Lesson Module Checklist

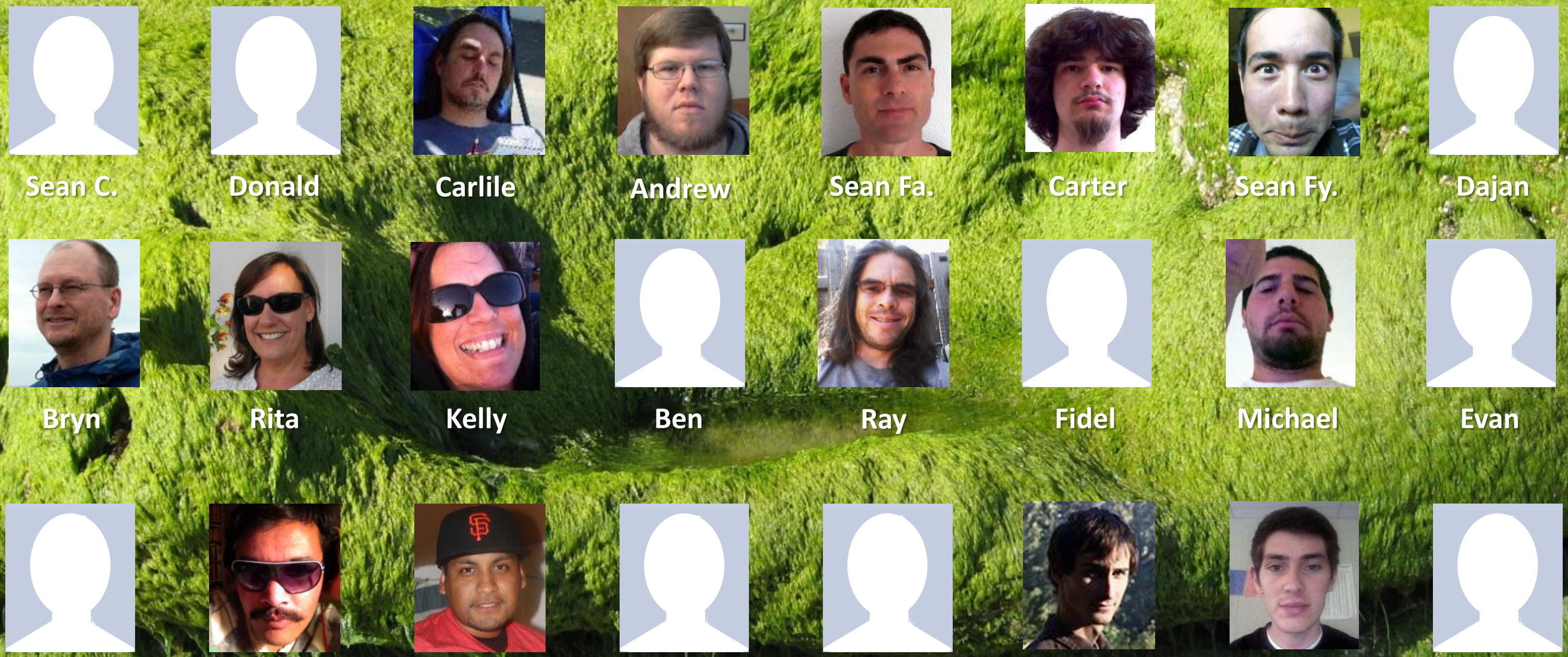
- Slides –
- Flash cards –
- Properties –
- Page numbers -
- 1st minute quiz –
- Web Calendar summary –
- Web book pages -
- Commands –
- Lab tested –

- Bring VTEA paper survey -

- CCC Confer room whiteboard –
- Wireless lapel mic backup battery -
- Backup slides, CCC info, handouts on flash drive -

CIS 90 - Lesson 4

Instructor: **Rich Simms**
Dial-in: **888-450-4821**
Passcode: **761867**



Email me (risimms@cabrillo.edu) a relatively current photo of your face for 3 points extra credit

Quiz

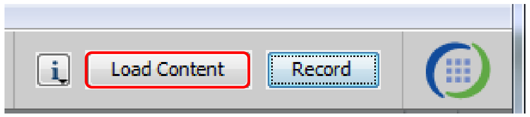
Please answer these questions **in the order** shown:

See electronic white board

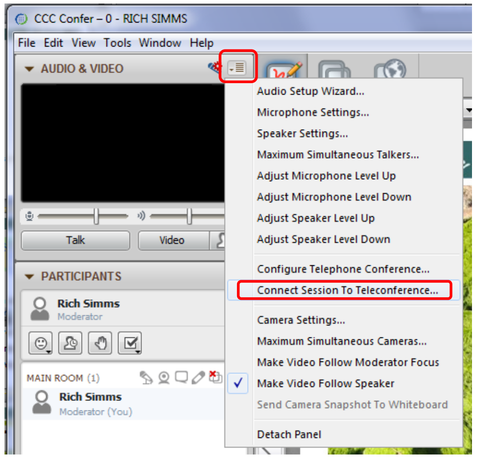
email answers to: `risimms@cabrillo.edu`

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

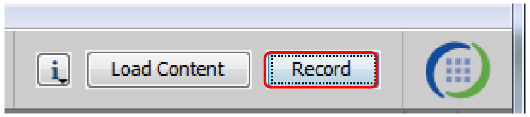
[ ] Load White Board with pics & quiz



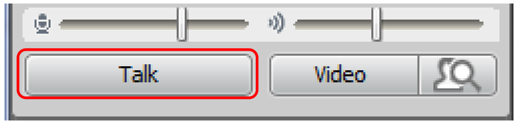
[ ] Connect session to Teleconference


CCC Confer – 0 - RICH SIMMS
File Edit View Tools Window Help
AUDIO & VIDEO
Audio Setup Wizard...
Microphone Settings...
Speaker Settings...
Maximum Simultaneous Talkers...
Adjust Microphone Level Up
Adjust Microphone Level Down
Adjust Speaker Level Up
Adjust Speaker Level Down
Configure Telephone Conference...
Connect Session To Teleconference...
Camera Settings...
Maximum Simultaneous Cameras...
Make Video Follow Moderator Focus
Make Video Follow Speaker
Send Camera Snapshot To Whiteboard
Detach Panel
Talk
Video
PARTICIPANTS
Rich Simms
Moderator
MAIN ROOM (1)
Rich Simms
Moderator (You)

[ ] Is recording on?



[ ] Toggle Talk button to not use Mic



[ ] Disable spelling on PowerPoint

[ ] Share slides, putties, Chrome and VLab

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 & Review
- Unix files
- Viewing files
- The UNIX Directory Hierarchy
- Navigating the file system
- File types
- Exercise: Enlightenment
- Wrap up

Previous material and assignment

1. Questions on previous material?

2. Lab 3 questions?

   - use mail -f uhistory to review what I'll grade
   - I'll grade using check3 script
   - clean up duplicates before last submittal

   - bash shell vs mail "shell"

Lab 2 Post Mortem

Lab 2 Results

Questions at end of lab	
Q1 – input from cmd line	4*
Q2 – input from keyboard	5*
Q3 – input from OS	1*

Each number shows where points were lost for incorrect or missing answers on Lab 2

** More details explained in the following slides*

Lab Steps		
1)	show shell	0
2)	type commands	23*
3)	echo variables	23*
4)	set TERM	6
5)	upper/lower case	2
6)	who –g	1
7)	number of arguments	4*
8)	CR and quotes	9*
9)	; to separate commands	6*
10)	change password	0
11)	uname options	9*
12)	banner	4
13)	finger	2
14)	id	0
15)	man	1
16)	whatis vs man –f	8*
17)	tryme	11*
18)	who –q	11*
19)	man –k vs apropos	14*
20)	info bash	2
21)	Google	0
22)	sqrt	1

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
```

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

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

There was no command named bogus on the path

Lab 2 Results – S2

3 arguments

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

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

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

The PS1 variable contains the your primary prompt string definition.

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

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

```
/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:.
```

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

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

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

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 $BOGUS

/home/cis90/simben $
```

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

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<newline>
> 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.

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.

```
/home/cis90/simben $ echo red white \<newline>
> 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.

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

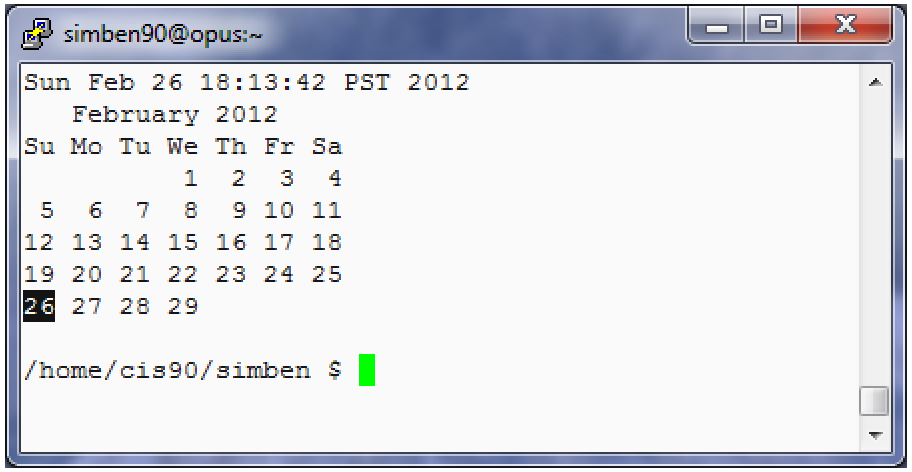
```
/home/cis90/simben $ echo $PS2
>
```

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
```



```
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. Using the **uname** command what options would you use to display just the operating system, it's kernel release numbers and the machine's network node hostname?

(Hint: Use the **man uname** command)

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
```

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

```
/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 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
```

$\underbrace{\texttt{opus.cabrillo.edu}}_{\textit{node hostname}}\ \underbrace{\texttt{2.6.18-164.el5}}_{\textit{kernel release}}\ \underbrace{\texttt{GNU/Linux}}_{\textit{OS}}$

*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
```

$\underbrace{\texttt{GNU/Linux}}_{\textit{OS}}\ \underbrace{\texttt{2.6.32-27-generic}}_{\textit{kernel release}}\ \underbrace{\texttt{eko-04}}_{\textit{node hostname}}$

This is one of the really cool things about Linux and the GNU General Public License … if you don't like something about it 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?

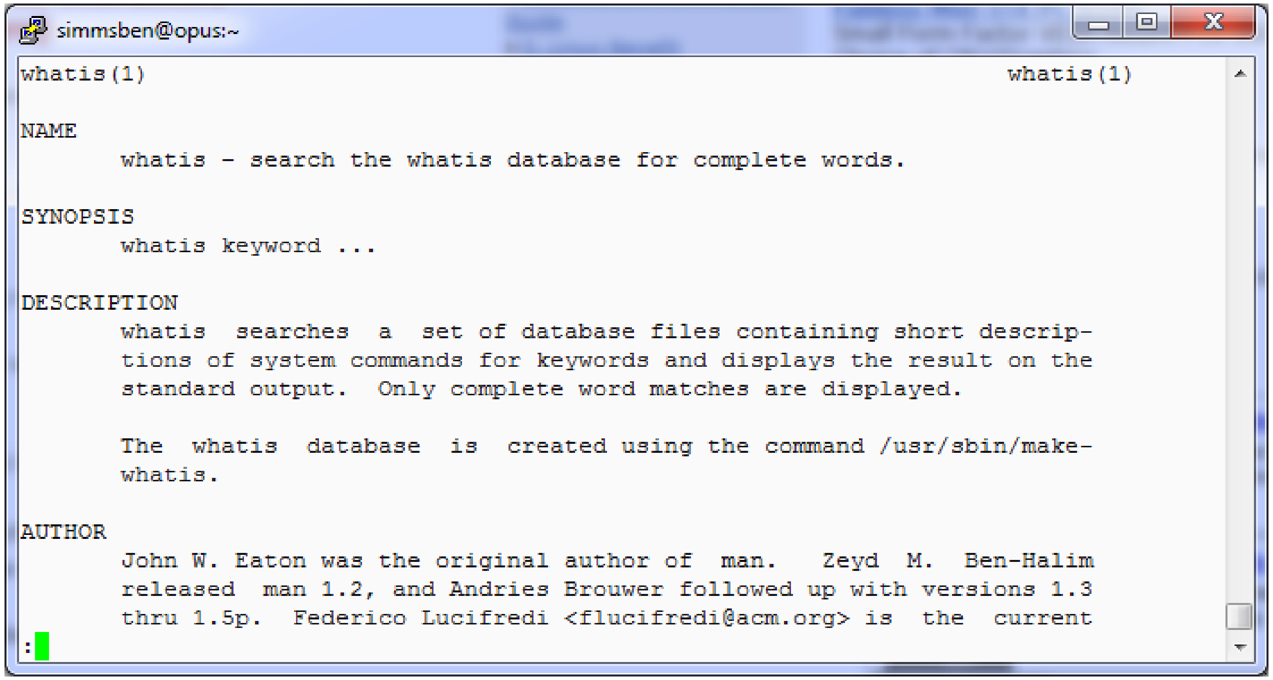
man -f bc

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**



```
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 descrip-
       tions 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).
```

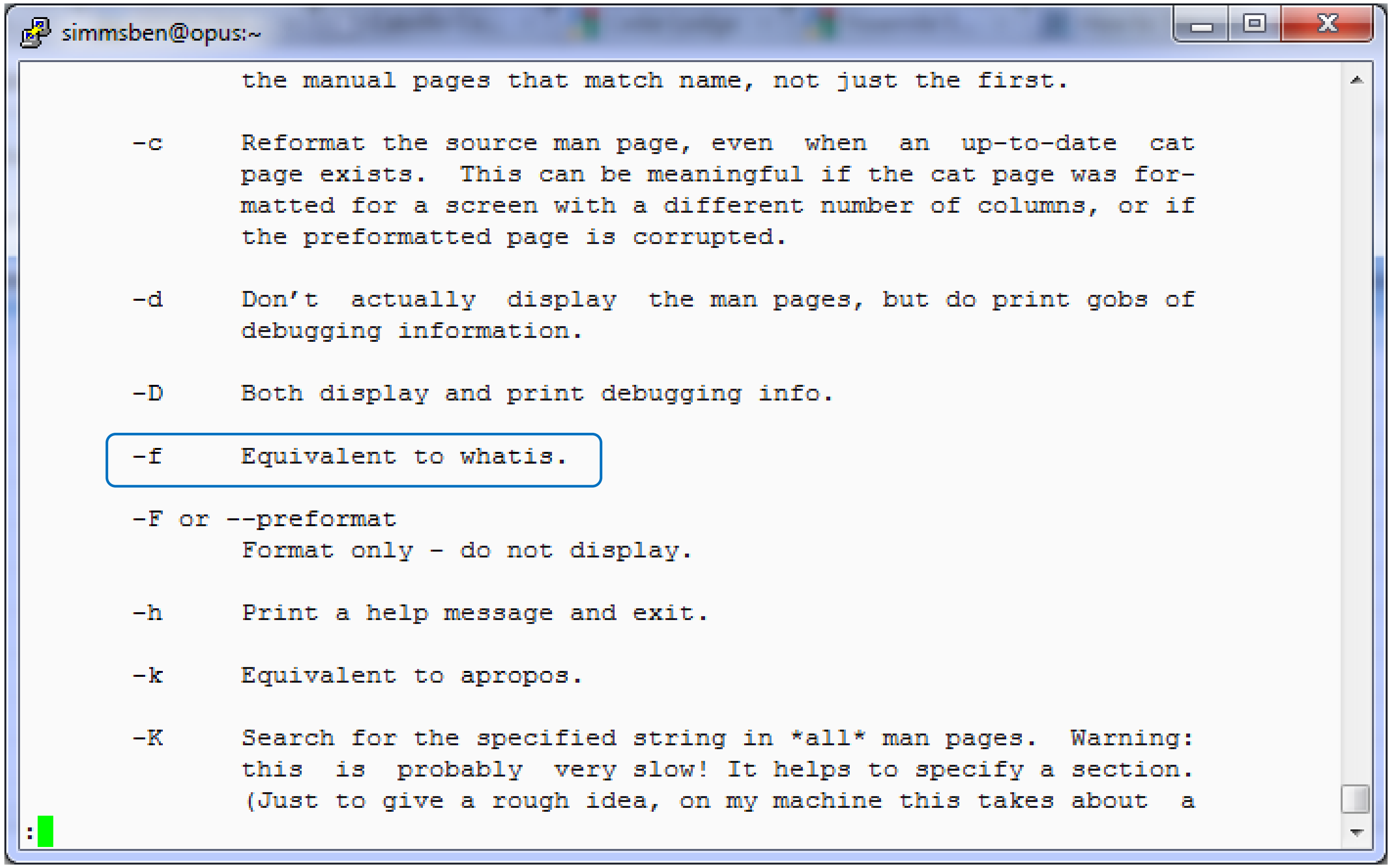
*Compare output with **man –f** command*

```
/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

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
```

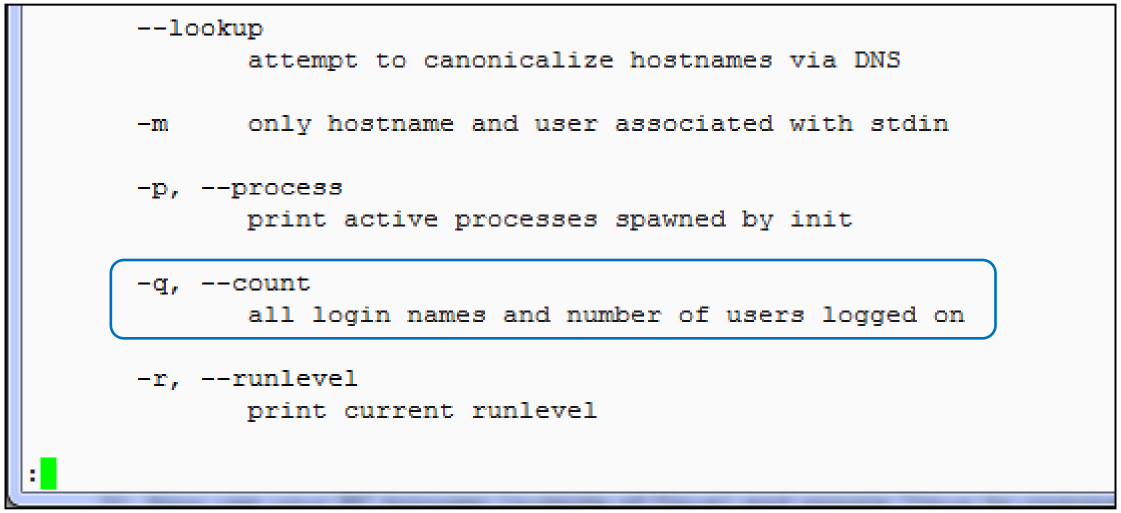
***cat** shows the actual **tryme** script itself*

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.*

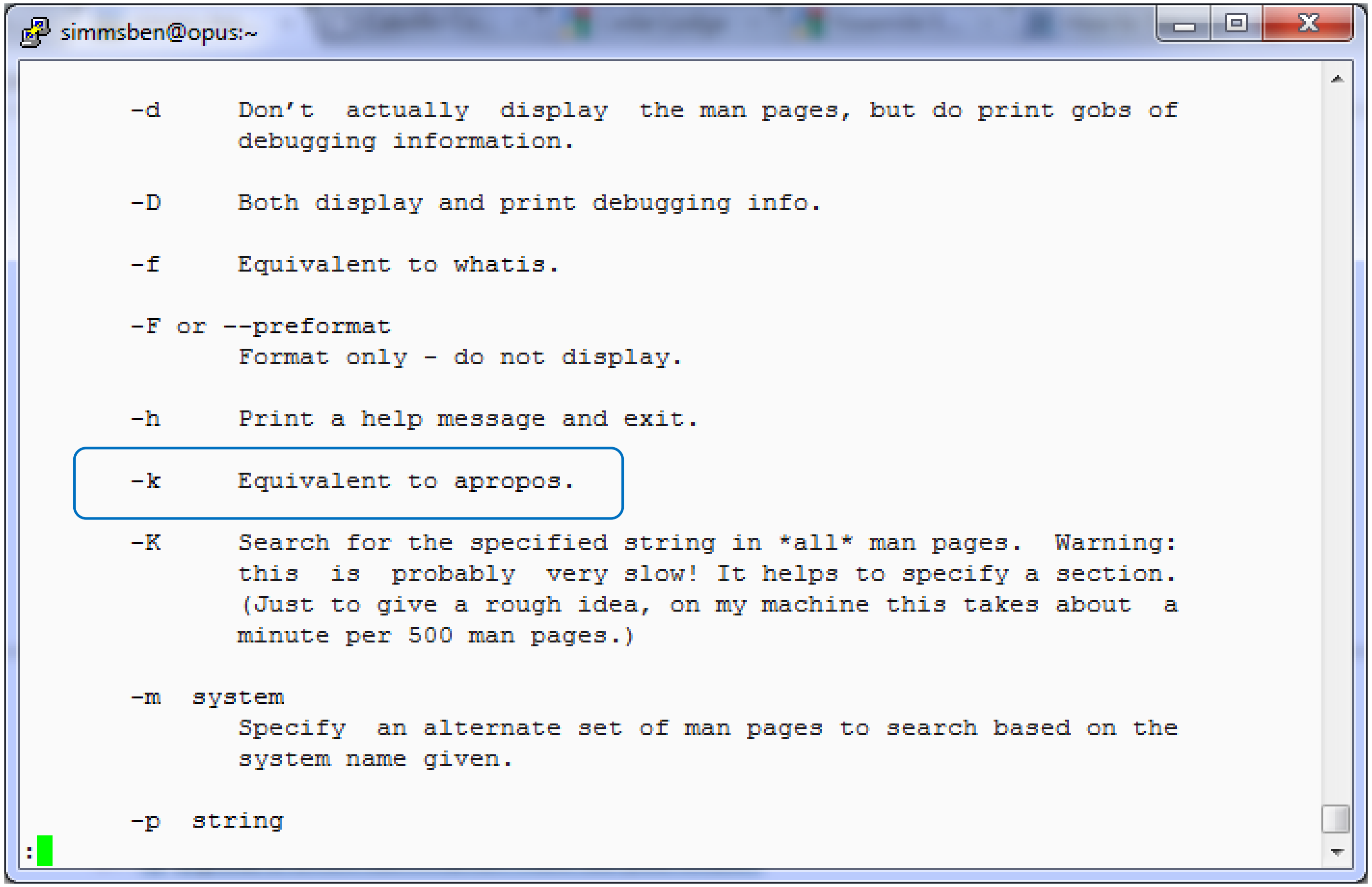
```
[rsimms@opus ~]$ who -q
helrog90 jimmel90 rsimms saljac193 vascar193
# users=5
```

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**



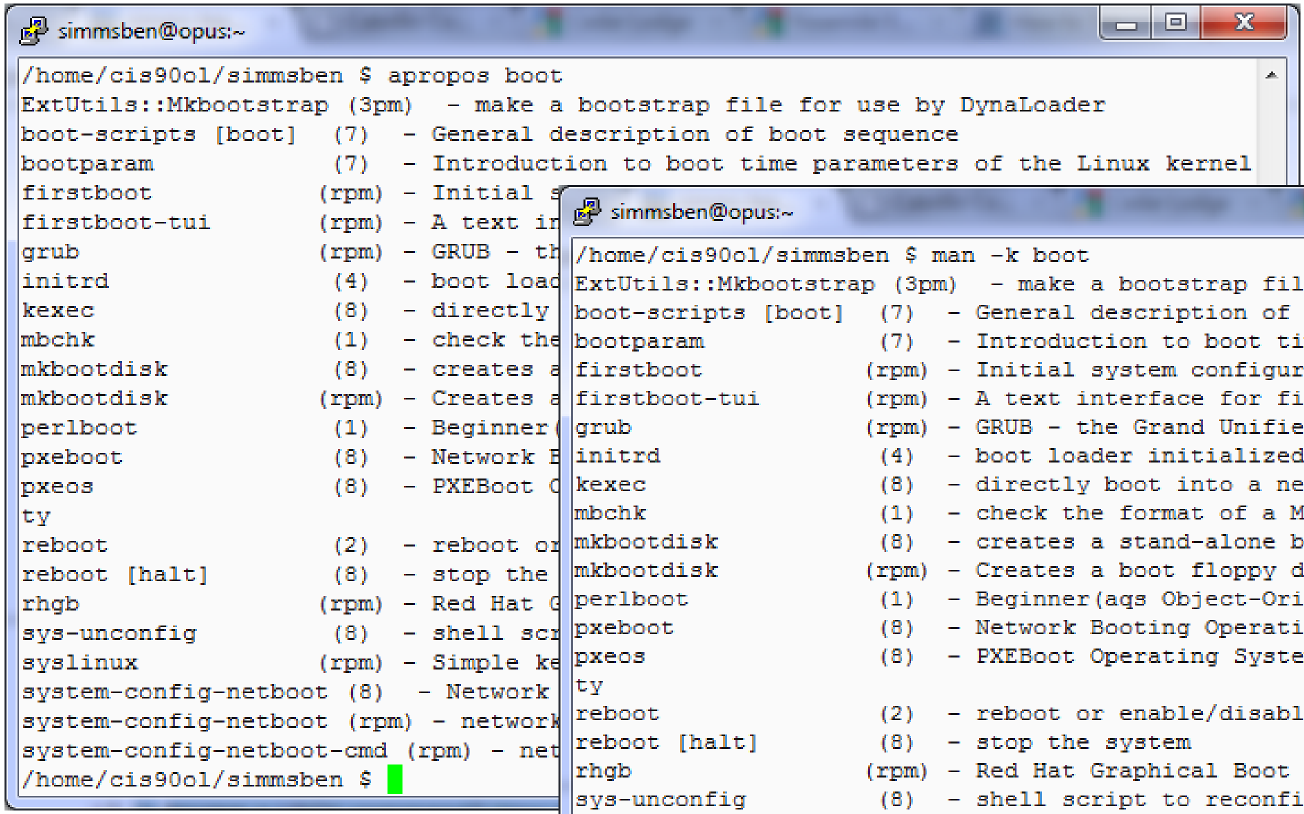
```
simmsben@opus:~

