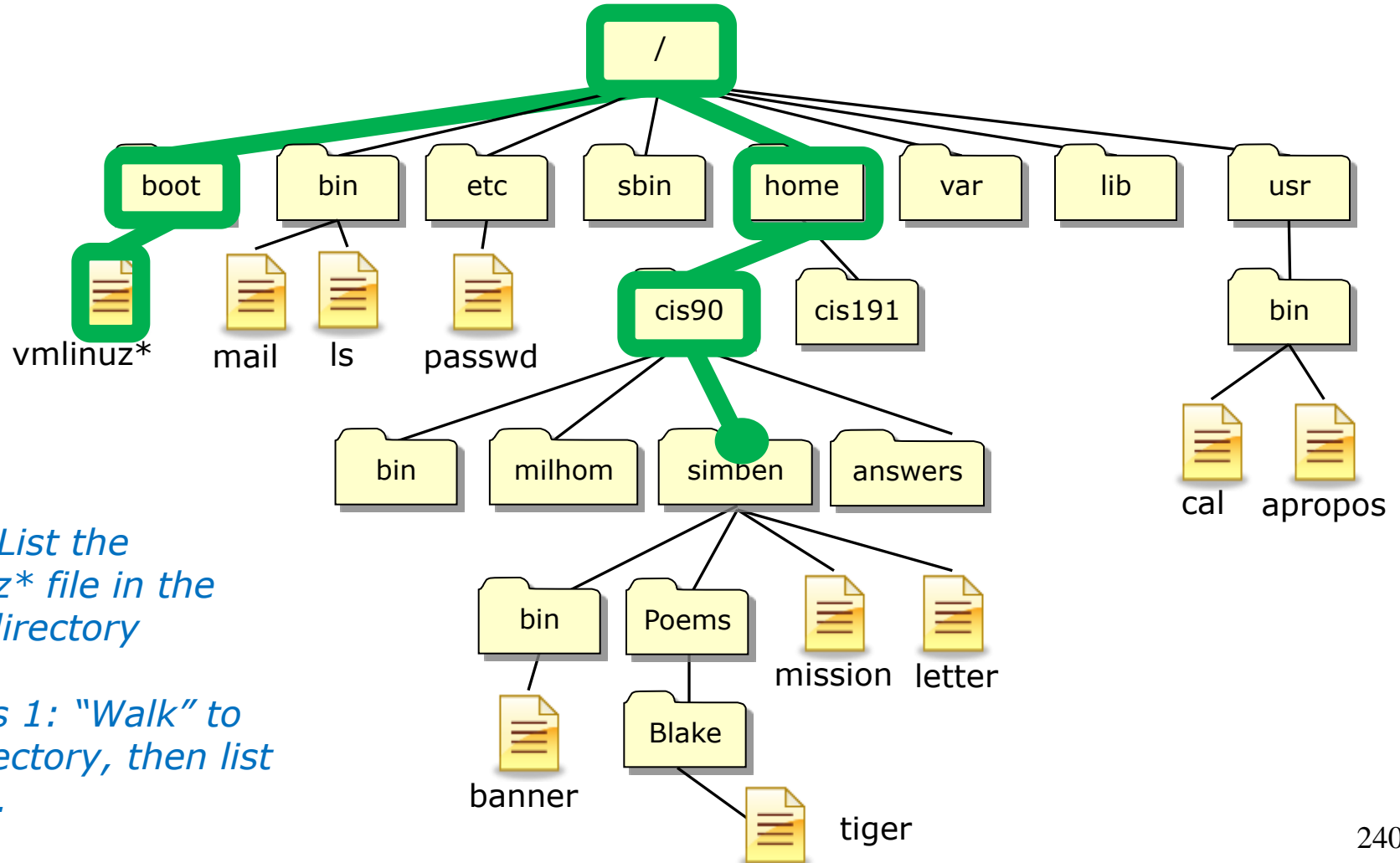
Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset shell keywords = if, then, else, case, for, while

Class Exercise

- Go to your home directory, type: **cd**
- Do a long listing of every file in your home directory and sub-directories and include inode numbers

ls -lR

Listing a file in another directory - Option 1



Task: List the `vmlinuz` file in the `/boot` directory*

Options 1: "Walk" to the directory, then list the file.

Option 1: Listing a file by navigating to the directory first

/home/cis90/simben/Poems/Blake \$ **cd** *start in your home directory*

/home/cis90/simben \$ **cd ..** *go up the tree*

/home/cis90 \$ **ls** *look around*

```
answers  davdon  farsha  hendaj  lyoben  mesmic  ramcar  simben
bin       depot   frocar  kanbry  marray  milhom  ramgus  verevi
calsea   ellcar  fyosea  kenrit  menfid  noreva  rawjes  wiljac
cis       evaand  guest   libkel  mescha  potjos  rodduk  zamhum
```

*student
home
directories*

/home/cis90 \$ **cd ..** *go up again*

/home \$ **ls** *look around*

```
cis172  cis90  cis98  gerlinde  guest  jimj  lost+found  rick  rsimms  turnin
```

*my home
directory*

*where labs are
submitted*

/home \$ **cd ..** *go up again*

/ \$ **ls** *look around*

```
bin  cgroup  etc  lib  media  mnt  opt  root  selinux  sys  u  var
boot dev  home  lost+found  misc  net  proc  sbin  srv  tmp  usr
```