       -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*

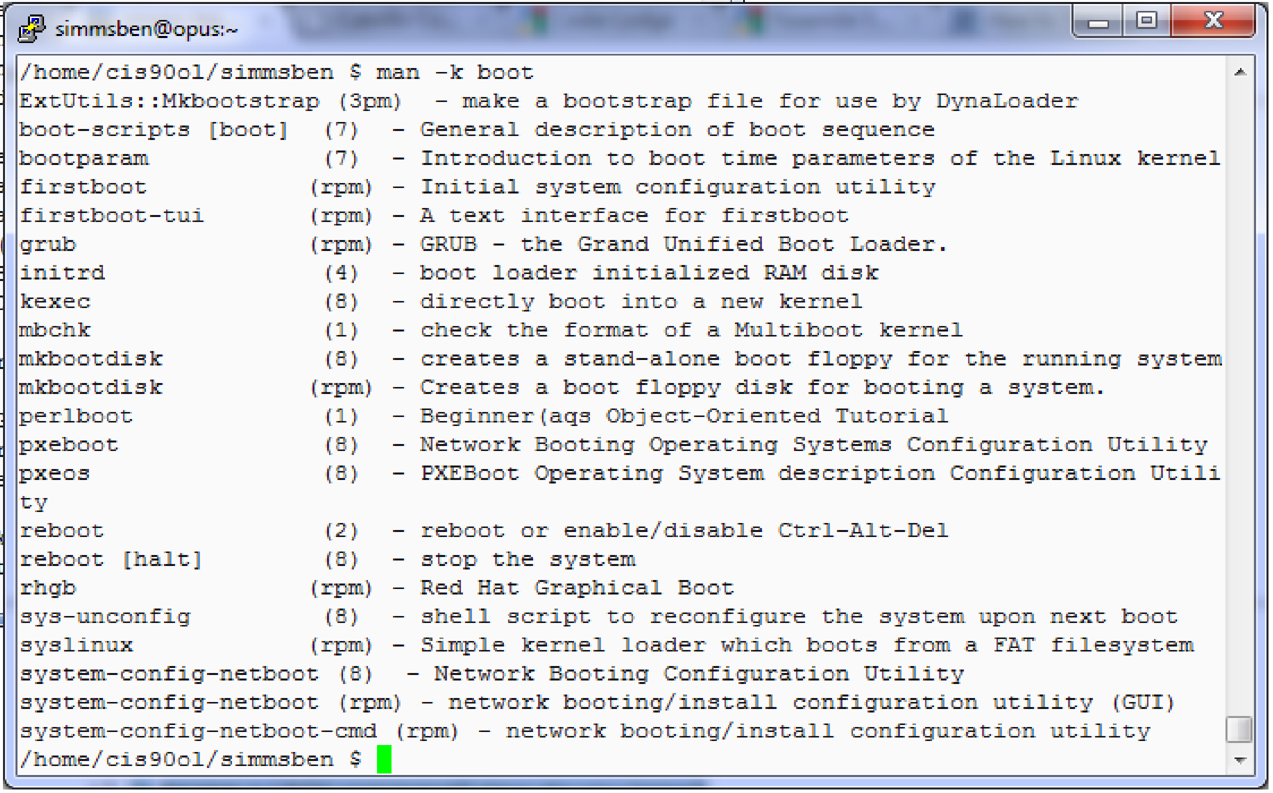
Lab 2 Results – S19

Output from **apropos boot**



```
simmsben@opus:~
/home/cis90ol/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 s
firstboot-tui       (rpm) - A text in
grub                (rpm) - GRUB - th
initrd               (4)  - boot load
kexec                (8)  - directly
mbchk                (1)  - check the
mkbootdisk           (8)  - creates a
mkbootdisk          (rpm) - Creates a
perlboot             (1)  - Beginner(
pxeboot              (8)  - Network B
pxeos                (8)  - PXEBoot O
ty
reboot               (2)  - reboot or
reboot [halt]        (8)  - stop the
rhgb                (rpm) - Red Hat G
sys-unconfig         (8)  - shell scr
syslinux            (rpm) - Simple ke
system-config-netboot (8)  - Network
system-config-netboot (rpm) - network
system-config-netboot-cmd (rpm) - net
/home/cis90ol/simmsben $
```

Output from **man -k boot**



```
simmsben@opus:~
/home/cis90ol/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/cis90ol/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)
```

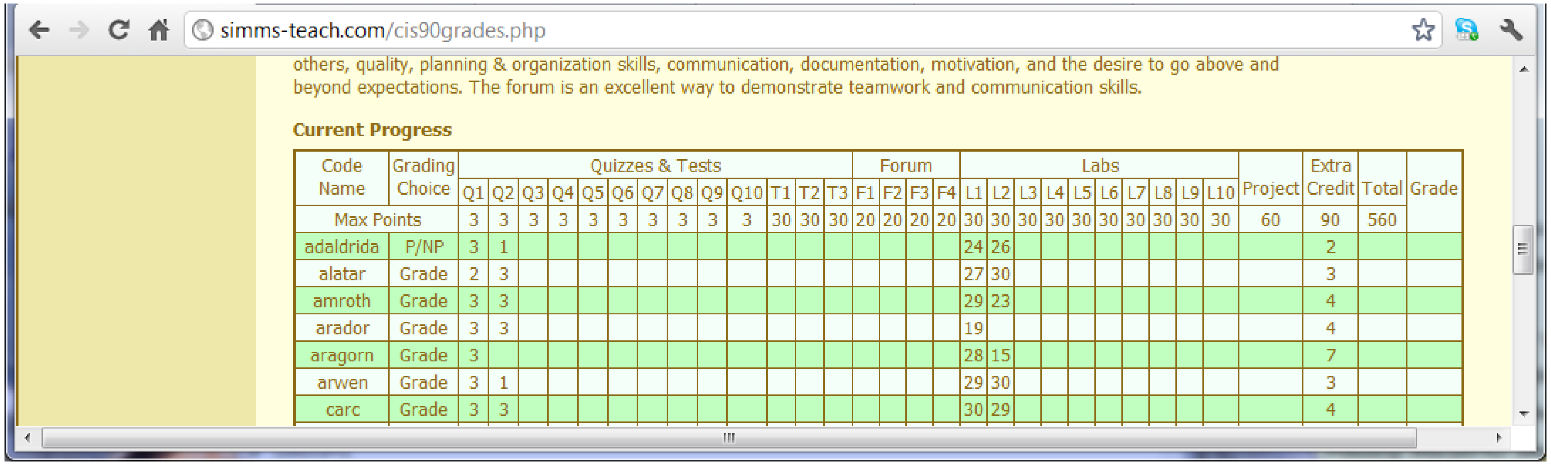
```
/home/cis90/simben $ uname
Linux
```

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

Housekeeping

- Grades posted on website

  (send me filled in student survey to get your grading code name)
- Graded labs placed in your home directory
- Answers to labs in /home/cis90/answers/ directory
- Lab 3 and five forum posts due tonight at 11:59PM (Opus time)



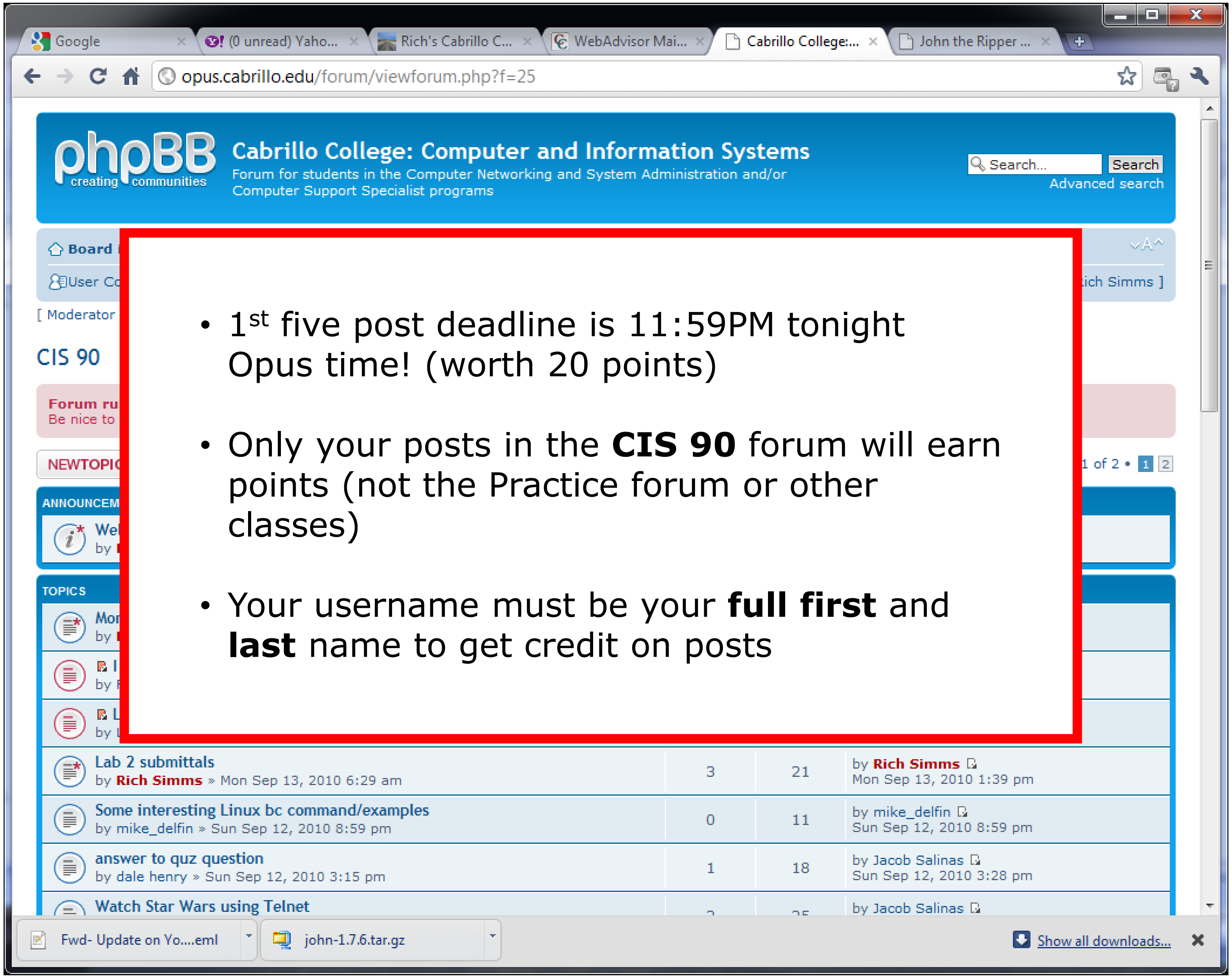
simms-teach.com/cis90grades.php

others, quality, planning & organization skills, communication, documentation, motivation, and the desire to go above and beyond expectations. The forum is an excellent way to demonstrate teamwork and communication skills.

Current Progress

Code Name	Grading Choice	Quizzes & Tests													Forum				Labs										Project	Extra Credit	Total	Grade
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	T1	T2	T3	F1	F2	F3	F4	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10				
Max Points		3	3	3	3	3	3	3	3	3	3	30	30	30	20	20	20	20	30	30	30	30	30	30	30	30	30	30	60	90	560	
adaldrida	P/NP	3	1																24	26										2		
alatar	Grade	2	3																27	30										3		
amroth	Grade	3	3																29	23										4		
arador	Grade	3	3																19											4		
aragorn	Grade	3																	28	15										7		
arwen	Grade	3	1																29	30										3		
carc	Grade	3	3																30	29										4		

CIS 90 - Lesson 4



- 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

opus.cabrillo.edu/forum/viewforum.php?f=25

Cabrillo College: Computer and Information Systems
Forum for students in the Computer Networking and System Administration and/or Computer Support Specialist programs

Lab 2 submittals by Rich Simms » Mon Sep 13, 2010 6:29 am	3	21	by Rich Simms Mon Sep 13, 2010 1:39 pm
Some interesting Linux bc command/examples by mike_delfin » Sun Sep 12, 2010 8:59 pm	0	11	by mike_delfin Sun Sep 12, 2010 8:59 pm
answer to quz question by dale henry » Sun Sep 12, 2010 3:15 pm	1	18	by Jacob Salinas Sun Sep 12, 2010 3:28 pm
Watch Star Wars using Telnet			by Jacob Salinas

```
[rsimms@oslab lab04]$ date
Wed Sep 19 09:06:08 PDT 2012
[rsimms@oslab lab04]$
[rsimms@oslab lab04]$ grep '^0$' forum | wc -l
9
[rsimms@oslab lab04]$ grep '4' forum | wc -l
4
[rsimms@oslab lab04]$ grep '8' forum | wc -l
1
[rsimms@oslab lab04]$ grep '12' forum | wc -l
2
[rsimms@oslab lab04]$ grep '16' forum | wc -l
1
[rsimms@oslab lab04]$ grep '20' forum | wc -l
7
[rsimms@oslab lab04]$ bc
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
24*20
480
(9*0)+(4*4)+(1*8)+(2*12)+(1*16)+(7*20)
204
480-204
276
quit
```

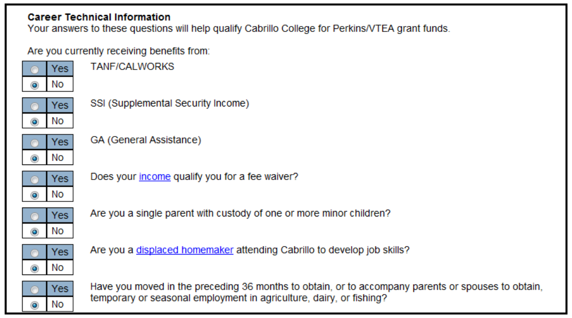
Perkins (VTEA) Web Advisor Survey Instructions

If you already filled this out in another class you don't need to do it again. This is the online survey for online classes.

- Log on to "www.cabrillo.edu" and go to the Cabrillo College Home Page
  - Select "WEBADVISOR" (from the listing near the top of the page below the Cabrillo College Logo)
  - Select the "LOG IN" tab
  - Fill-in the "User ID" and "Password"
  - Click on "SUBMIT"

- Select "STUDENTS: Click Here" (navy blue bar)
  - Under "Academic Profile" Click on "Student Update Form"
    - Use drop down list under "Select the earliest term for which you are registered" and click on the current term.
    - Select "SUBMIT"

- Scroll down to the "Career Technical Information"
  - **Answer questions** by clicking on the circle to the left of your "Yes" or "No" answers
    - You can get details about a question by clicking on blue underlined phrase
  - After answering all questions Select "SUBMIT"
  - Then "LOG OUT"



Career Technical Information
Your answers to these questions will help qualify Cabrillo College for Perkins/VTEA grant funds.

Are you currently receiving benefits from:

Yes [✓ No] TANF/CALWORKS

Yes [✓ No] SSI (Supplemental Security Income)

Yes [✓ No] GA (General Assistance)

Yes [✓ No] Does your income qualify you for a fee waiver?

Yes [✓ No] Are you a single parent with custody of one or more minor children?

Yes [✓ No] Are you a displaced homemaker attending Cabrillo to develop job skills?

Yes [✓ No] Have you moved in the preceding 36 months to obtain, or to accompany parents or spouses to obtain, temporary or seasonal employment in agriculture, dairy, or fishing?

Thank you for taking a few minutes to help Cabrillo receive funding to support student services for CTE programs at Cabrillo College.

UNIX Files

File Systems
Linux

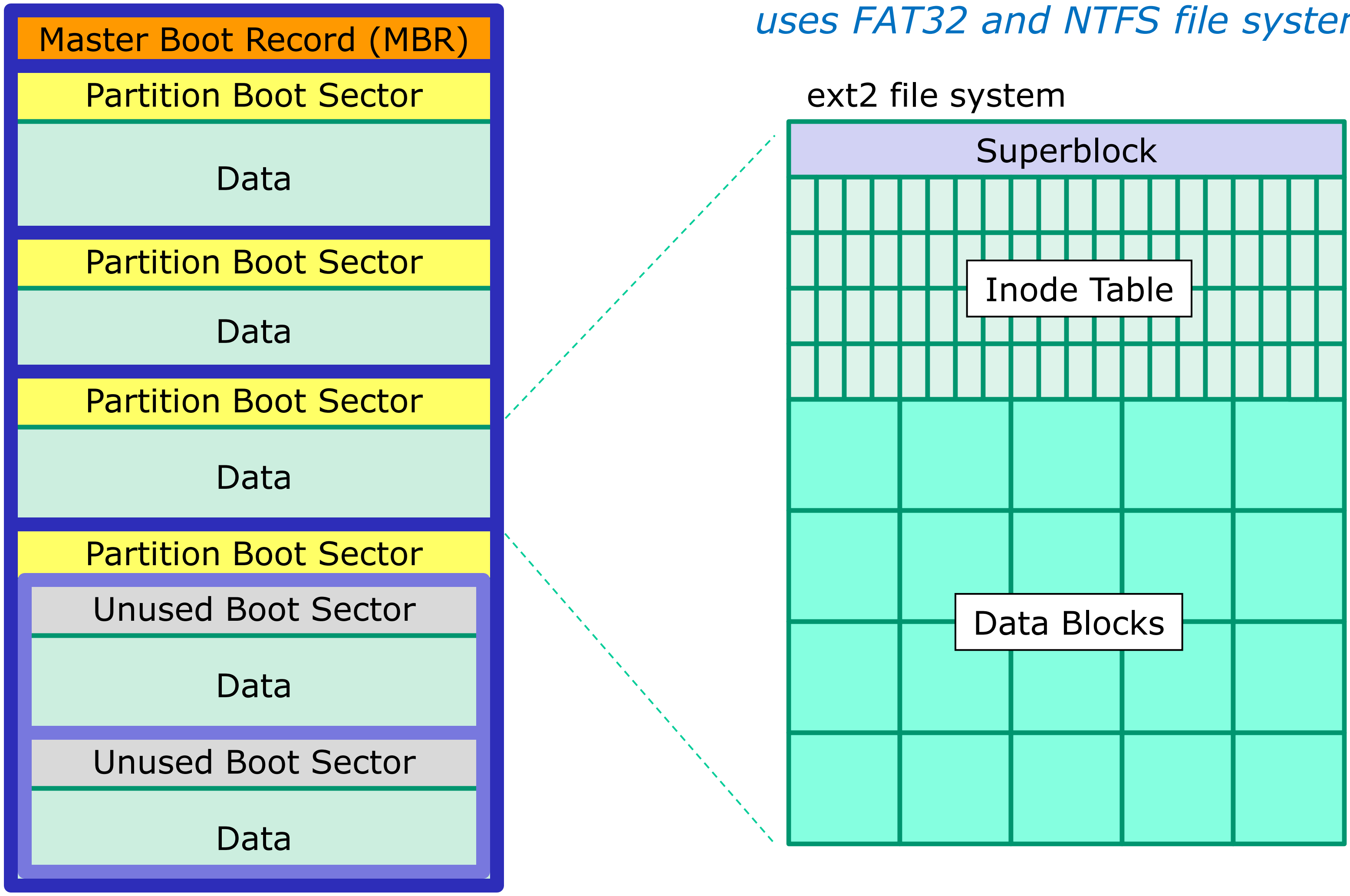
A typical hard drive

This is where your files actually reside

File Systems

Linux

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.



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?"
```

name

+

inode

+

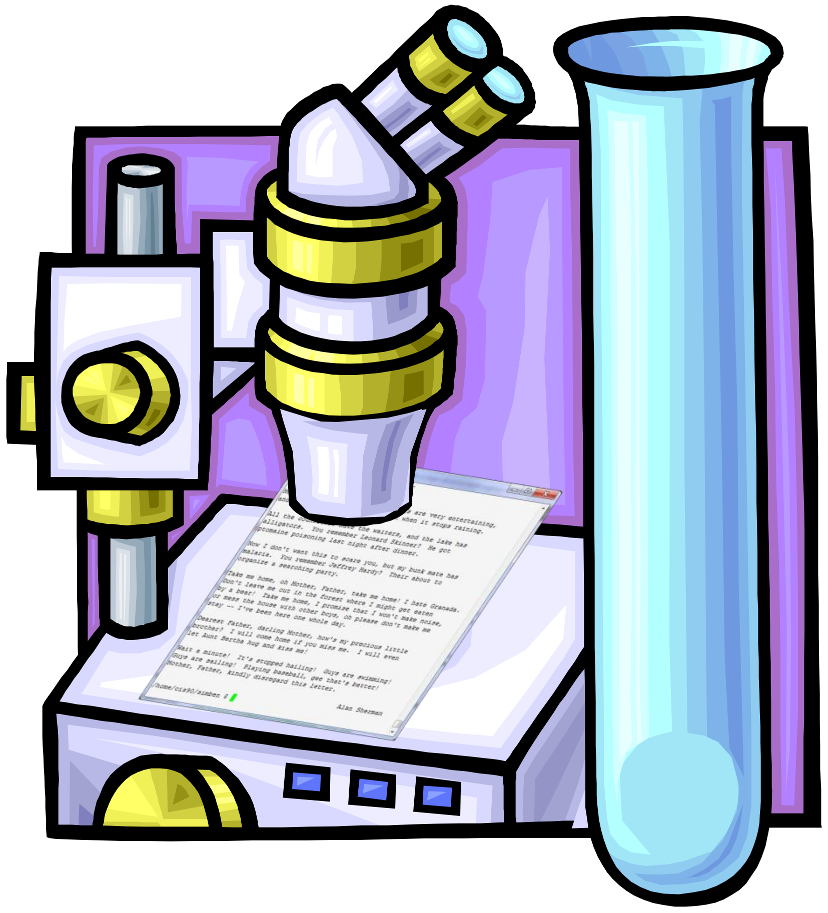
data

UNIX Files
Directories are files too

- Directories are also files
- The data in a directory file includes pairs of filenames and inode numbers (kind of like a phone book)
- Every directory can have 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.

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

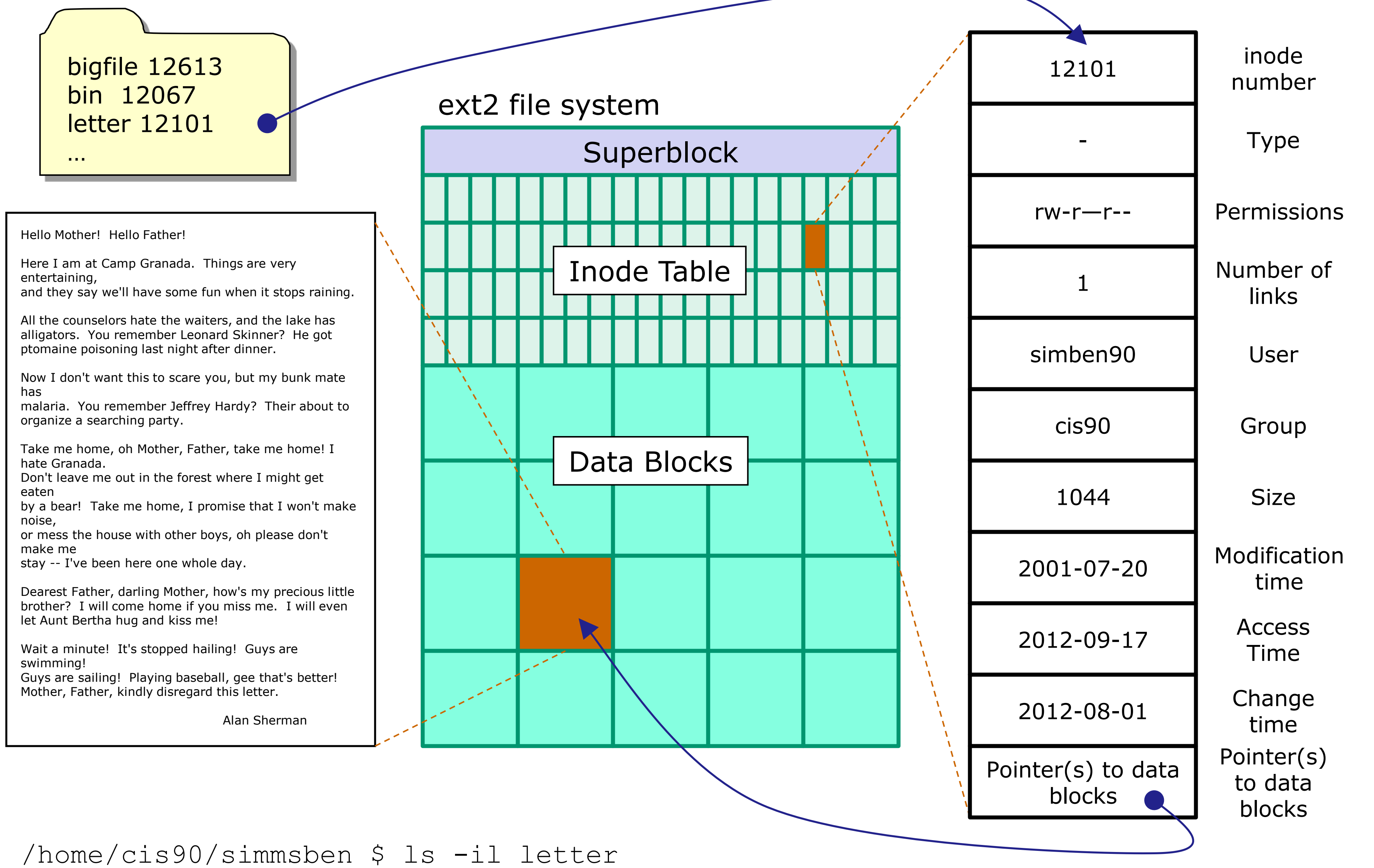


```
ls -l /home/cis90/simben/letter
cat /home/cis90/simben/letter
```

Cabrillo College est. 1959

CIS 90 - Lesson 4

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



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

```
/home/cis90/simmsben $ ls -il letter
12101 -rw-r--r--. 1 simben90 cis90 1044 Jul 20  2001 letter
```

Unix Filename Conventions

UNIX file name conventions

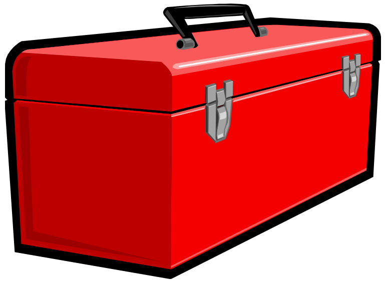
Unix filenames are case sensitive

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

Don’t use the following characters in filenames

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



More commands for your toolbox

Commands for this lesson

	• cat	*to print a text file*
NEW	• more	*to print a large text file by scrolling down*
NEW	• less	*to print a large text file by scrolling down and up*
NEW	• head	*to print the beginning lines of a text file*
NEW	• tail	*to print the last lines of a text file*
NEW	• wc	*count the words and lines in a text file*
NEW	• xxd	*view a binary data file using a hex dump*
	• cd	*change to a different directory*
	• ls	*list files*
NEW	• pwd	*print name of current/working directory*
	• file	*show additional file information*
	• type	*show location of command on path*

Viewing Text Files

Viewing text files:

• 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*
• file	*useful for identifying text/binary files*

Identifying text files

```
/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 $
```

*Use the **file** command to identify text files*

We learned about the file command in Lesson 2

cat command
viewing single text file

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

```
/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.
```

< 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

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

```
/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 $
```

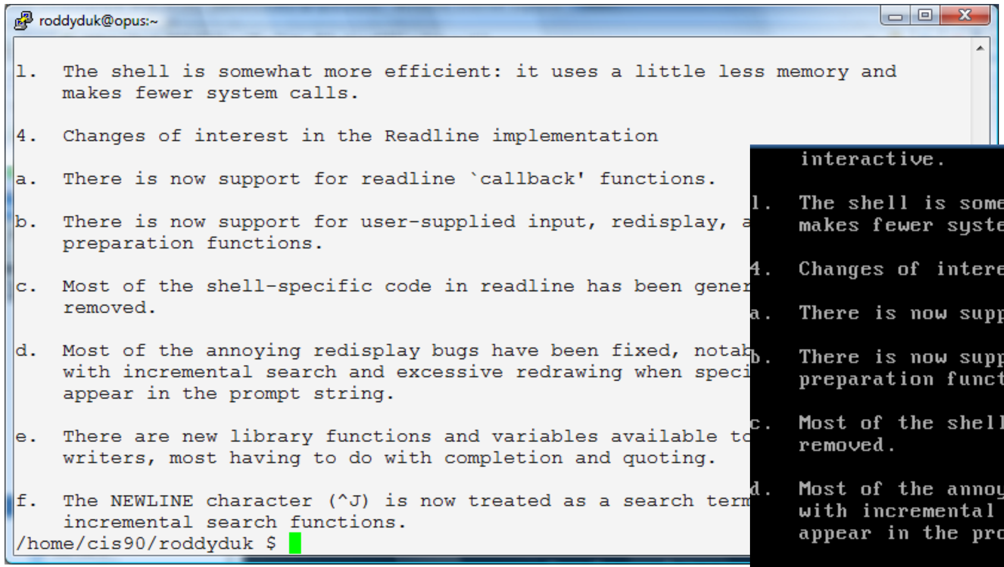
spellk

letter

cat command

viewing long text files

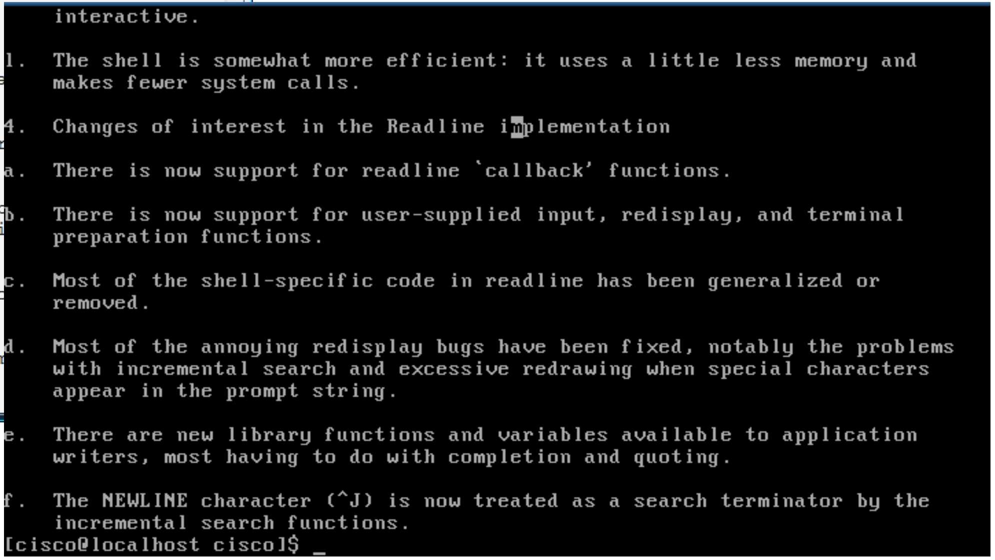
- Problem: if you **cat** really long files the text at the beginning is scrolled off and cannot be read.
- For example: **`cat /usr/share/doc/bash-3.2/NEWS`**



```
roddyduk@opus:~
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, a
    preparation functions.

c.  Most of the shell-specific code in readline has been gener
    removed.

d.  Most of the annoying redisplay bugs have been fixed, notab
    with incremental search and excessive redrawing when speci
    appear in the prompt string.

e.  There are new library functions and variables available to
    writers, most having to do with completion and quoting.

f.  The NEWLINE character (^J) is now treated as a search term
    incremental search functions.
/home/cis90/roddyduk $
```

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

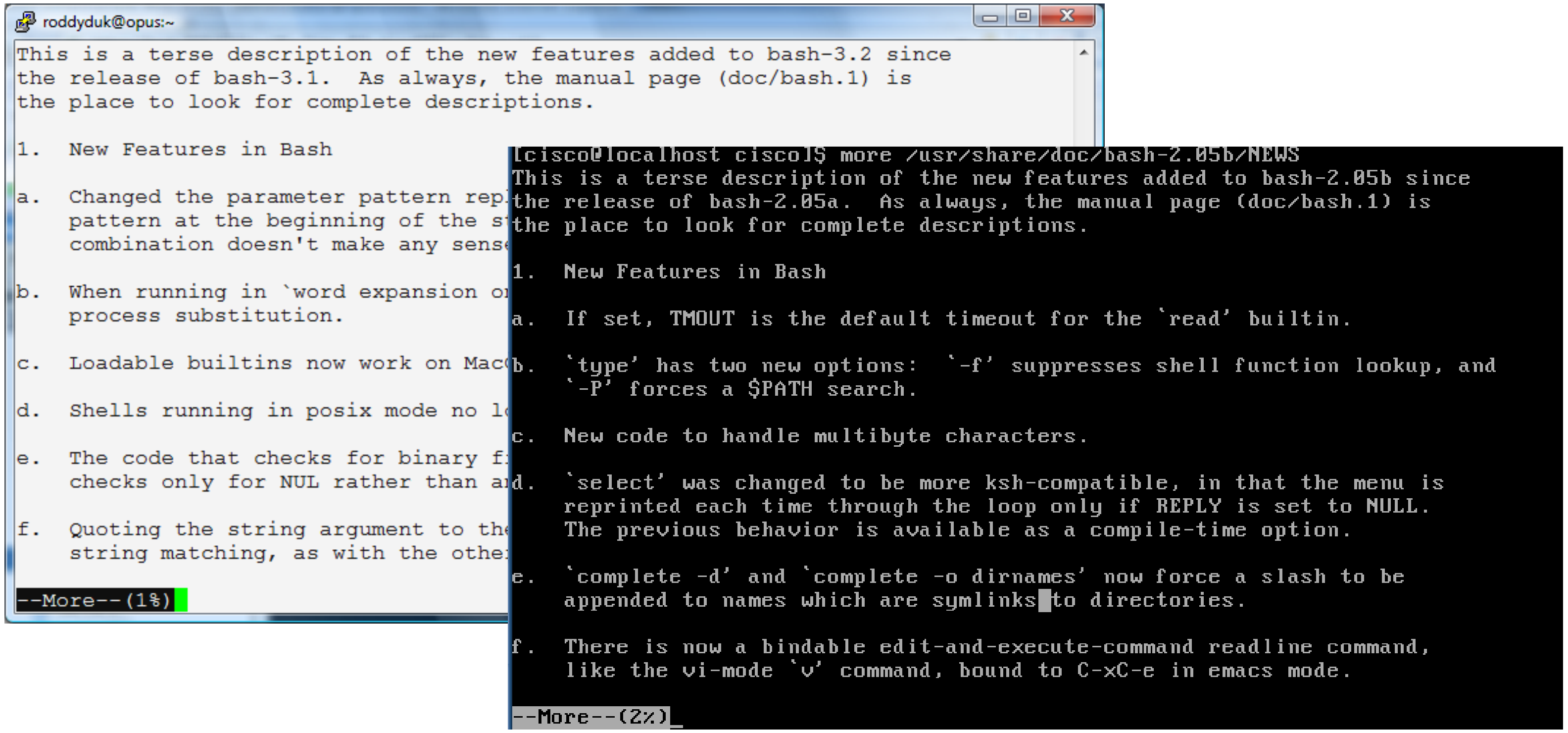
And virtual terminals have no scroll bars!



```
    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]$ _
```

more command
viewing long text files

- Use the **more** command for scolling 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 rep
     pattern at the beginning of the s
     combination doesn't make any sense

b.   When running in `word expansion o
     process substitution.

c.   Loadable builtins now work on Mac

d.   Shells running in posix mode no l

e.   The code that checks for binary f
     checks only for NUL rather than a

f.   Quoting the string argument to the
     string matching, as with the othe

--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, TMOUT 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

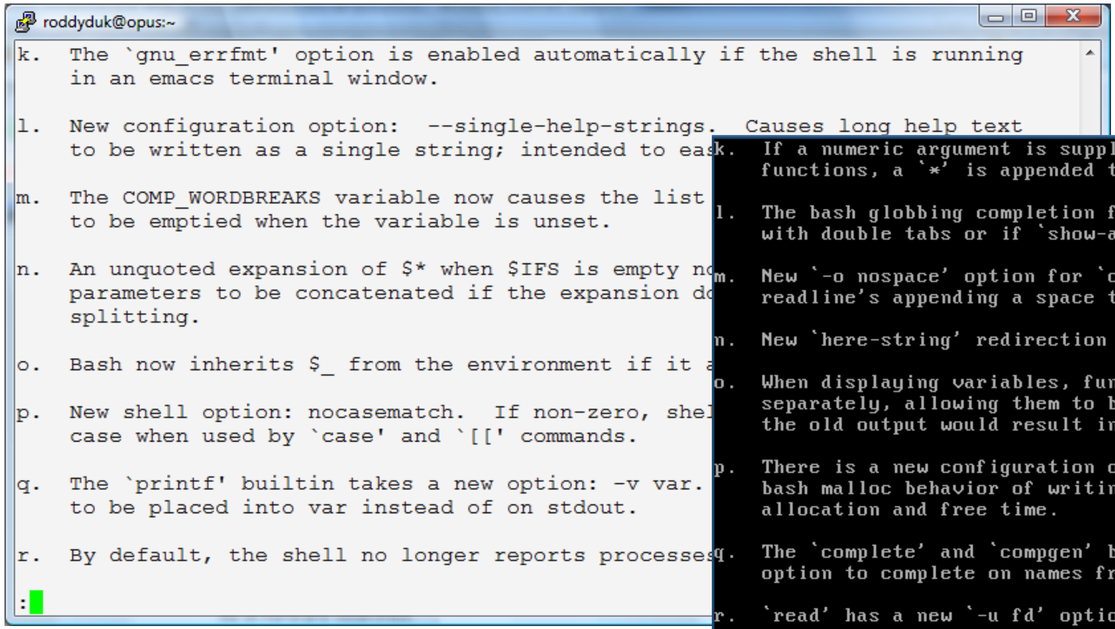
- Use the **more** command can take multiple arguments

```
/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.

                                                Alan Sherman
/home/cis90/simben $
```

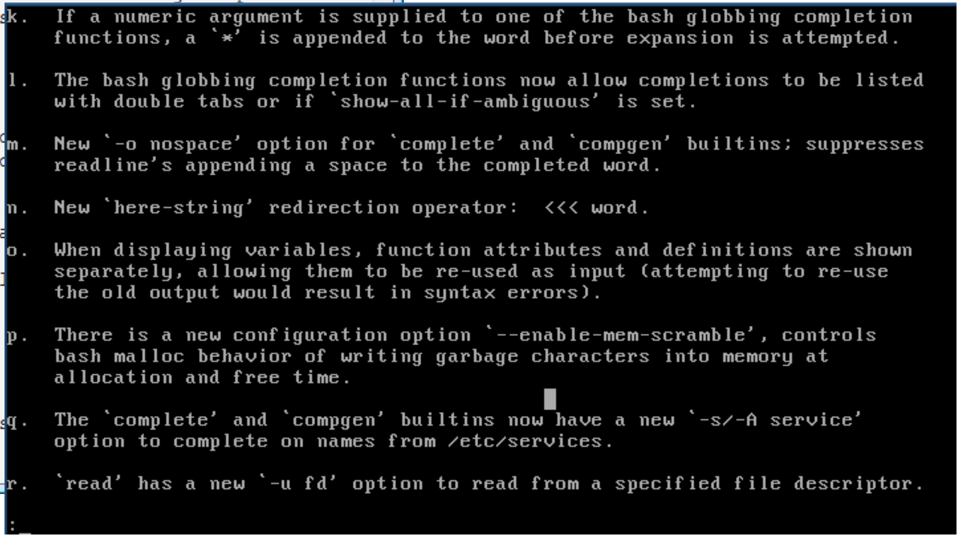
Notice with multiple files as arguments, each file has a header to separate it from the other files

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`**



```
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 ea

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

n.  An unquoted expansion of $* when $IFS is empty no
    parameters to be concatenated if the expansion do
    splitting.

o.  Bash now inherits $_ from the environment if it a

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

q.  The `printf' builtin takes a new option: -v var.
    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.

:
```

"less is more" ☺

*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.

A single argument, proposal1, is passed from the shell to the head command to process

```
/home/cis90/simben $ head 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 modifaiing vowlz and
/home/cis90/simben $
```

One option, –n 3, and a single argument, proposal1, is passed from the shell to the head command to process

```
/home/cis90/simben $ head -n 3 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
==> mission <==
                        Mission * Purpose * Values



==> letter <==
Hello Mother!  Hello Father!



==> spellk <==
Spell Check



==> 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
/home/cis90/simben $
```

One option , -n2, and multiple arguments are passed to the head command to process

Note the small banners containing the filename which separates each file

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.

A single argument, mission, is passed from the shell to the tail command to process

```
/home/cis90/simben $ tail mission
        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 $
```

One option, –n3, and a single argument, mission, is passed from the shell to the tail command to process

```
/home/cis90/simben $ tail -n3 mission
        teaching methods for diverse learning styles, and involve and
        enrich our community.

/home/cis90/simben $
```

wc command

count words and lines in a text file

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

#bytes
#words
#lines

```
/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 2 lines of mission, letter, spellk and log files

  head -n2 mission letter spellk log

- Count the number of words in small_town

  wc –w small_town

Viewing binary files

Viewing binary files:

- xxd *do a hex dump of a binary file*
- file *useful for identifying binary/text files*

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 cannot be viewed with cat, more, less, head, tail, etc.

```
/home/cis90/simben $ cat /bin/uname
ELF04`I4(444444▒>▒>@▒( A  ▒▒HHH  P▒td▒644Q▒td/lib/ld-
linux.so.2GNU      (B▒`(*▒K▒▒G-▒▒>K▒▒▒y▒cg▒}Ti▒▒▒▒ w▒)
                             ▒▒C52L▒/9=@▒x▒H_▒^fOI ▒
G▒<▒▒▒▒'▒▒6?▒▒▒w▒C*Y▒▒A▒▒$),K,ʃ▒"▒),K▒▒H▒.▒▒▒▒▒▒.▒▒.▒▒▒▒▒▒▒.▒▒▒▒
▒▒.▒▒▒▒▒/▒d8/▒▒▒▒▒</▒▒▒▒▒▒/▒▒▒▒▒▒▒/▒▒▒▒▒▒▒sii▒▒▒/▒▒ii▒▒▒w▒▒~▒▒w
▒~▒▒w▒▒~▒▒w▒▒~▒▒w▒▒ii
▒▒▒)▒*+, $(,08▒ <▒
```

< snipped >

```
PuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuT
TYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYP
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

```
/home/cis90/simben $ xxd /bin/uname
0000000: 7f45 4c46 0101 0100 0000 0000 0000 0000  .ELF............
0000010: 0200 0300 0100 0000 308b 0408 3400 0000  ........0...4...
0000020: 6049 0000 0000 0000 3400 2000 0800 2800  `I......4. ...(.
0000030: 1f00 1e00 0600 0000 3400 0000 3480 0408  ........4...4...
0000040: 3480 0408 0001 0000 0001 0000 0500 0000  4...............
0000050: 0400 0000 0300 0000 3401 0000 3481 0408  ........4...4...
0000060: 3481 0408 1300 0000 1300 0000 0400 0000  4...............
0000070: 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

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
```

File Types

Some Common File Types

Column 1 of long listing	Type	How to make one
d	**Directory**	mkdir
-	**Regular** • Programs • Text • Data (binary) • Many more … *Use the **file** command to further classify regular files*	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 using **long listings**.*

Benji's home directory on Opus (CentOS)

*The -l option on the ls command produces what is called a **Long Listing***