/ \$ **cd boot** *go down into boot*

/boot \$ **ls** *look around*

```
config-2.6.32-220.23.1.el6.i686
```

```
config-2.6.32-71.el6.i686
```

```
efi
```

```
grub
```

```
initramfs-2.6.32-220.23.1.el6.i686.img
```

```
initramfs-2.6.32-71.el6.i686.img
```

```
symvers-2.6.32-220.23.1.el6.i686.gz
```

```
symvers-2.6.32-71.el6.i686.gz
```

```
System.map-2.6.32-220.23.1.el6.i686
```

```
System.map-2.6.32-71.el6.i686
```

```
vmlinuz-2.6.32-220.23.1.el6.i686
```

```
vmlinuz-2.6.32-71.el6.i686
```

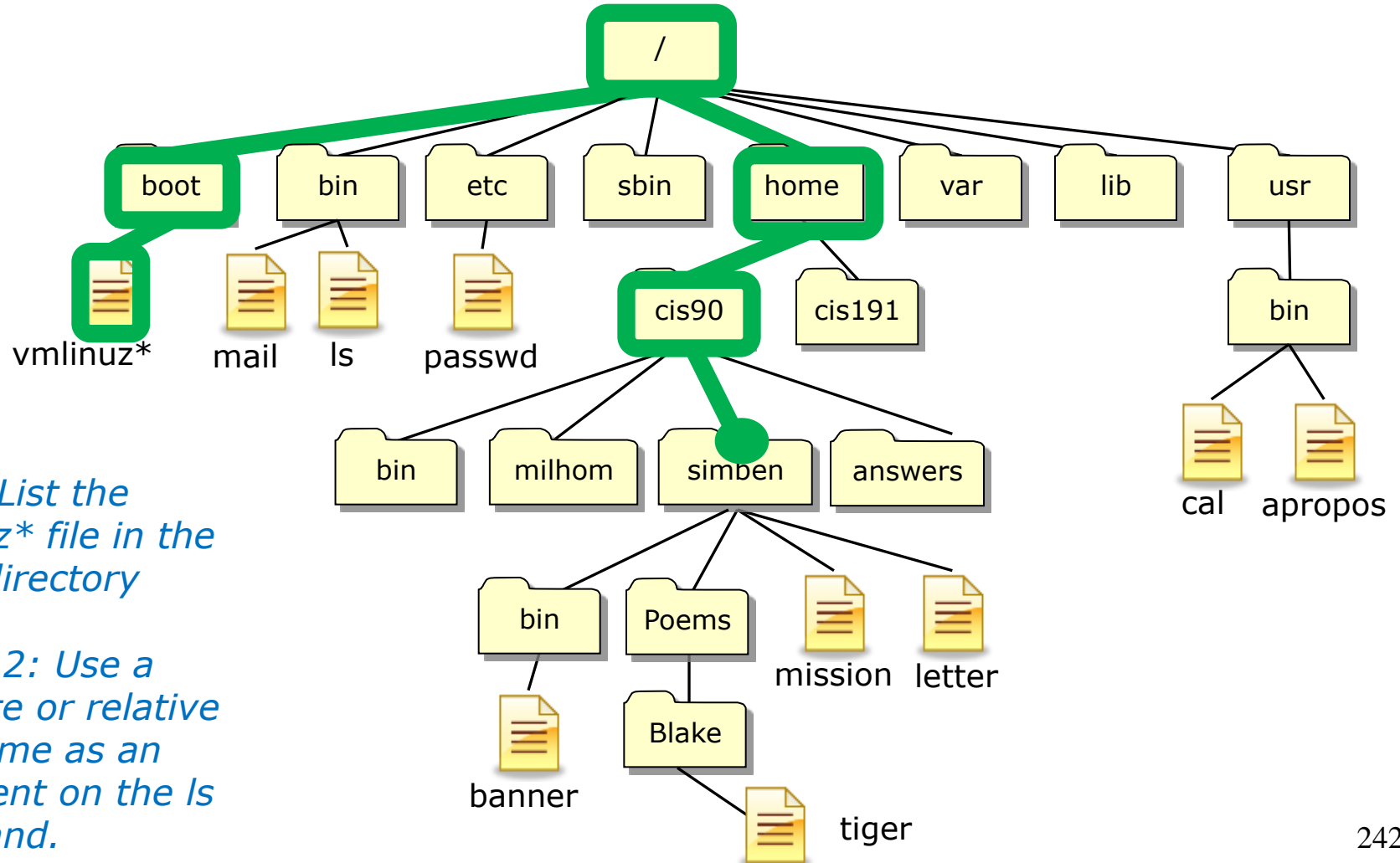
Newer Linux kernel

Older Linux kernel

/boot \$ **ls -l vmlinuz-2.6.32-220.23.1.el6.i686**

```
-rwxr-xr-x. 1 root root 3813888 Jun 18 09:14 vmlinuz-2.6.32-220.23.1.el6.i686
```


Listing a file in another directory - Option 2



Task: List the `vmlinuz` file in the `/boot` directory*

Option 2: Use a absolute or relative pathname as an argument on the `ls` command.



Option 2: Listing a file by using a pathname as an argument

/home/cis90/simben/Poems/Blake \$ **cd** *start in your home directory*

/home/cis90/simben \$ **ls -l /boot/vmlinuz-2.6.32-220.23.1.el6.i686** *using an absolute pathname as the argument*

```
-rwxr-xr-x. 1 root root 3813888 Jun 18 09:14 /boot/vmlinuz-2.6.32-220.23.1.el6.i686
```

/home/cis90/simben \$ *FYI, this is the Linux kernel*