```
/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
```

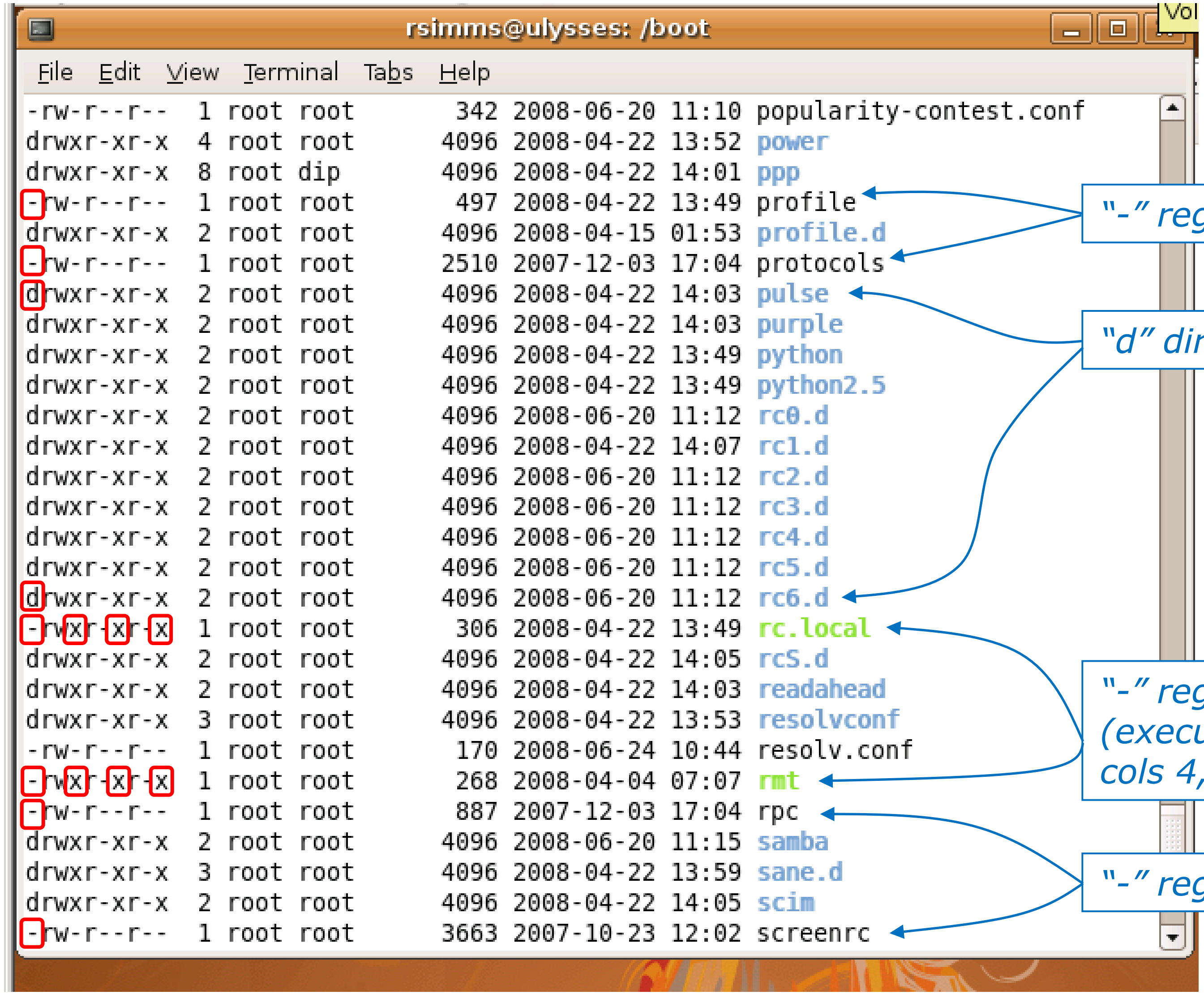
*A "d" indicates a **directory***

*A "-" indicates a **regular file***

Column 1 of long listings shows basic file types

Directory filenames also appear in blue

The /etc directory (Ubuntu)



```
-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
```

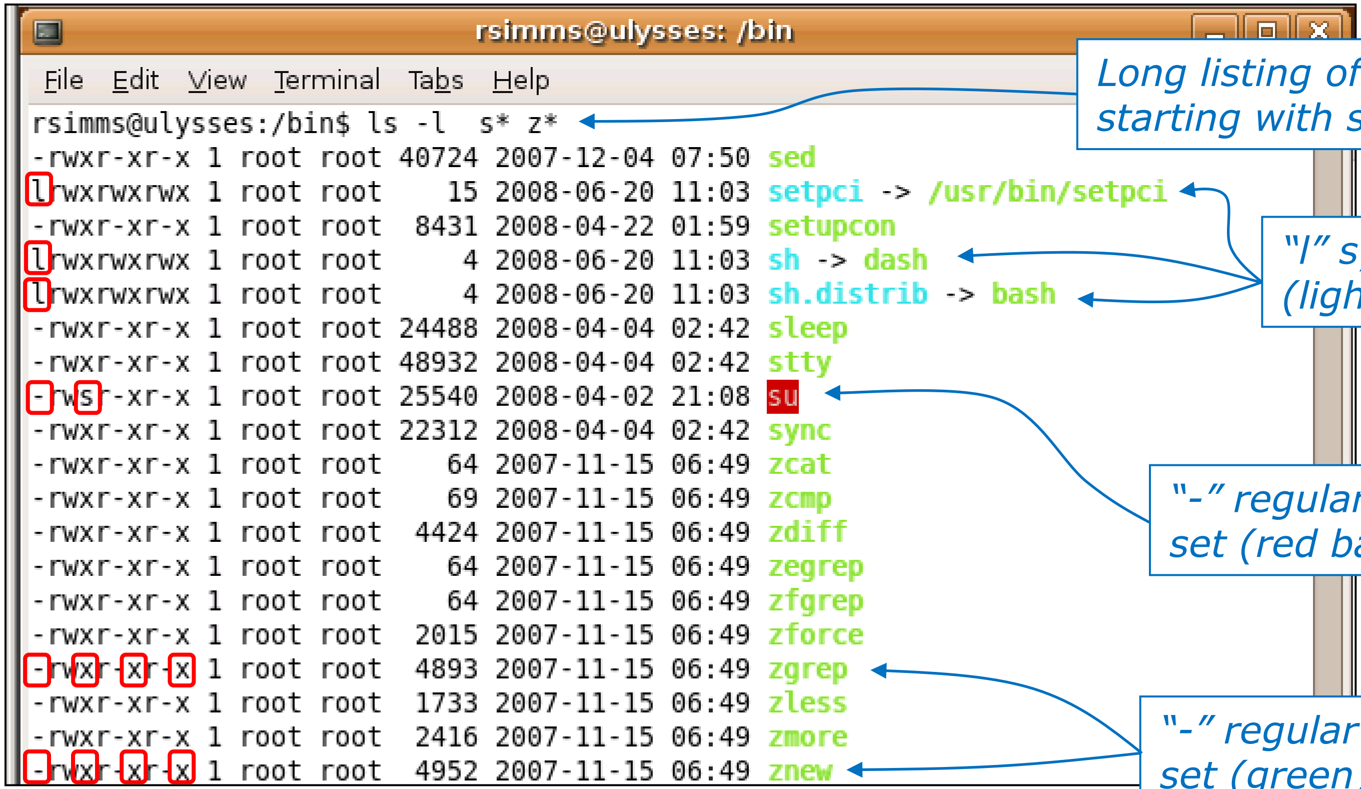
"-" regular files (black)

"d" directories (blue)

"-" regular files with x (execute) bit set (green) in cols 4,7, 10

"-" regular file (black)

A portion of the /bin directory (Ubuntu)



```
rsimms@ulysses:/bin$ ls -l  s* z*
-rwxr-xr-x 1 root root 40724 2007-12-04 07:50 sed
lrwxrwxrwx 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
lrwxrwxrwx 1 root root     4 2008-06-20 11:03 sh -> dash
lrwxrwxrwx 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
-rwsr-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
-rwxr-xr-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
-rwxr-xr-x 1 root root  4952 2007-11-15 06:49 znew
```

*Long listing of files with names starting with s or z (more on * later)*

"l" symbolic links (light blue)

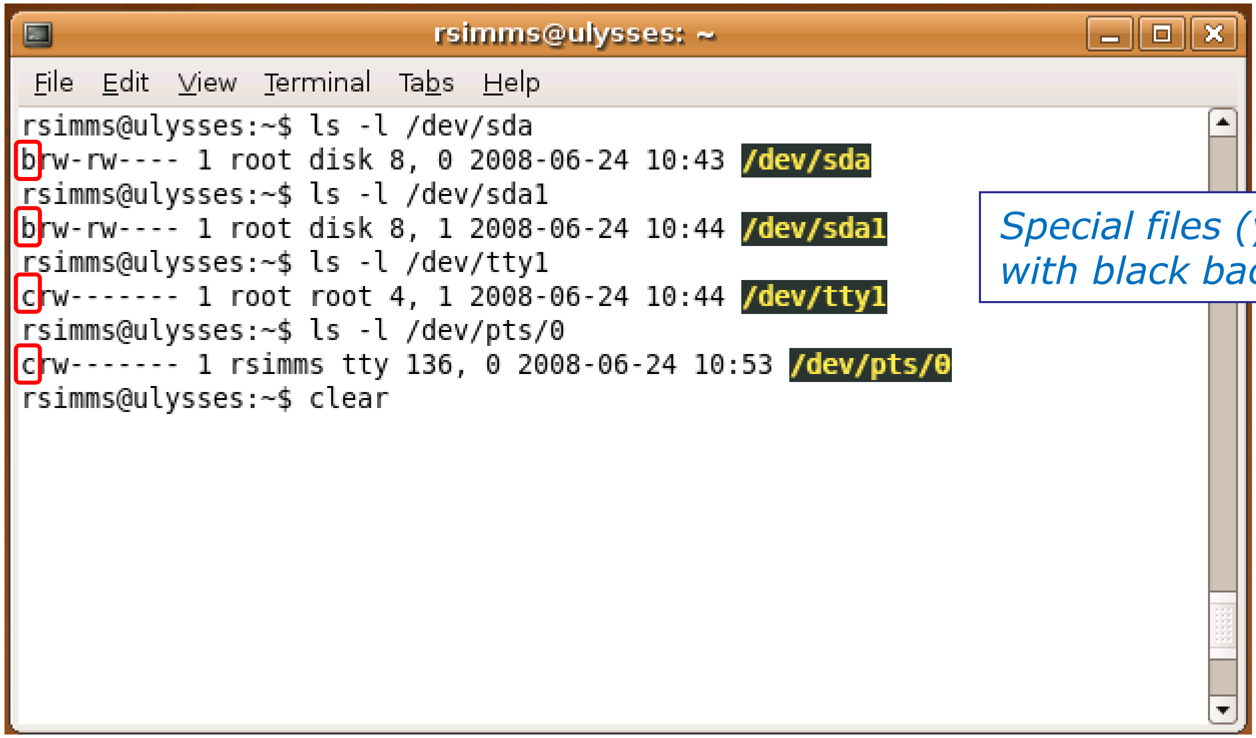
"-" regular file with setuid bit set (red background)

"-" 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



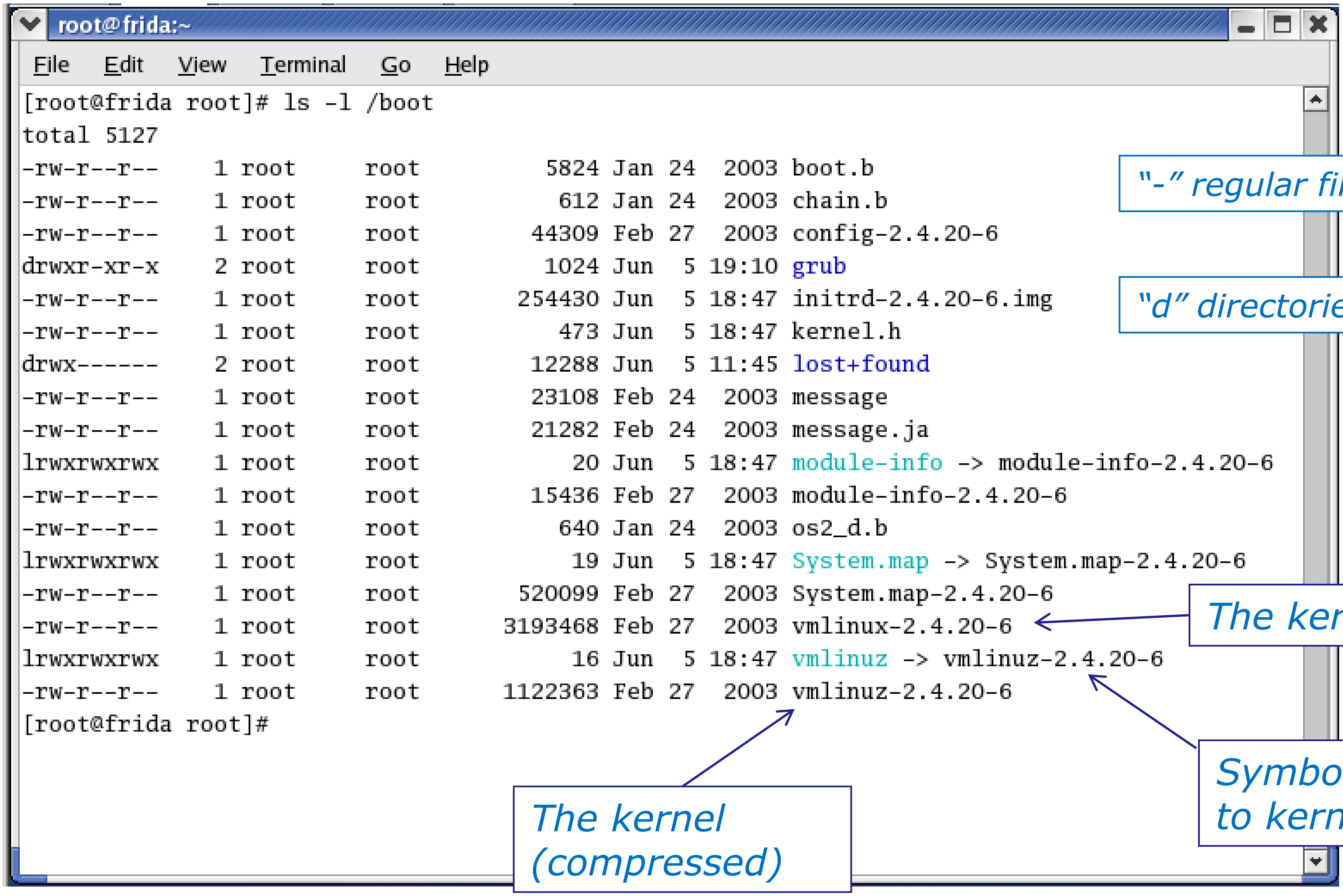
```
rsimms@ulysses:~$ ls -l /dev/sda
brw-rw---- 1 root disk 8, 0 2008-06-24 10:43 /dev/sda
rsimms@ulysses:~$ ls -l /dev/sda1
brw-rw---- 1 root disk 8, 1 2008-06-24 10:44 /dev/sda1
rsimms@ulysses:~$ ls -l /dev/tty1
crw------- 1 root root 4, 1 2008-06-24 10:44 /dev/tty1
rsimms@ulysses:~$ ls -l /dev/pts/0
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 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 vmlinux-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?
- Is *config-2.6.32-71.el6.i686* a directory or a regular file?
- Which files are executable?

Further Classifying Files

Probe a file to see what it is

- file *displays more detailed file information*

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 as to what type they are.

*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 identify the type of data in a file

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

*You can use **cat**, **more**, **head**, etc. safely on text files*

```
/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 $
```

*If it's not a text file then it is a binary file. Use **xxd** to view binary files.*

Using file command to further classify files

```
/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
```

Long listings indicate whether a file is a directory or regular file.

```
/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
```

Output from the file command provides additional information on regular files

Class Activity

Classify the following files in your home directory:

- mbox
- letter
- Poems
- timecal

```
/home/cis90/simben $ file mbox letter Poems timecal
mbox:     ASCII mail text, with very long lines
letter:   ASCII English text
Poems:    directory
timecal: shell archive or script for antique kernel text
/home/cis90/simben $
```

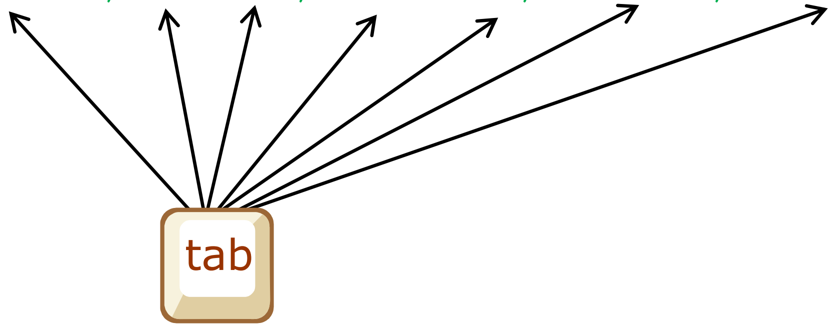
Shell tips

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
```

↑ home 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 $
```

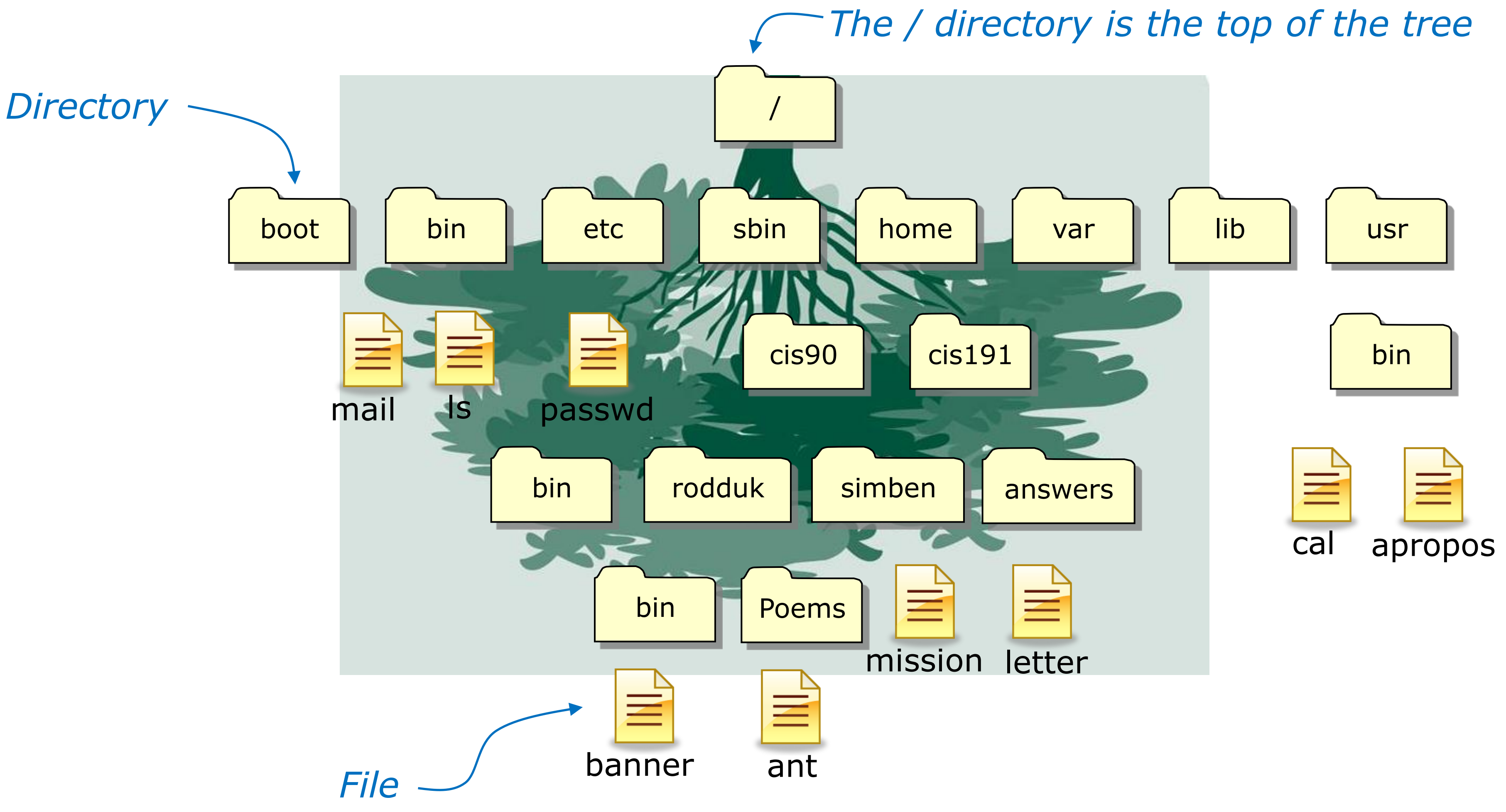
The UNIX Directory Hierarchy

UNIX File Tree

/ = root of the tree

UNIX File Tree

/ = root of the tree

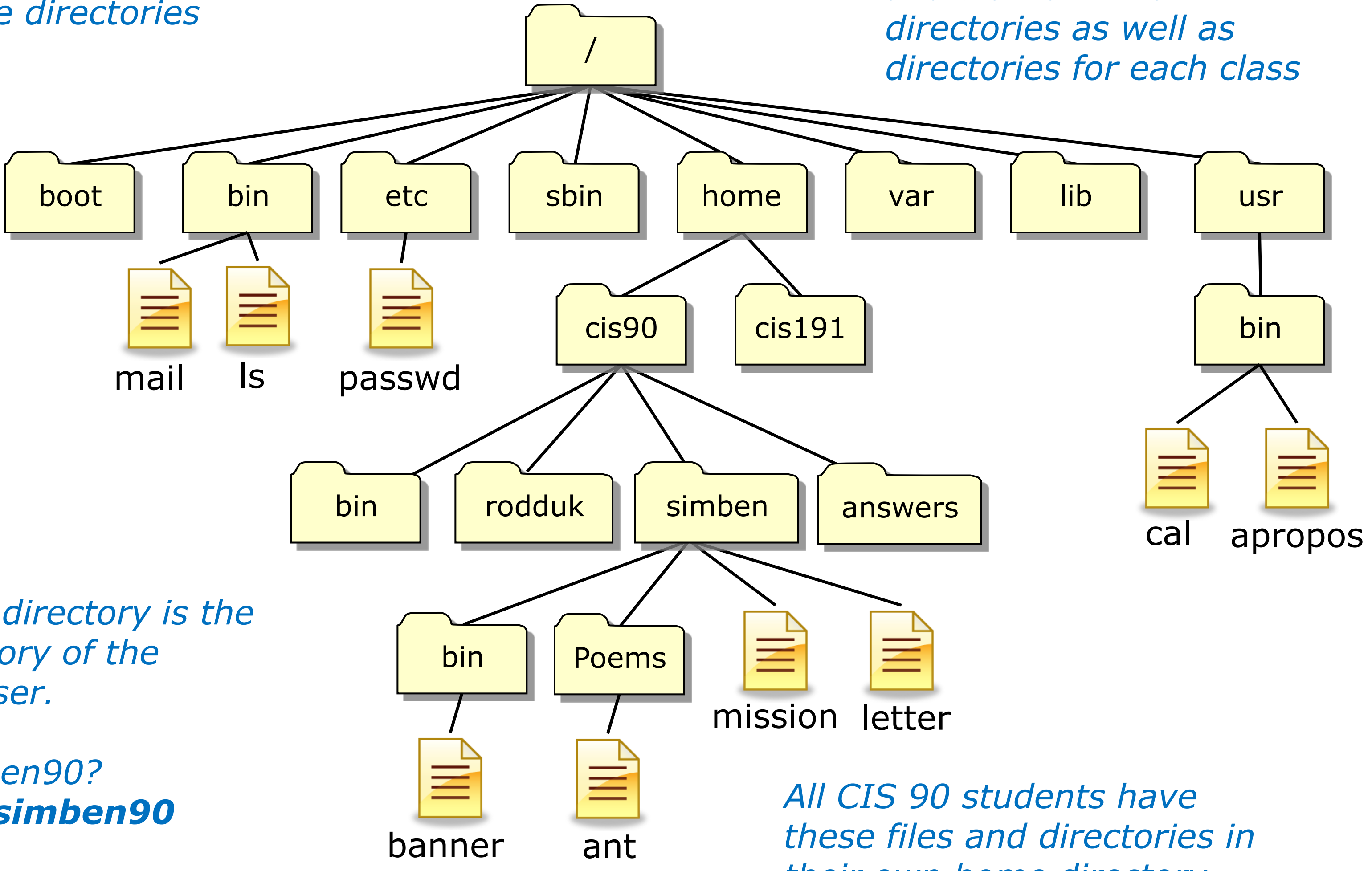


UNIX File Tree

/ = root of the tree

Directories may contain files or more directories

On Opus, the directory named "home" has faculty and staff user home directories as well as directories for each class

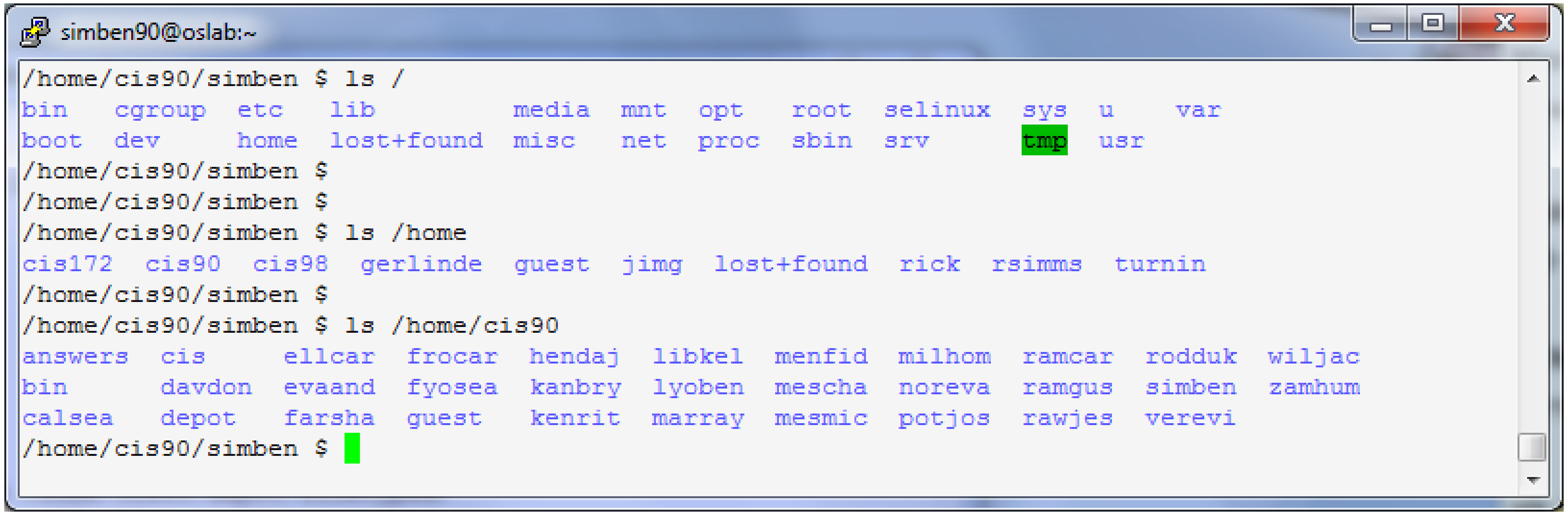


The simben directory is the home directory of the simben90 user.

Who is simben90?
*Use **finger simben90***

All CIS 90 students have these files and directories in their own home directory

A portion of the Opus file tree



simben90@oslab:~

```
/home/cis90/simben $ ls /
bin    cgroup  etc   lib         media  mnt  opt   root  selinux  sys  u    var
boot   dev     home  lost+found  misc   net  proc  sbin  srv      tmp  usr
/home/cis90/simben $
/home/cis90/simben $
/home/cis90/simben $ ls /home
cis172  cis90  cis98  gerlinde  guest  jimg  lost+found  rick  rsimms  turnin
/home/cis90/simben $
/home/cis90/simben $ ls /home/cis90
answers  cis     ellcar  frocar  hendaj  libkel  menfid  milhom  ramcar  rodduk  wiljac
bin      davdon  evaand  fyosea  kanbry  lyoben  mescha  noreva  ramgus  simben  zamhum
calsea   depot   farsha  guest   kenrit  marray  mesmic  potjos  rawjes  verevi
/home/cis90/simben $
```

1) List the top level **/** directory

   Locate ***/etc***, ***/home***, ***/bin***, ***/sbin*** *and* ***/usr*** directories in the output

2) List the contents of the directory named ***/home***

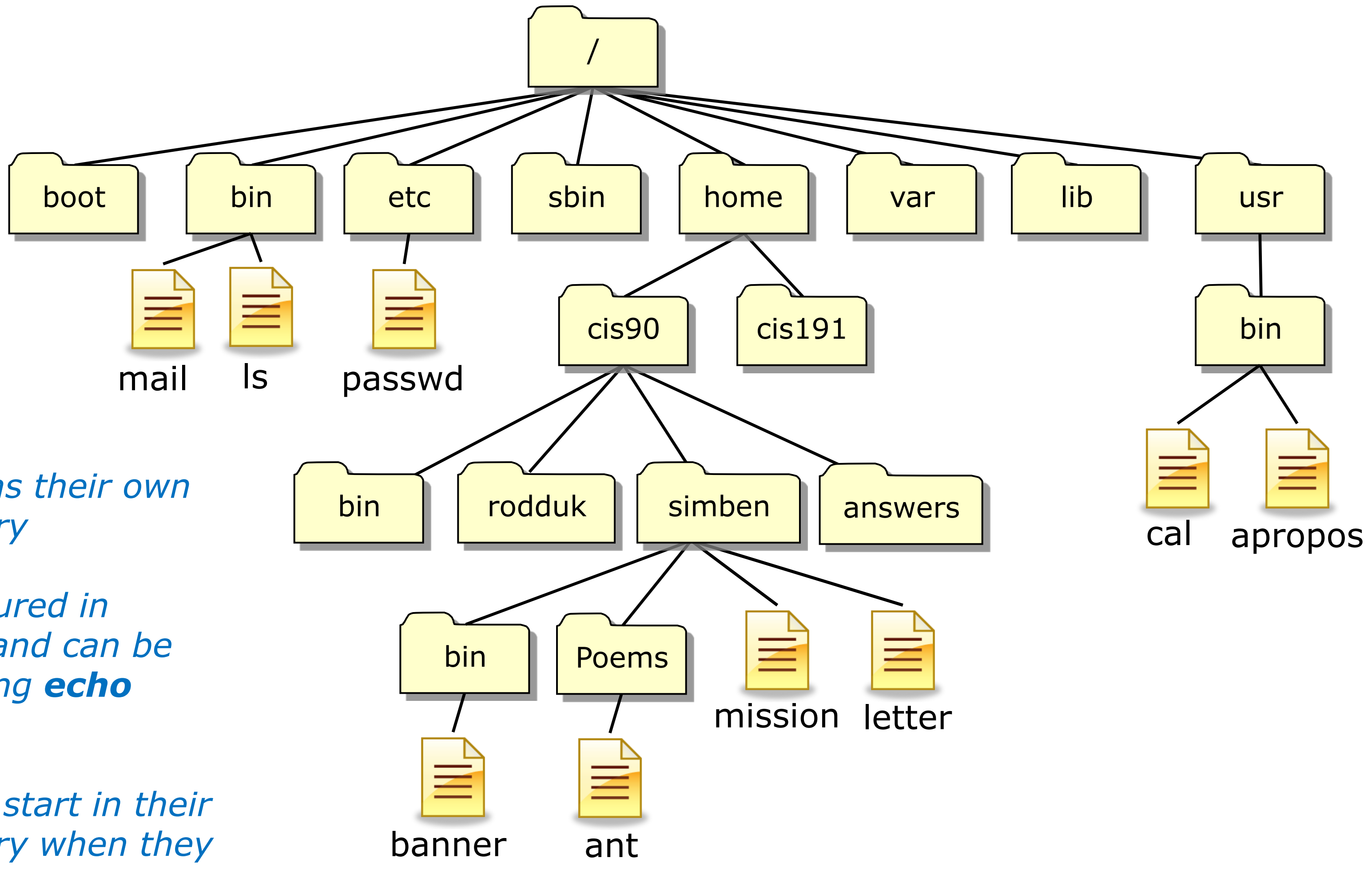
   Locate the instructors home directory in the output

3) List the contents of the directory named ***/home/cis90***

   Locate your own home directory in the output

UNIX File Tree

/ = root of the tree

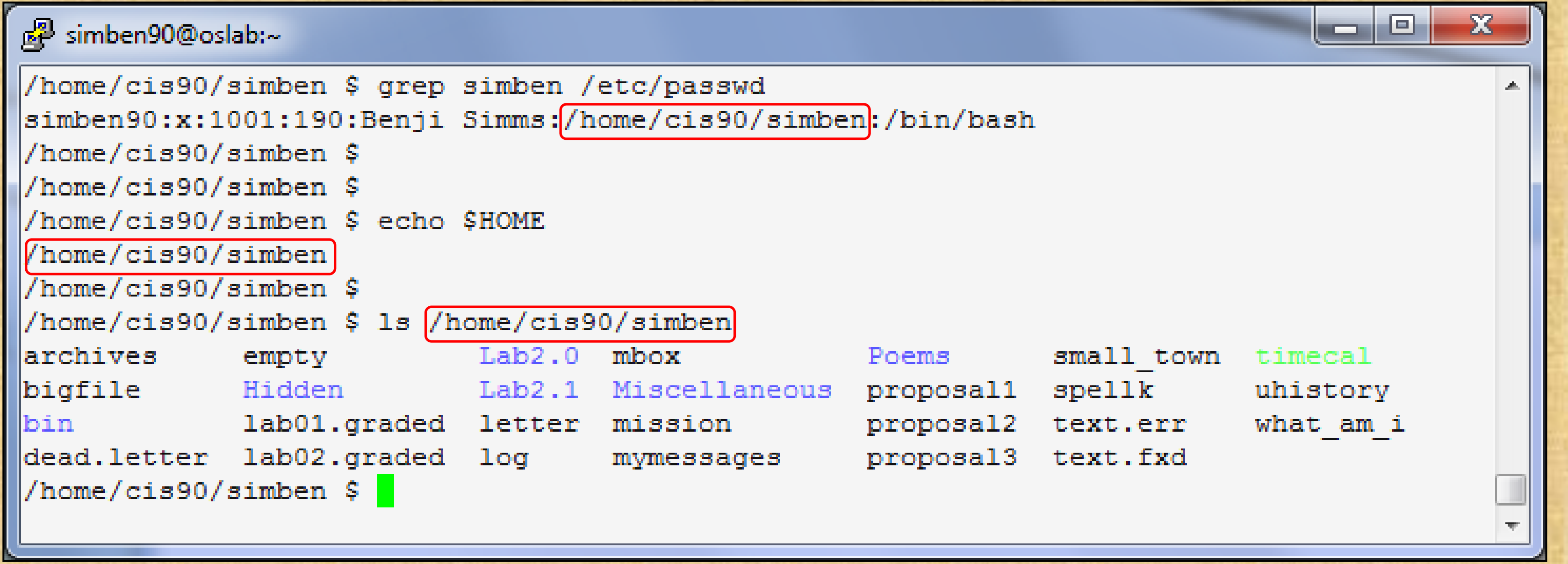


Every user has their own home directory

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

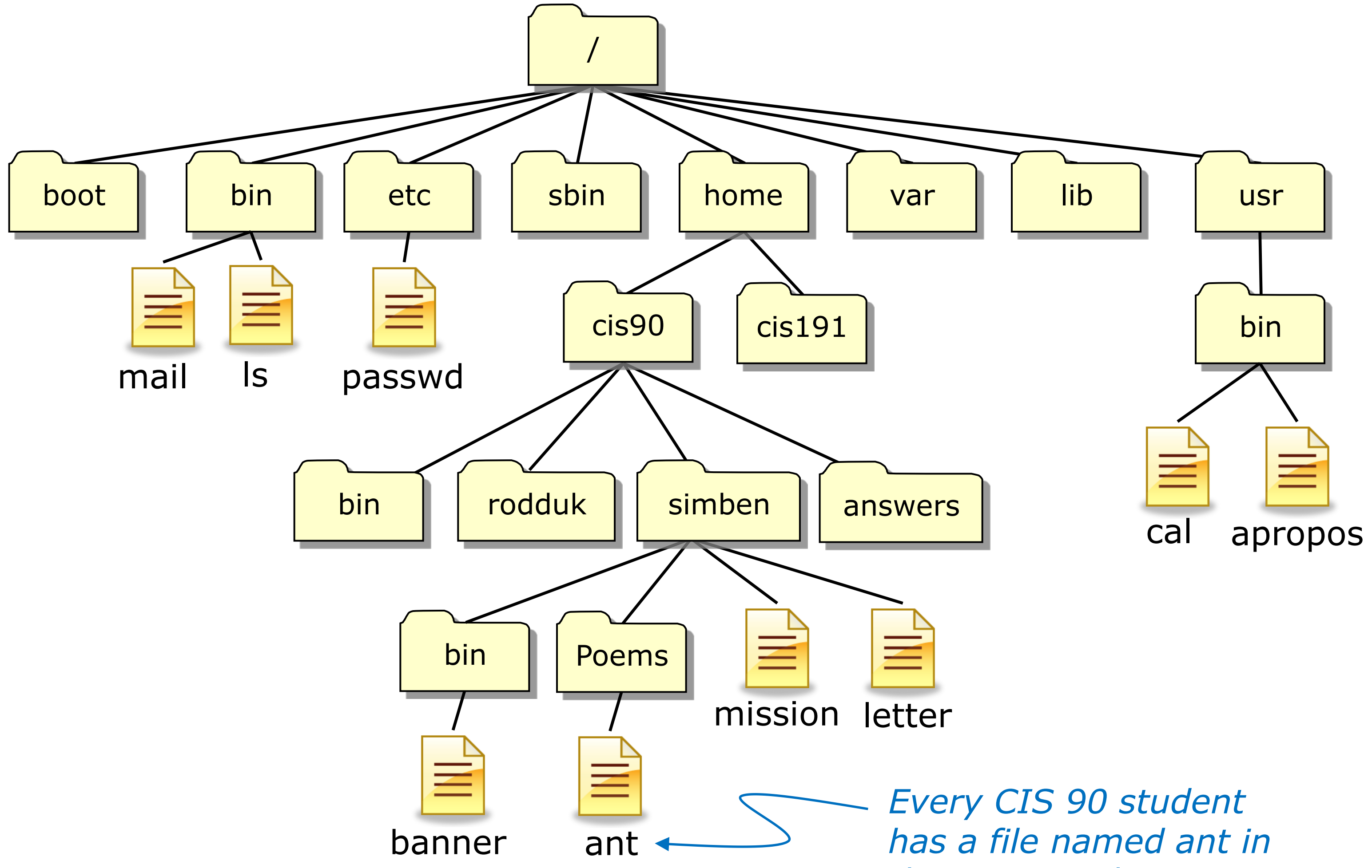


```
/home/cis90/simben $ grep simben /etc/passwd
simben90:x:1001:190:Benji Simms:/home/cis90/simben:/bin/bash
/home/cis90/simben $
/home/cis90/simben $
/home/cis90/simben $ echo $HOME
/home/cis90/simben
/home/cis90/simben $
/home/cis90/simben $ ls /home/cis90/simben
archives      empty         Lab2.0   mbox           Poems       small_town  timecal
bigfile       Hidden        Lab2.1   Miscellaneous  proposal1   spellk      uhistory
bin           lab01.graded  letter   mission        proposal2   text.err    what_am_i
dead.letter   lab02.graded  log      mymessages     proposal3   text.fxd
/home/cis90/simben $
```

1) Find your entry in /etc/passwd and locate your home directory
2) Show the contents of the HOME variable
3) List the contents of your home directory

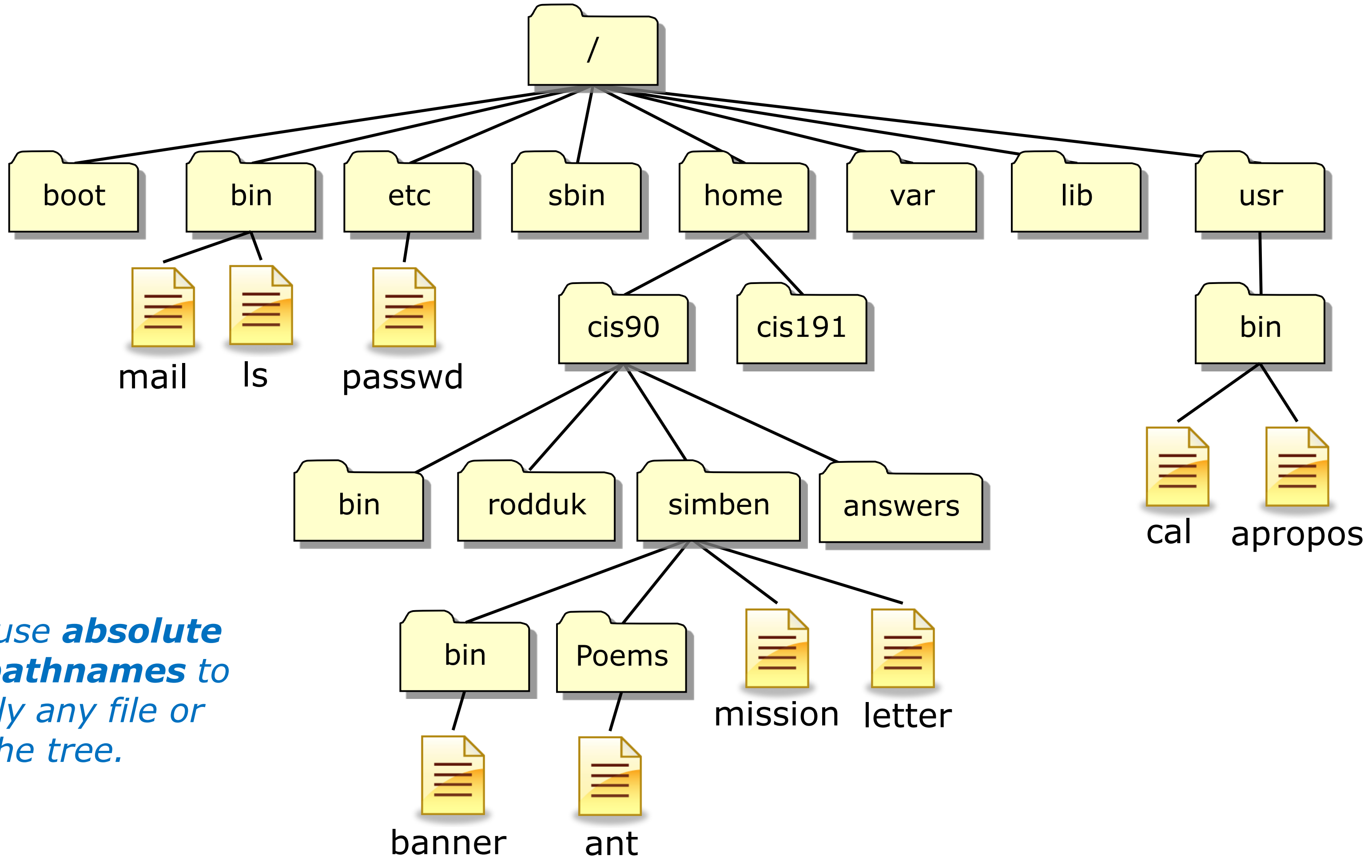
The need for pathnames

There may be multiple files in the file tree with the same name. Question: How can we specify exactly any file or directory in the file tree?



Every CIS 90 student has a file named ant in their Poems directory

The need for pathnames



Answer: We use ***absolute*** *or* ***relative pathnames*** *to specify exactly any file or directory in the tree.*

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 to all commands that deal with files and directories.

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)*

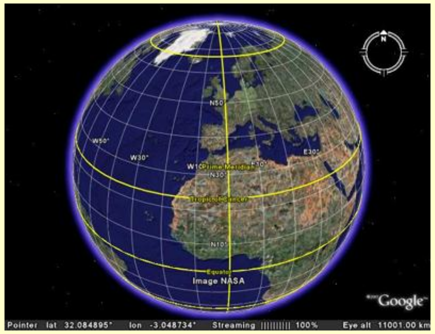
/home/cis90/bin/ *(directory)*

/bin/mail *(file)*

***** Important *****
Notice all absolute pathnames start with a / (forward slash)

An analogy …

http://www.engineeringtoolbox.com/



Latitude and longitude is an example of specifying a location in an absolute fashion based on the equator and prime meridian

Aptos, CA
Latitude: 36-58'52" N
Longitude: 121-52'28" W

Absolute Pathnames

Using absolute pathnames as command arguments

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

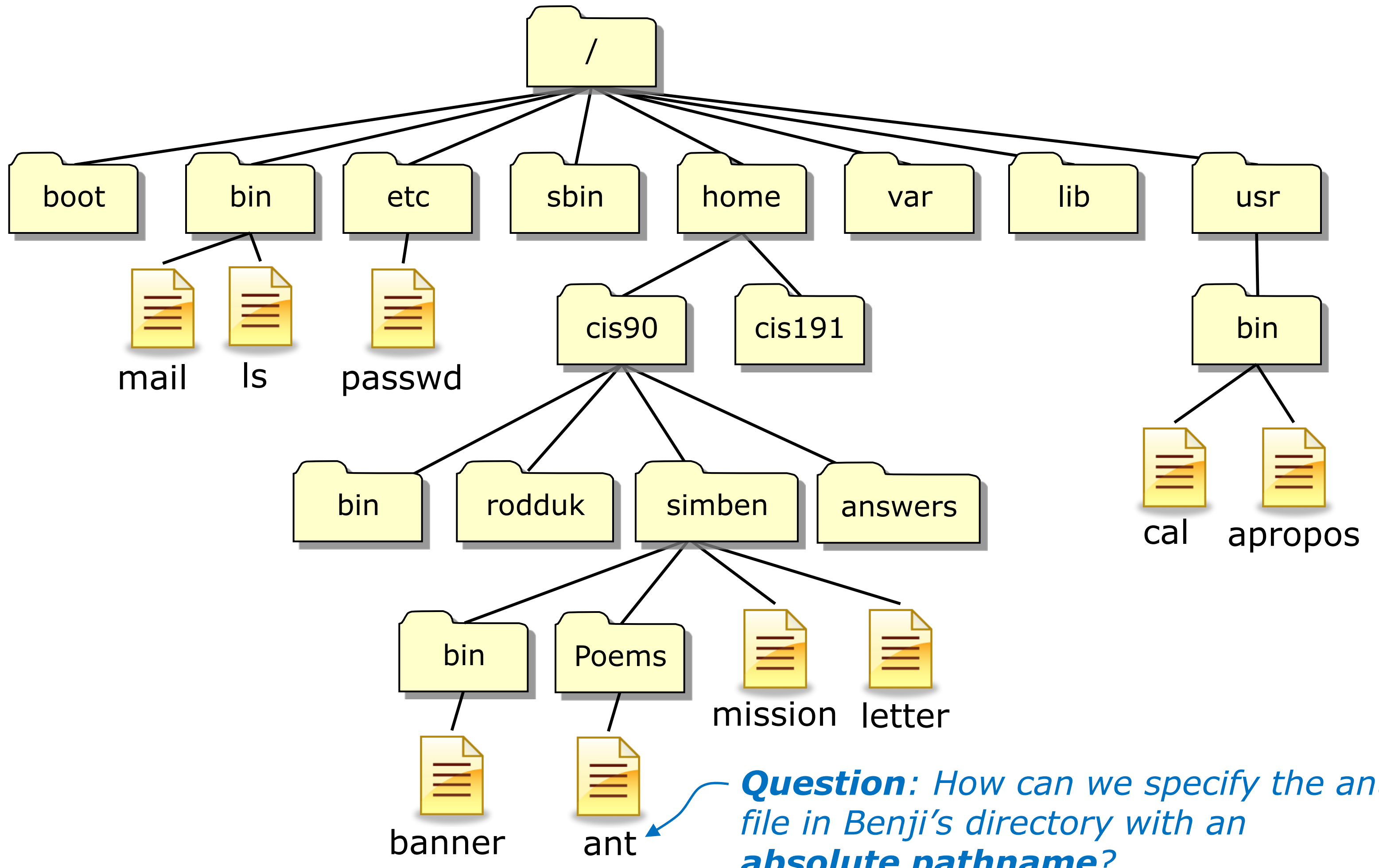
Examples of absolute pathnames used as command arguments:

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

file  /usr/bin/cal

cd  /home/cis90/simben/Poems/Shakespeare

ls  -l  /bin/mail

cp  /etc/passwd  /home/cis90/simben/misc
```

***** Important *****
Notice all absolute pathnames start with a / (forward slash)

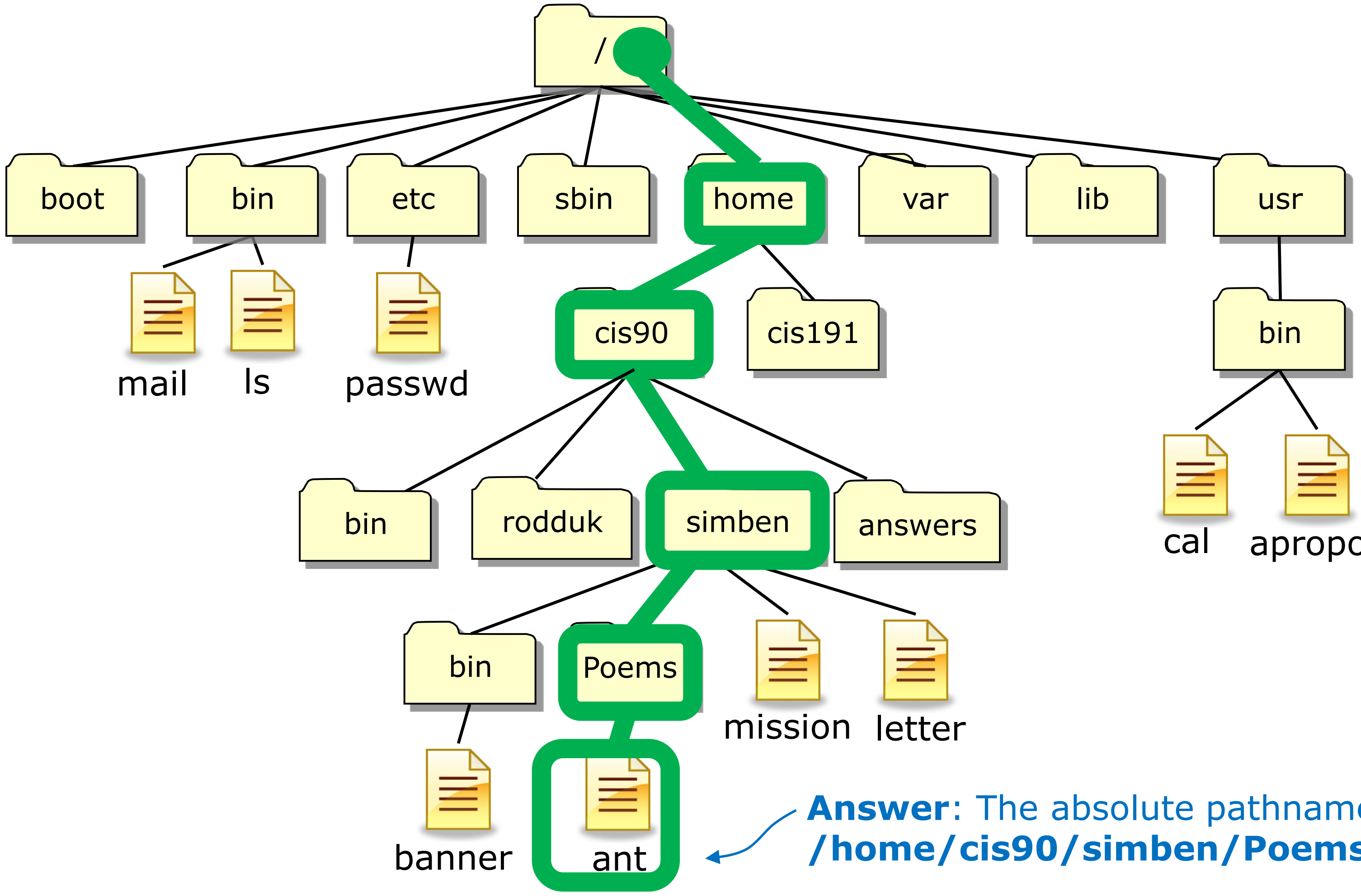
CIS 90 - Lesson 4

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



***Question**: How can we specify the ant file in Benji's directory with an **absolute pathname**?*

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

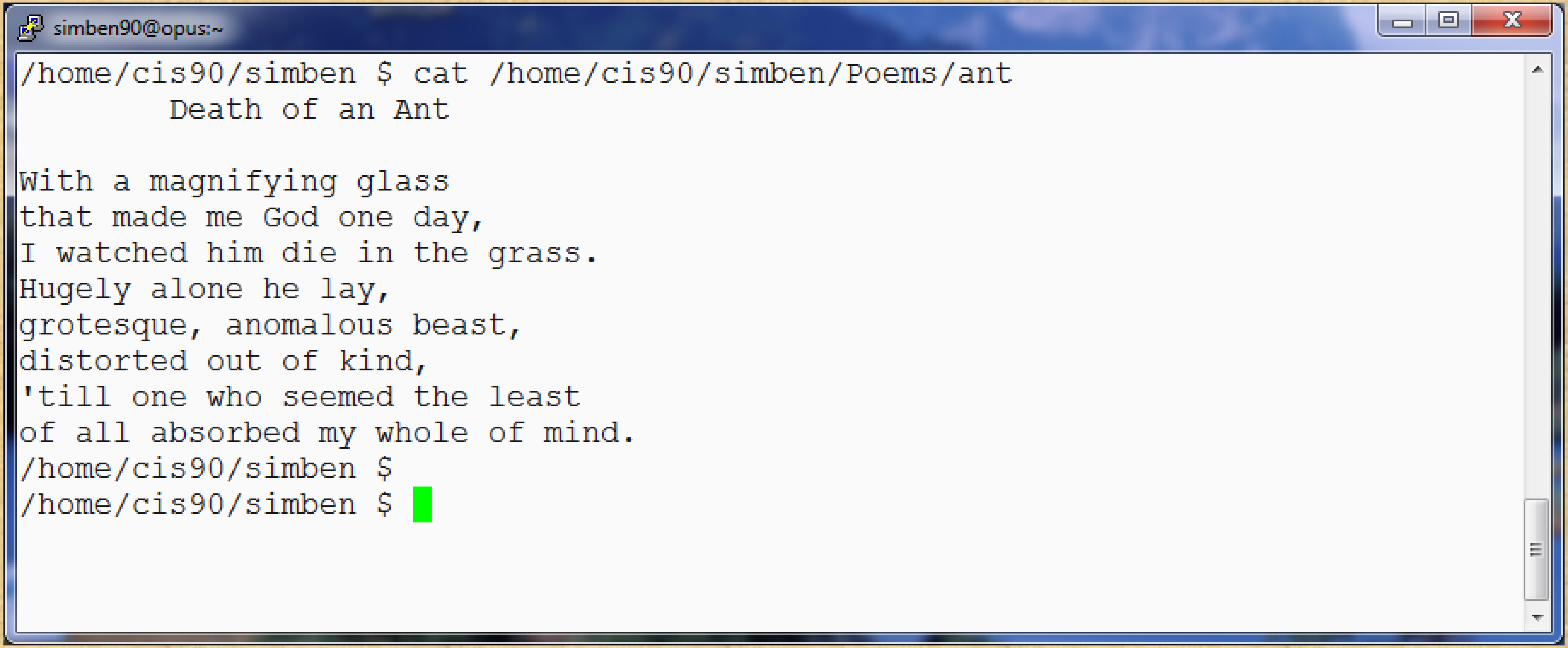


Answer: The absolute pathname is **/home/cis90/simben/Poems/ant**

/home/cis90/simben/Poems/ant

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 *Poems* directory, there you will find the *ant* file.

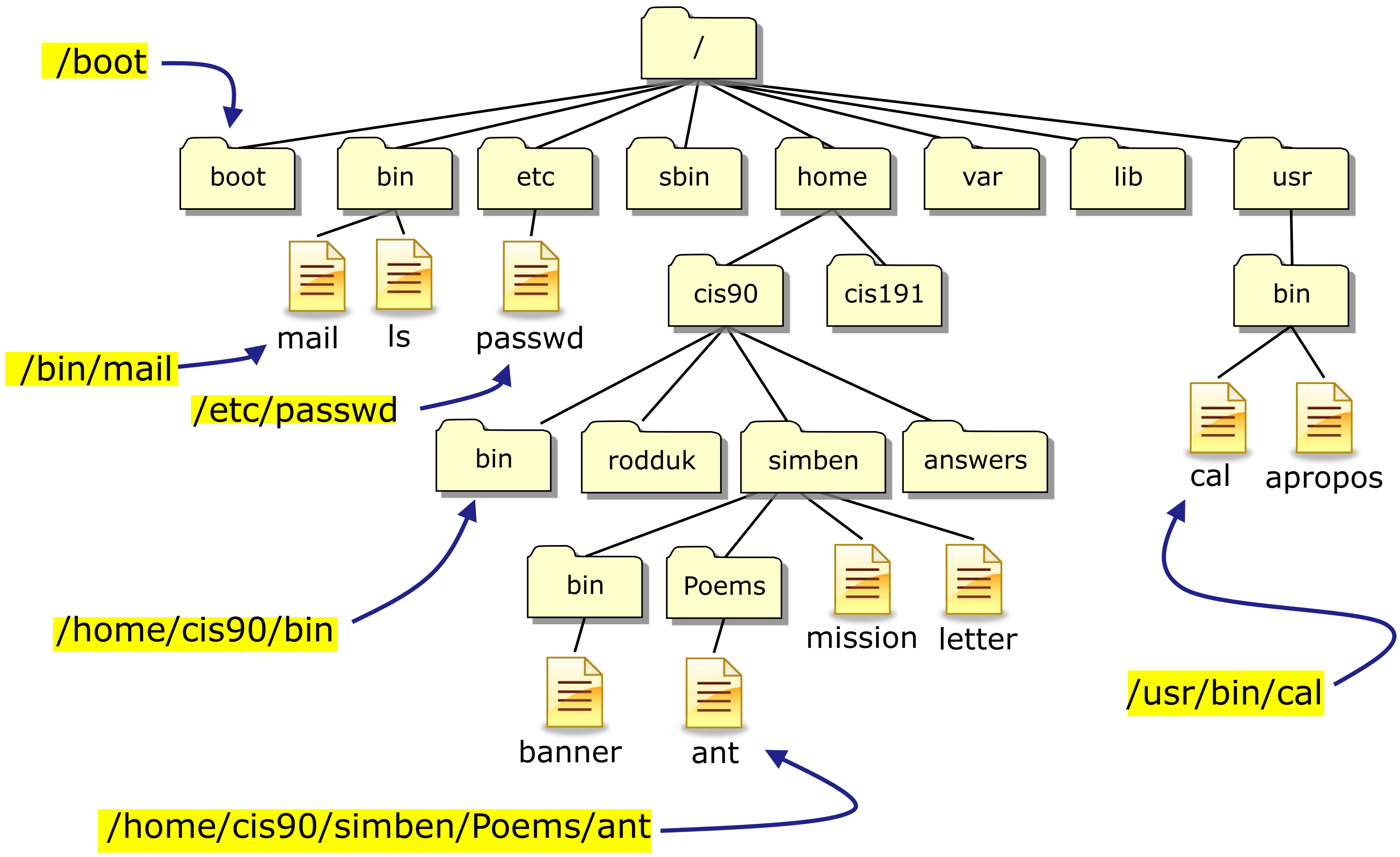
Class Activity



```
/home/cis90/simben $ cat /home/cis90/simben/Poems/ant
        Death of an Ant

With a magnifying glass
that made me God one day,
I watched him die in the grass.
Hugely alone he lay,
grotesque, anomalous beast,
distorted out of kind,
'till one who seemed the least
of all absorbed my whole of mind.
/home/cis90/simben $
/home/cis90/simben $
```

1) Cat the *ant* file belonging to simben90 using an absolute pathname
2) Cat **your** *ant* file using an absolute pathname

Some example absolute pathnames



Relative Pathnames

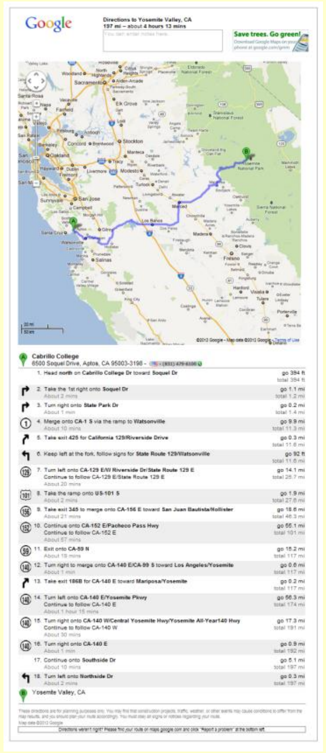
*A **relative pathname** specifies the path from your current directory to the target directory or file.*

Examples:

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

Note that relative pathnames do NOT start with a /

An analogy …



Directions to Yosemite Valley, CA
197 mi – about 4 hours 13 mins

Cabrillo College
6500 Soquel Drive, Aptos, CA 95003-3198 - [illegible]

```
1. Head north on Cabrillo College Dr toward Soquel Dr                  go 394 ft
                                                                      total 394 ft
2. Take the 1st right onto Soquel Dr                                   go 1.1 mi
   About 2 mins                                                       total 1.2 mi
3. Turn right onto State Park Dr                                       go 0.2 mi
   About 1 min                                                        total 1.4 mi
4. Merge onto CA-1 S via the ramp to Watsonville                       go 9.9 mi
   About 10 mins                                                      total 11.3 mi
5. Take exit 425 for California 129/Riverside Drive                    go 0.3 mi
                                                                      total 11.6 mi
6. Keep left at the fork, follow signs for State Route 129/Watsonville go 92 ft
                                                                      total 11.6 mi
7. Turn left onto CA-129 E/W Riverside Dr/State Route 129 E            go 14.1 mi
   Continue to follow CA-129 E/State Route 129 E                      total 25.7 mi
   About 20 mins
8. Take the ramp onto US-101 S                                         go 1.9 mi
   About 2 mins                                                       total 27.6 mi
9. Take exit 345 to merge onto CA-156 E toward San Juan Bautista/Hollister  go 18.6 mi
   About 21 mins                                                      total 46.3 mi
10. Continue onto CA-152 E/Pacheco Pass Hwy                            go 55.1 mi
    Continue to follow CA-152 E                                       total 101 mi
    About 57 mins
11. Exit onto CA-59 N                                                  go 15.2 mi
    About 18 mins                                                     total 117 mi
12. Turn right to merge onto CA-140 E/CA-99 S toward Los Angeles/Yosemite  go 0.6 mi
    About 1 min                                                       total 117 mi
13. Take exit 186B for CA-140 E toward Mariposa/Yosemite               go 0.2 mi
                                                                      total 117 mi
14. Turn left onto CA-140 E/Yosemite Pkwy                              go 56.3 mi
    Continue to follow CA-140 E                                       total 174 mi
    About 1 hour 15 mins
15. Turn right onto CA-140 W/Central Yosemite Hwy/Yosemite All-Year140 Hwy  go 17.3 mi
    Continue to follow CA-140 W                                       total 191 mi
    About 30 mins
16. Turn right onto CA-140 E                                           go 0.9 mi
    About 1 min                                                       total 192 mi
17. Continue onto Southside Dr                                         go 5.1 mi
    About 10 mins                                                     total 197 mi
18. Turn left onto Northside Dr                                        go 0.3 mi
    About 2 mins                                                      total 197 mi
```

Yosemite Valley, CA

These directions are for planning purposes only. You may find that construction projects, traffic, weather, or other events may cause conditions to differ from the map results, and you should plan your route accordingly. You must obey all signs or notices regarding your route.
Map data ©2012 Google

Directions weren't right? Please find your route on maps.google.com and click "Report a problem" at the bottom left.

Google Maps can specify a route to a destination beginning with your current location

Relative Pathnames

Using relative pathnames as command arguments

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

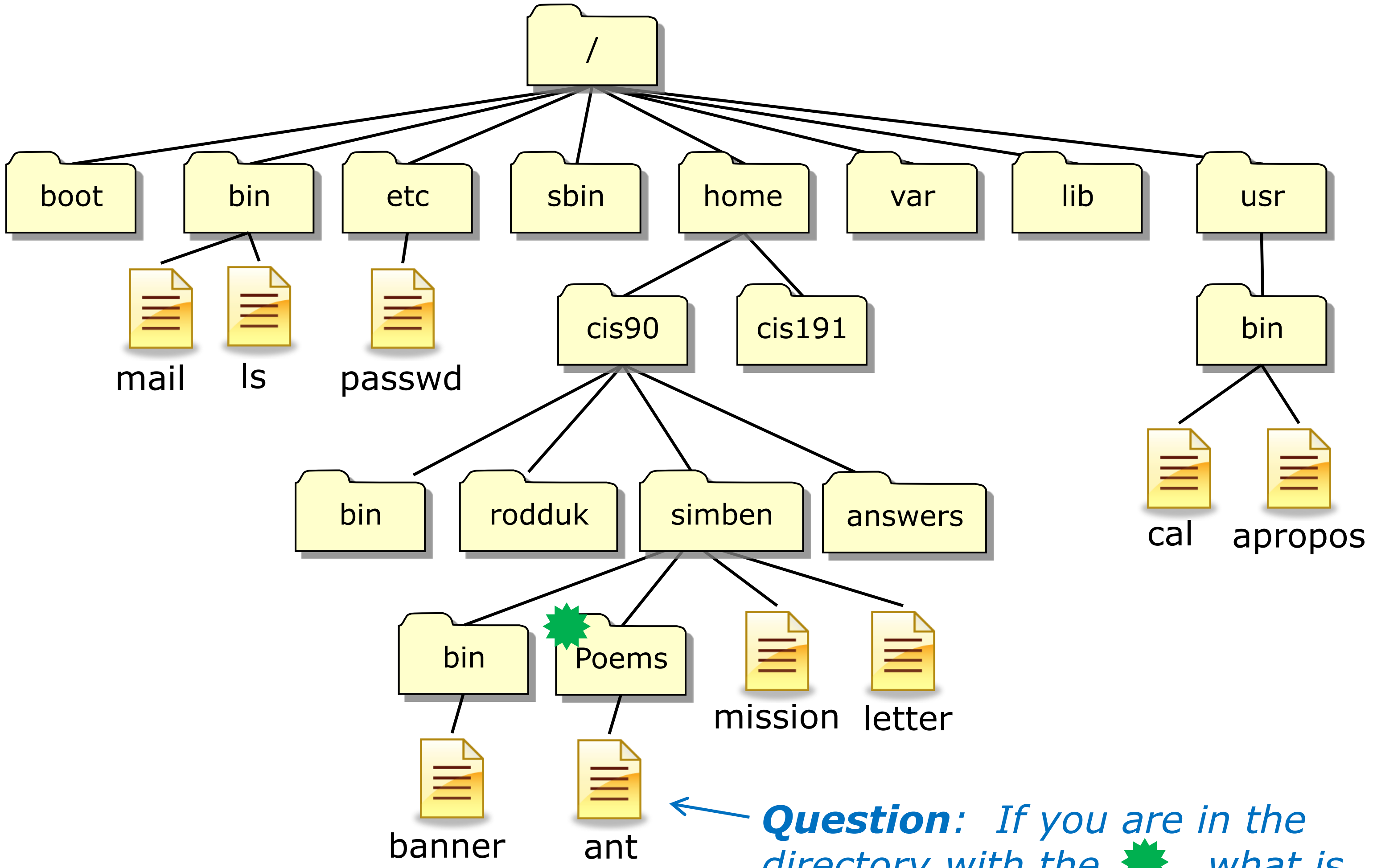
Examples of using relative pathnames as command arguments:

```
ls -l ant
file  ../../../../bin/mail
cd  Poems/Blake
ls -l  ../bin/check3
file Poems/Shakespeare/sonnet4
cd Poems/Shakespeare
```

The .. is used to represent the parent directory

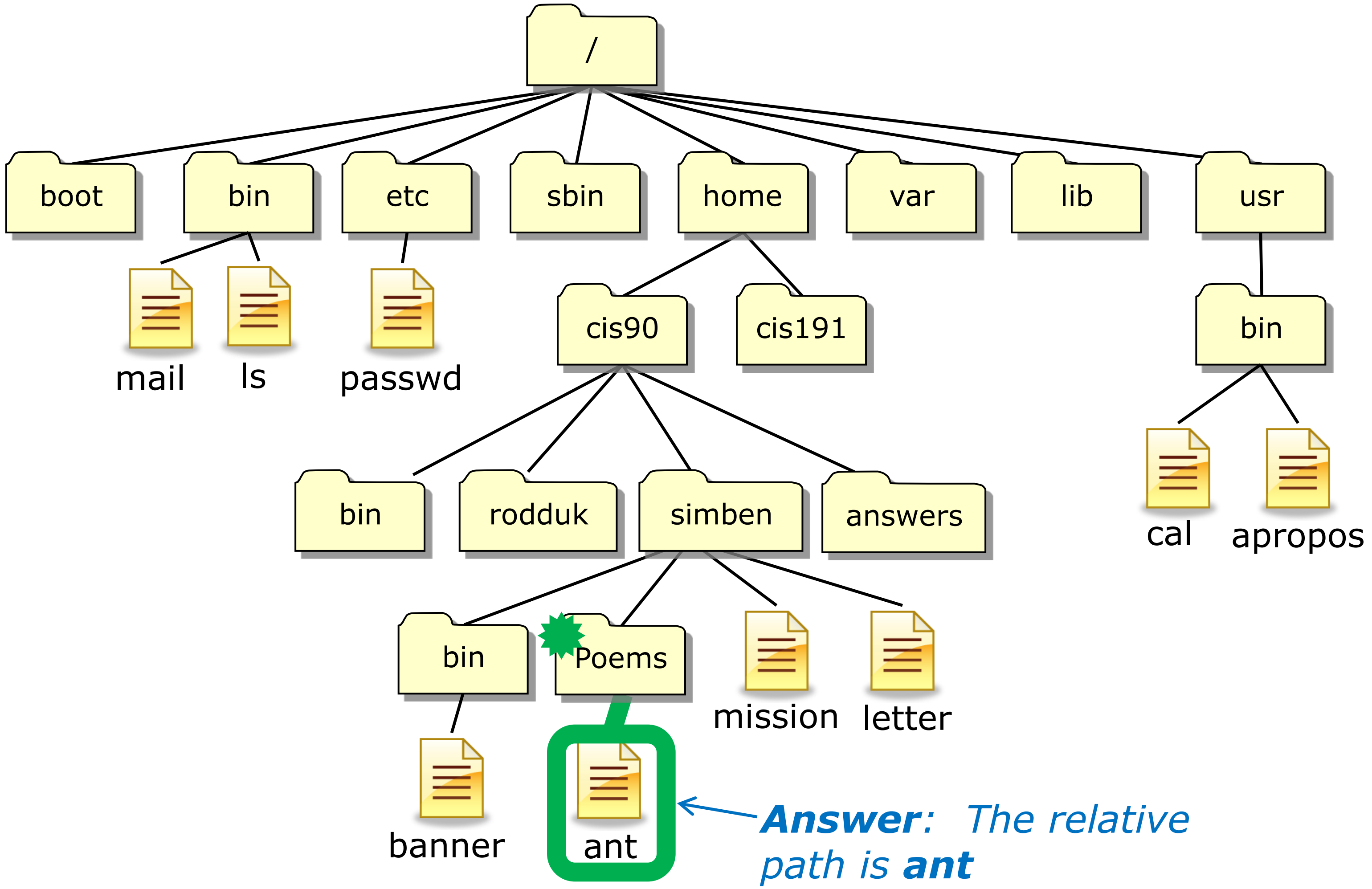
Notice that these pathnames do NOT start with the /

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



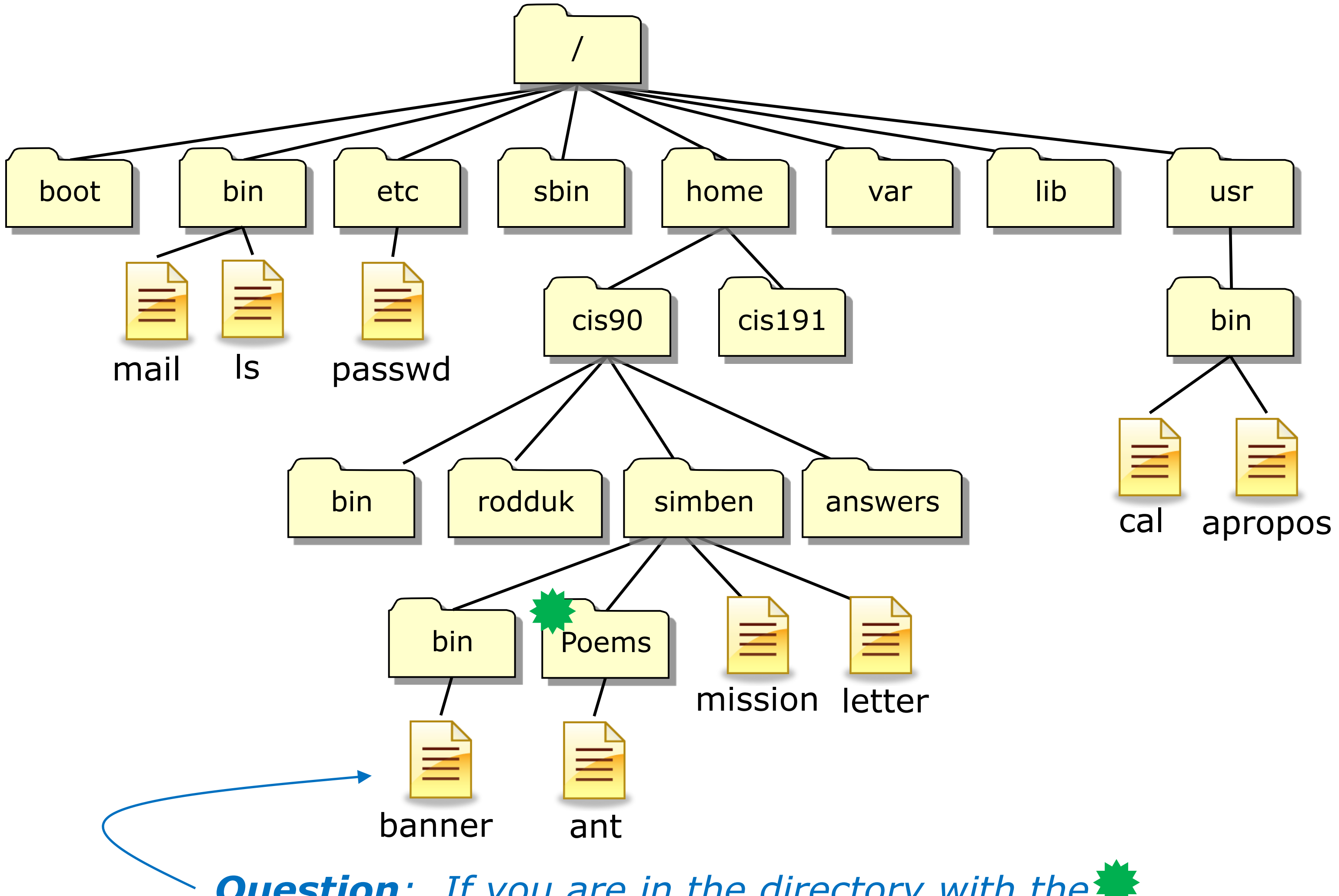
***Question**: If you are in the directory with the ✸, what is the relative path to this file?*

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



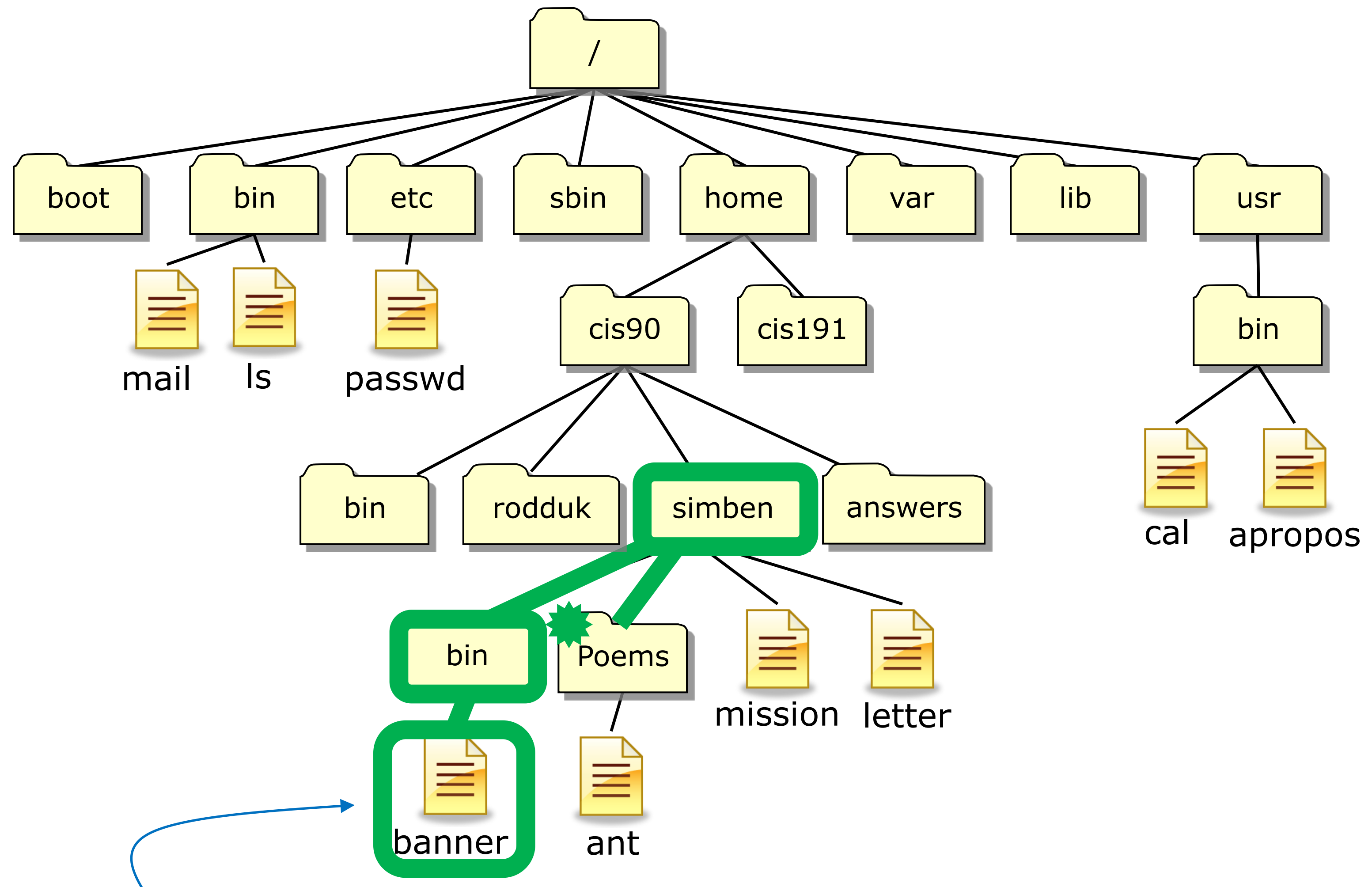
Answer**: The relative path is **ant

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



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

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

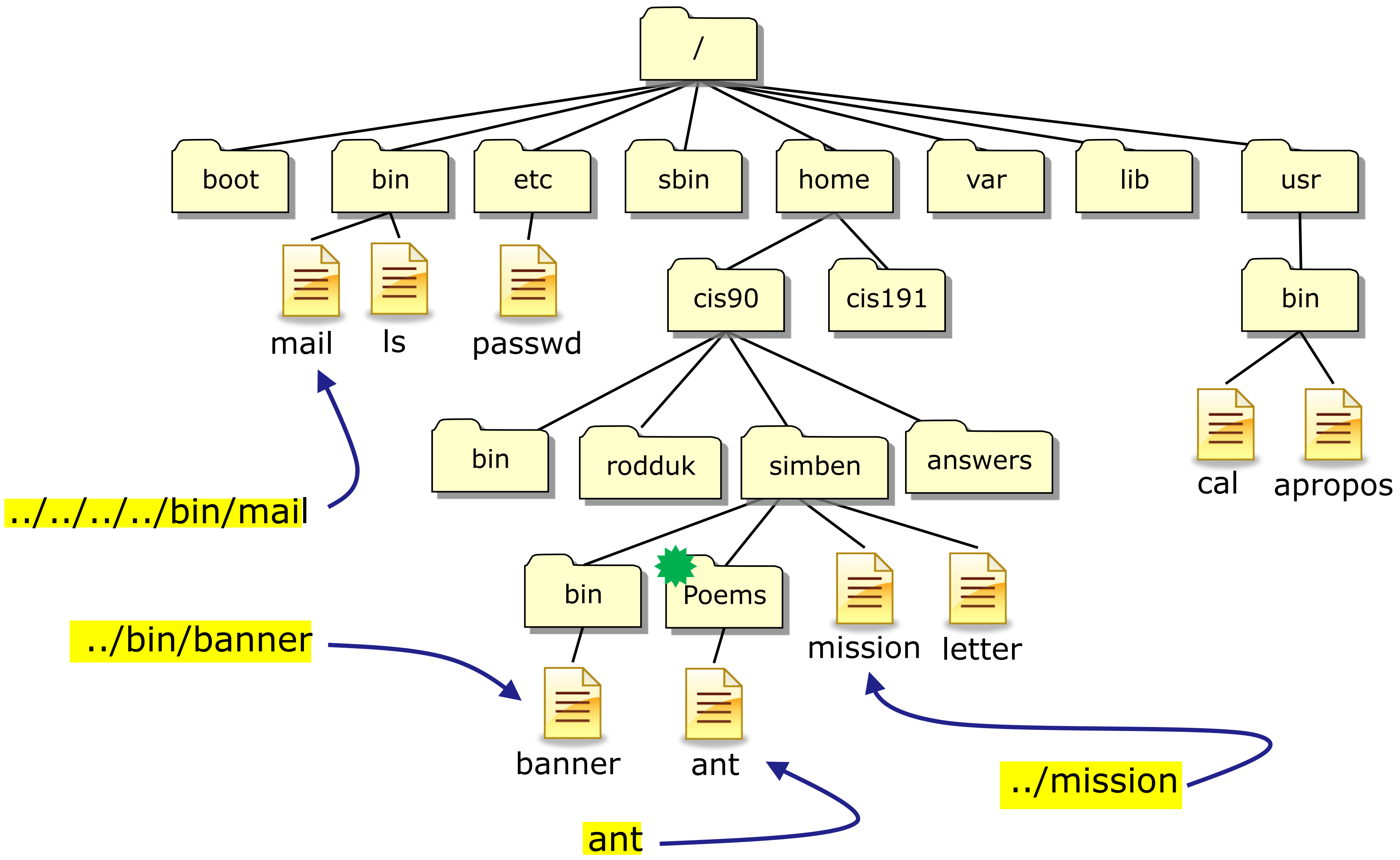


*Answer: The relative path to this file is **../bin/banner***

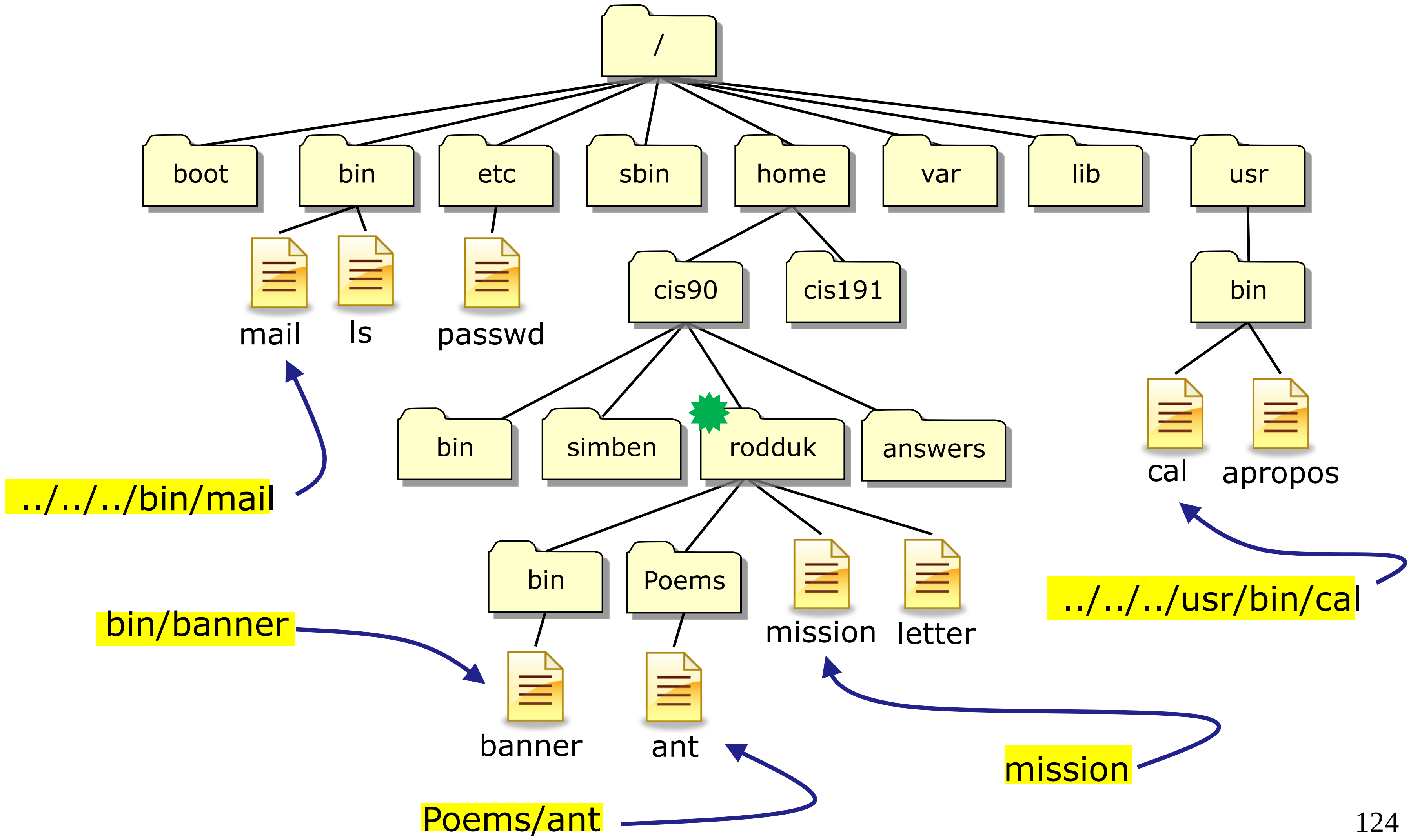
../bin/banner

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

```
ls ../../../etc/passwd
```

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

```
ls /etc/passwd
```

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

- List the letter file using a relative pathname

```
ls letter
```

Sometimes it's easier to specify a filename using a relative pathname

- List the letter file using an absolute pathname

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

use your home directory instead

Heads up on a future test question

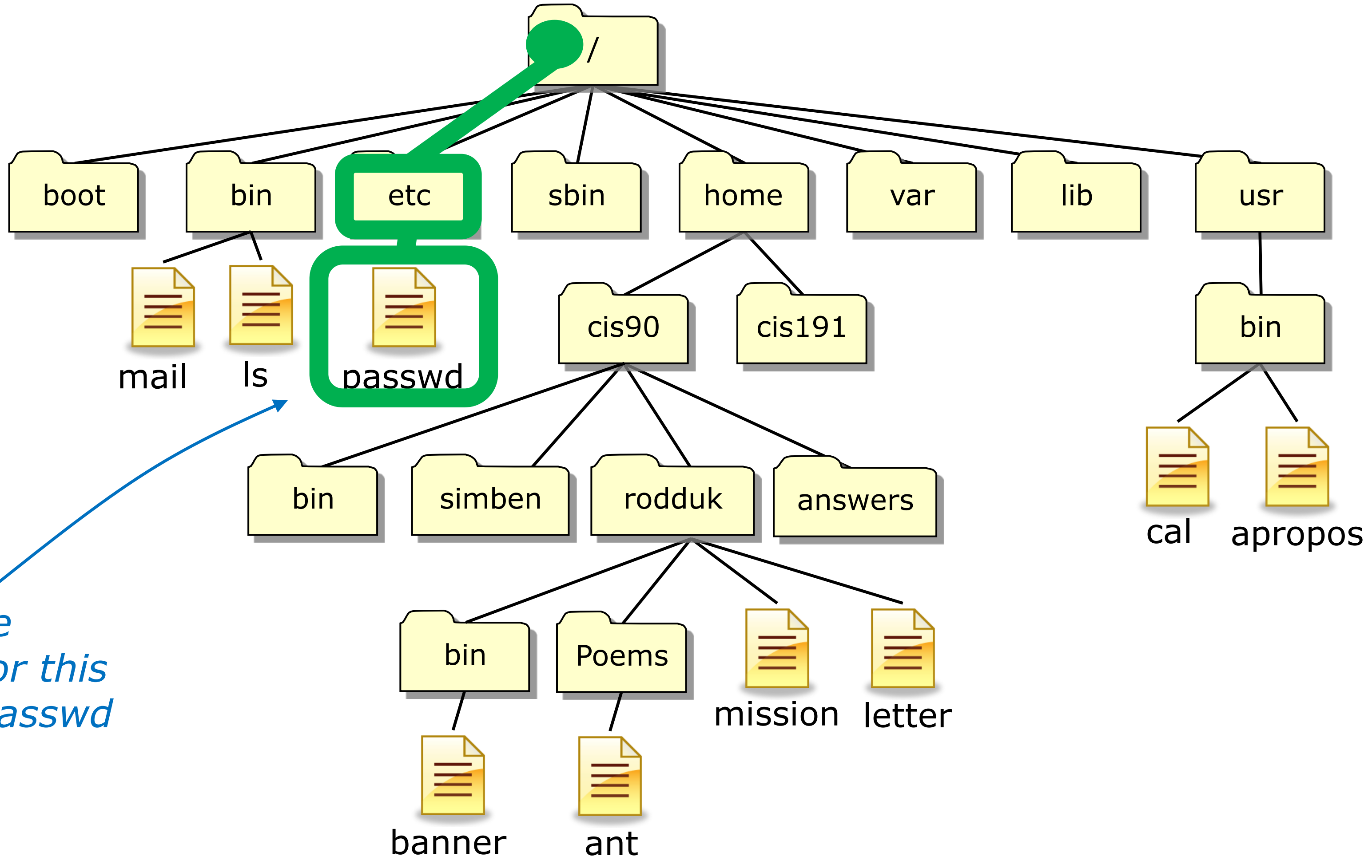
Question: What is the absolute pathname of /etc/passwd?

Answer: /etc/passwd

What is the color of Washington's white horse?

UNIX File Tree

/ = root of the tree



The absolute pathname for this file is /etc/passwd

/

.

..

~

More on Directories

/ 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 shorthand for the absolute path to your current **parent** directory

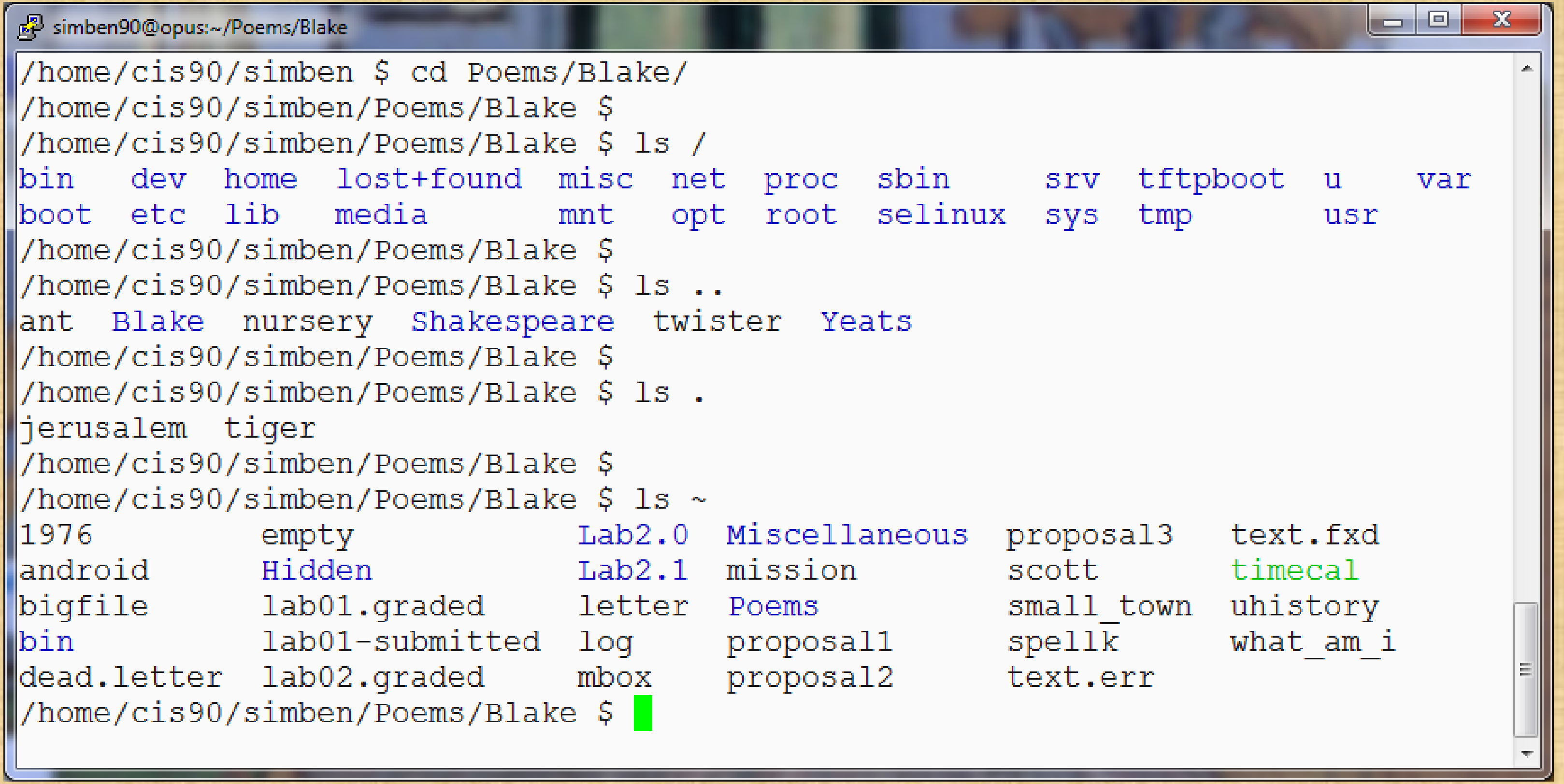
. is shorthand for the absolute path to your current directory = “here”

~ is shorthand for the absolute path to your home directory

. and .. are hidden files, more on hidden files later

Class Activity

1. Change to your Poems/Blake directory using a relative pathname
2. List the directories in the / directory using an absolute pathname
3. List the directories in your current parent directory using ..
4. List the directories in your current directory using .
5. List the file in your home directory using ~



simben90@opus:~/Poems/Blake

```
/home/cis90/simben $ cd Poems/Blake/
/home/cis90/simben/Poems/Blake $
/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
/home/cis90/simben/Poems/Blake $
/home/cis90/simben/Poems/Blake $ ls ..
ant  Blake  nursery  Shakespeare  twister  Yeats
/home/cis90/simben/Poems/Blake $
/home/cis90/simben/Poems/Blake $ ls .
jerusalem  tiger
/home/cis90/simben/Poems/Blake $
/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
/home/cis90/simben/Poems/Blake $
```

UNIX File Hierarchy

/
/bin
/boot
/dev
/etc
/home
/lib
/lost+found
/mnt
/opt
/proc
/root
/sbin
/tmp
/usr

The UNIX/Linux File System Hierarchy

There are standard top level directories in every version of UNIX/Linux

CIS 90 - Lesson 4

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

Navigating the UNIX file tree

Navigating the tree

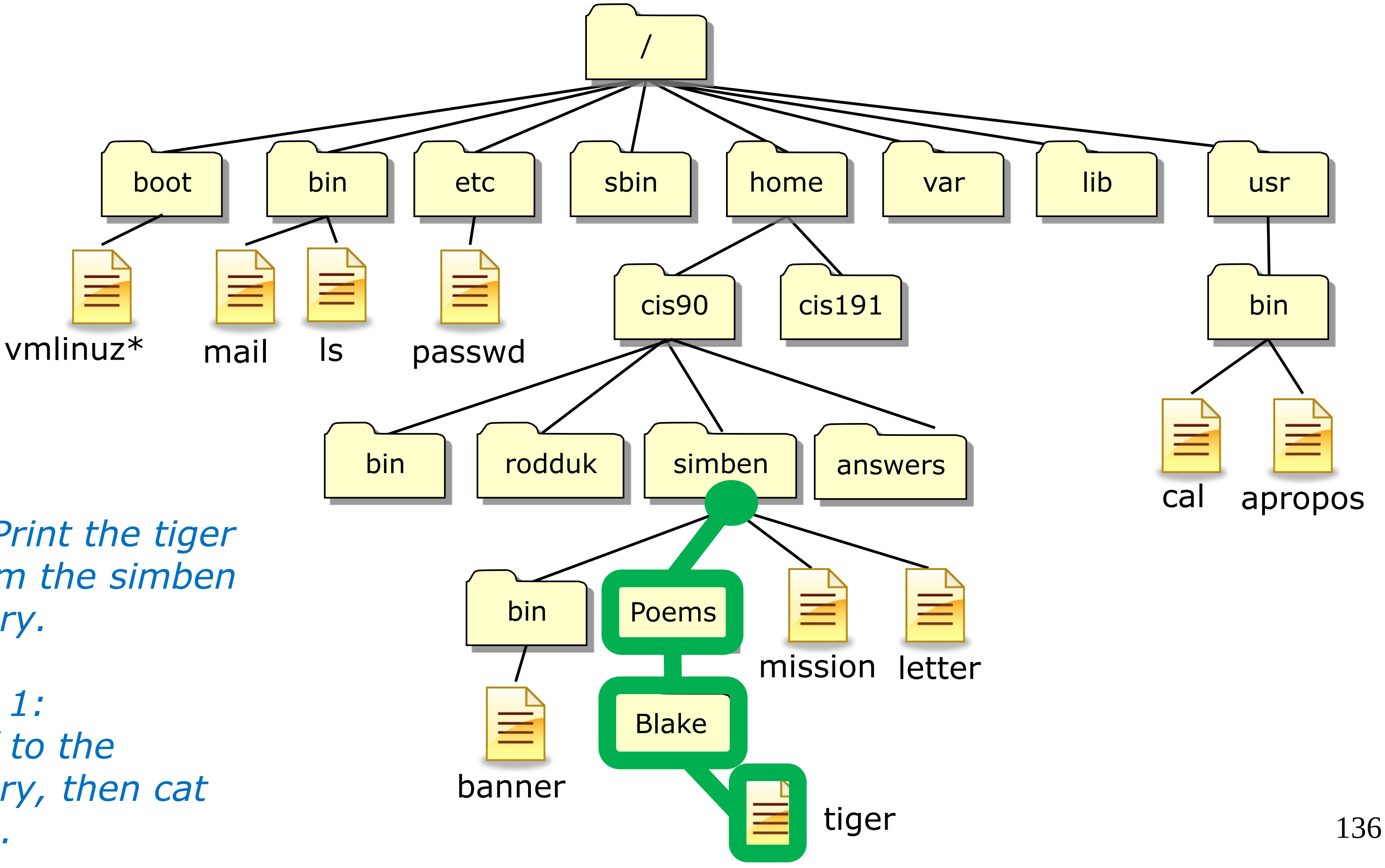
- Use the **cd** command to change directories *(your legs)*

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

- Use the **pwd** command to check where you are (your GPS)

*Note, as CIS 90 students your command prompt has been configured to show what you would normally get with the **pwd** command. As you move around the tree your command prompt will change to show your current location.*

Printing a file in another directory



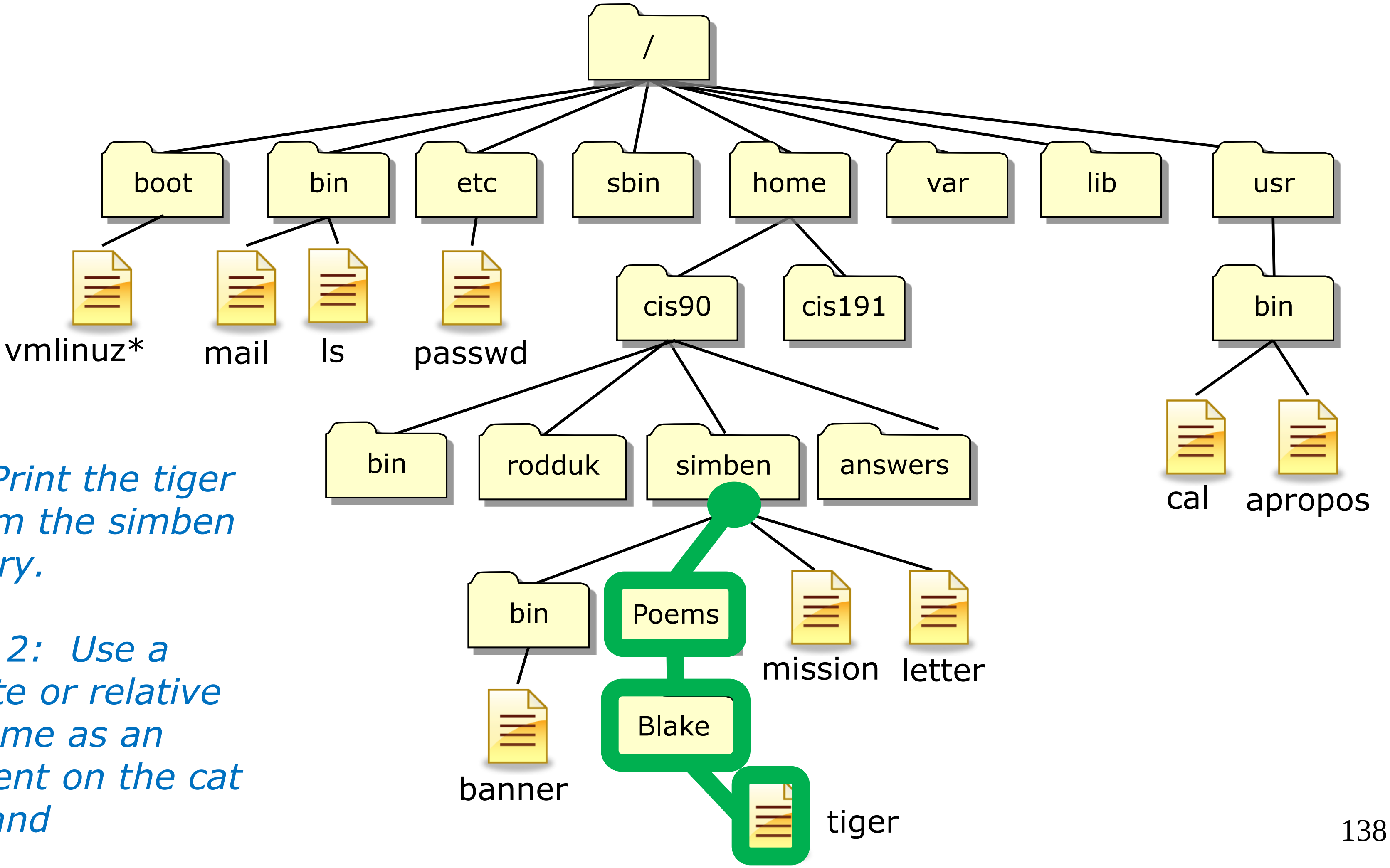
Task: Print the tiger file from the simben directory.

Option 1: "Walk" to the directory, then cat the file.

Printing a file by navigating to the directory first

```
/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/         go down into Poems directory
/home/cis90/simben/Poems $ ls          see what's there
ant  Blake  nursery  Shakespeare  twister  Yeats

/home/cis90/simben/Poems $ cd Blake/        go down into 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,               print tiger file
What immortal hand or eye
Dare frame thy fearful symmetry?
```

Printing a file in another directory



Task: Print the tiger file from the simben directory.

Option 2: Use a absolute or relative pathname as an argument on the cat command

Printing files in other directories using pathnames

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

start in our home directory

```
/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 $
```

cat the tiger file using a relative pathname

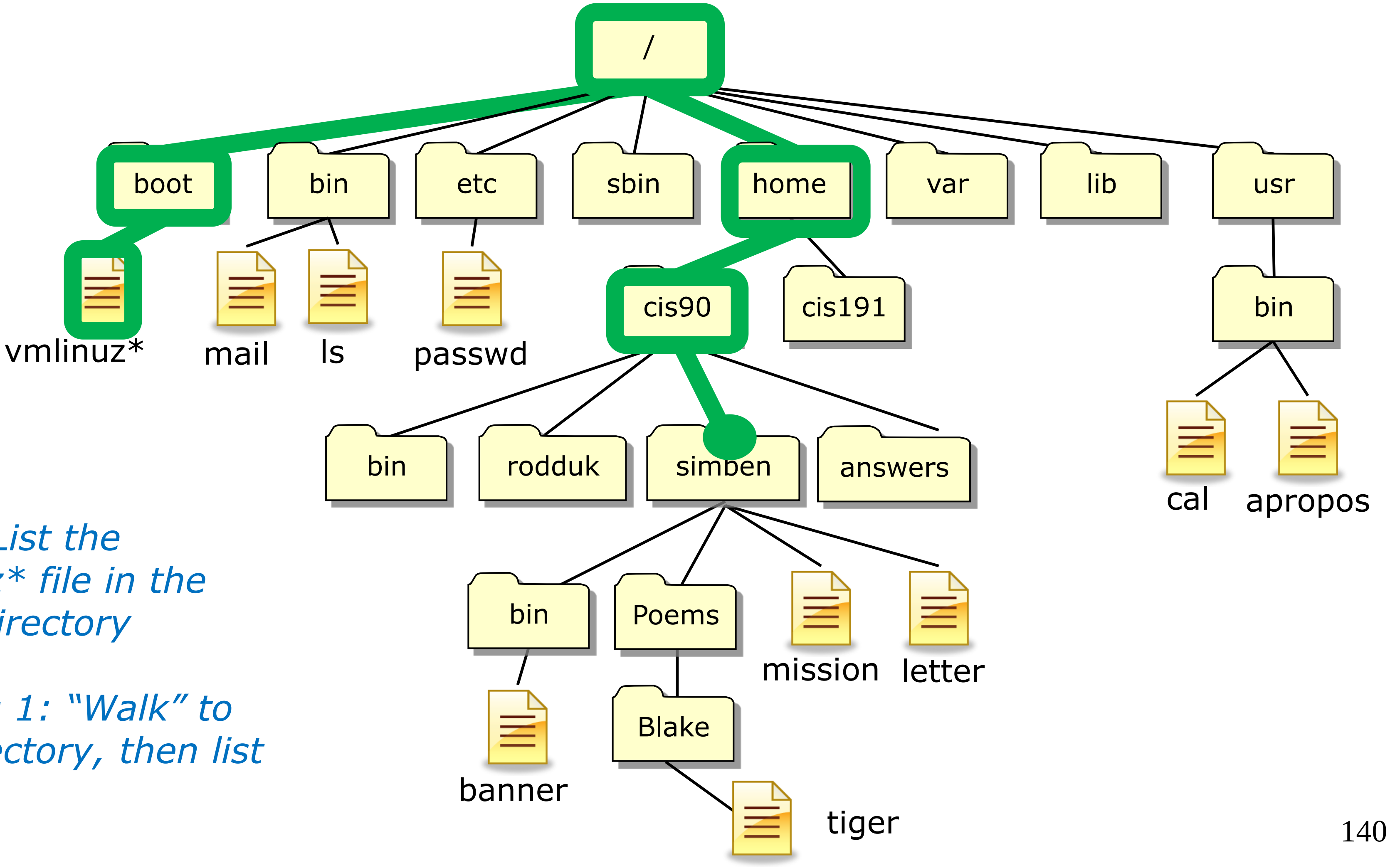
```
/home/cis90/simben $ cat /home/cis90/simben/Poems/Blake/tiger
Tiger, Tiger burning bright
In the forest of the night,
What immortal hand or eye
Dare frame thy fearful symmetry?
/home/cis90/simben $
```

cat the tiger file using an absolute pathname

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

*NOTE: Attempting to cat the tiger file with an incorrect pathname doesn't work (the tiger file is **not** in our home directory)*

Listing a file in another directory – Option 1



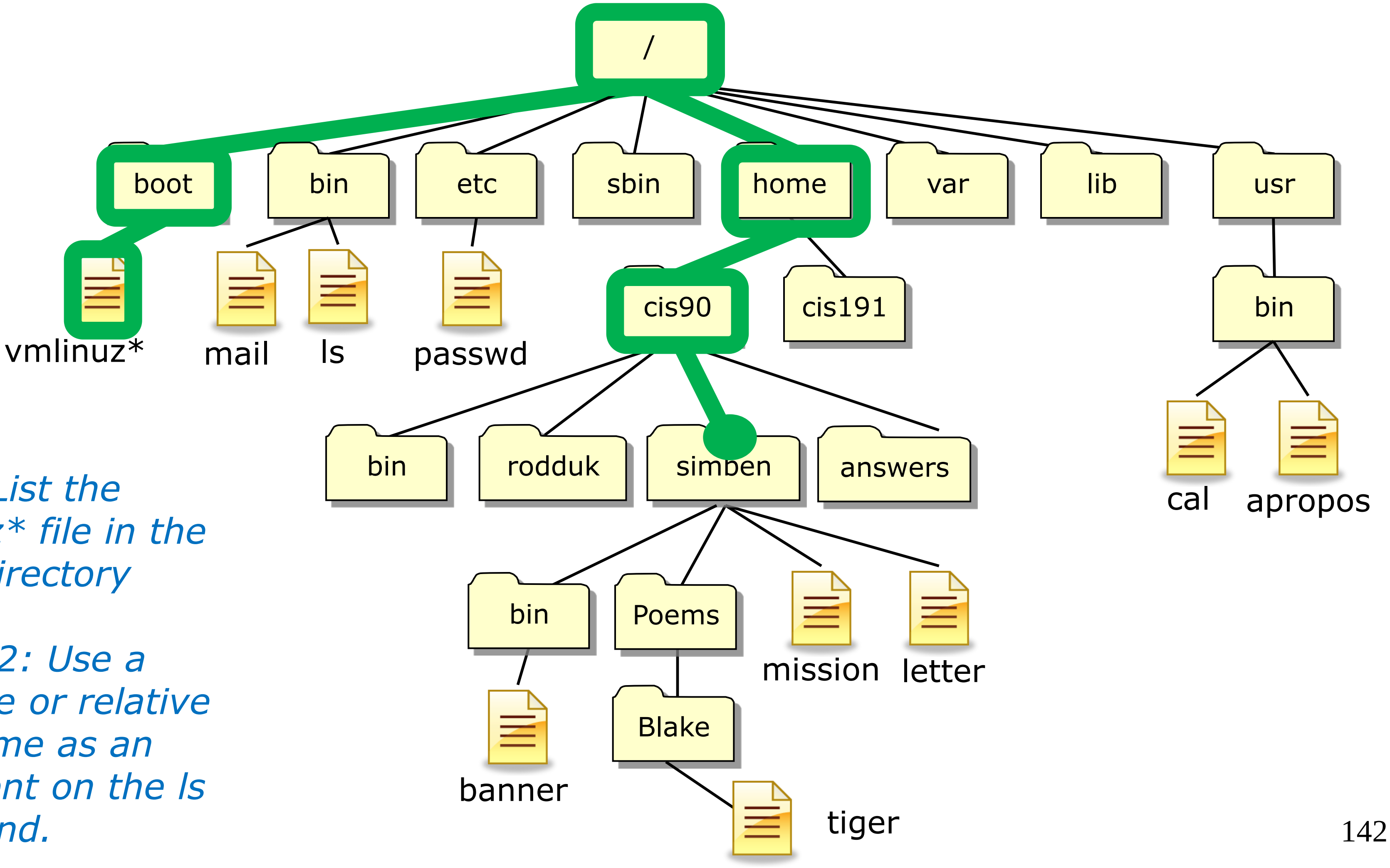
Task: List the vmlinuz file in the /boot directory*

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     student
calsea    ellcar   fyosea   kenrit   menfid   noreva   rawjes   wiljac     home
cis       evaand   guest    libkel   mescha   potjos   rodduk   zamhum     directories

                                                       my home
/home/cis90 $ cd ..   go up again                      directory            where labs are
/home $ ls    look around                                                   submitted
cis172  cis90  cis98  gerlinde  guest  jimg  lost+found  rick  rsimms  turnin
                                        our class directory
/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          symvers-2.6.32-220.23.1.el6.i686.gz
config-2.6.32-71.el6.i686                symvers-2.6.32-71.el6.i686.gz
efi                                      System.map-2.6.32-220.23.1.el6.i686
grub                                     System.map-2.6.32-71.el6.i686
initramfs-2.6.32-220.23.1.el6.i686.img   vmlinuz-2.6.32-220.23.1.el6.i686  ← Newer Linux kernel
initramfs-2.6.32-71.el6.i686.img         vmlinuz-2.6.32-71.el6.i686  ← 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

using an absolute pathname as the argument

```
/home/cis90/simben $ ls -l /boot/vmlinuz-2.6.32-220.23.1.el6.i686
-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

Navigating

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/milhom/Poems/ant**
  A relative pathname, e.g. **cd Poems, 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
```

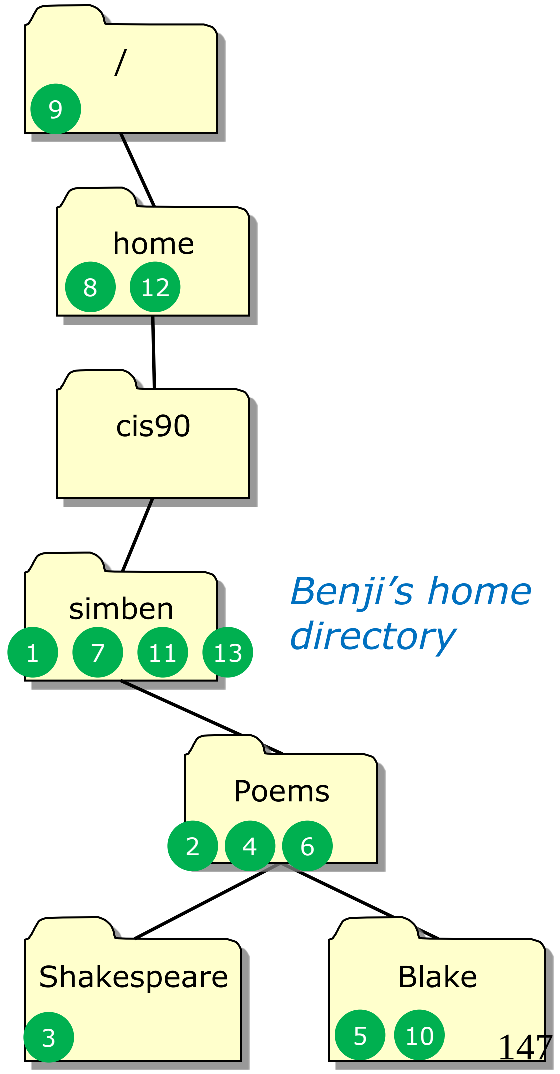
The .. directory

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/simmen $ echo $HOME
   /home/cis90/simben
   /home/cis90/simmsben $ echo $PS1
   $PWD $
1  /home/cis90/simmen $ 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 $
```



Navigating

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

This is a UNIX command

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

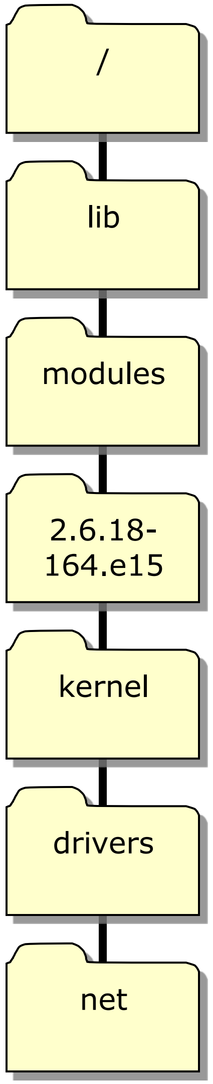
This is a UNIX command

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

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

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.



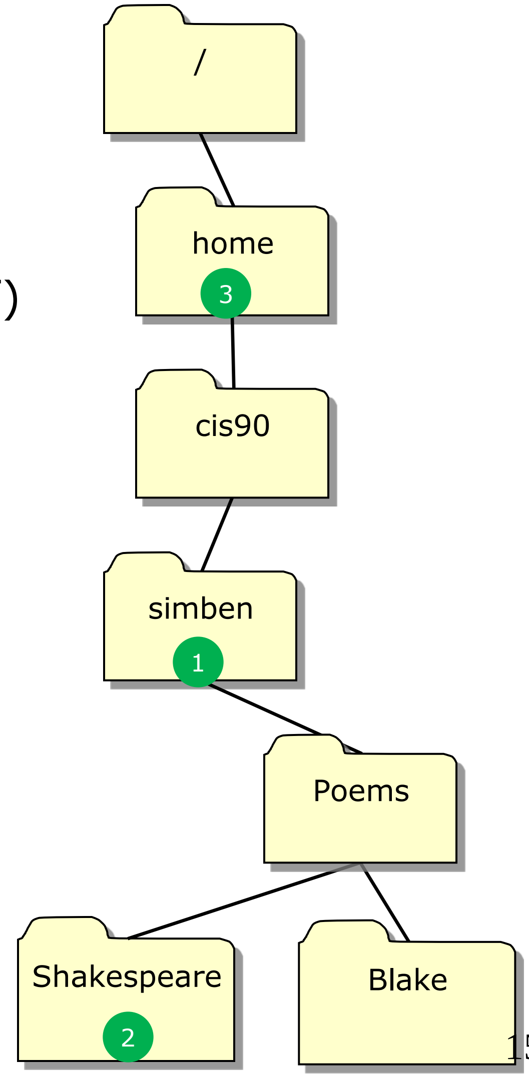
pwd command
print working directory

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 $
```



Navigating

ls command (your eyes)

ls 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
-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

Regular files in black

```
/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
```

Directories in blue

Executables (programs or scripts) in green

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

ls command example

with the -F option

Regular files have no suffix

```
/home/cis90/simmsben $ ls -F
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*
```

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

```
/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    .lesshst   .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 ".")

***..** is the parent directory*

***.** 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     12101 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*

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?

Answer: At least four ways:

1

```
/home/cis90/simben $ ls -id /home/cis90/simben/
8971 /home/cis90/simben/
```

Specify the absolute pathname of the home directory

2

```
/home/cis90/simben $ ls -id .
8971 .
```

Using the . if you are currently in your home directory

3

```
/home/cis90/simben $ ls -id ~
8971 /home/cis90/simben
```

The ~ is always an absolute pathname to home directory

4

```
/home/cis90/simben $ ls -i /home/cis90
13658 answers    9135 evaand   9015 kenrit   8975 milhom   8971 simben
12625 bin       15015 farsha   9019 libkel   9039 noreva   9123 verevi
 8991 calsea     9003 frocar   9023 lyoben   9059 potjos   9071 wiljac
 8967 cis        9099 fyosea   9027 marray   9044 ramcar   9075 zamhum
 8995 davdon    11282 guest    9031 menfid   9063 ramgus
12656 depot      9007 hendaj   9131 mescha   9127 rawjes
 8999 ellcar     9011 kanbry   9035 mesmic   8979 rodduk
```

Using contents of the parent directory

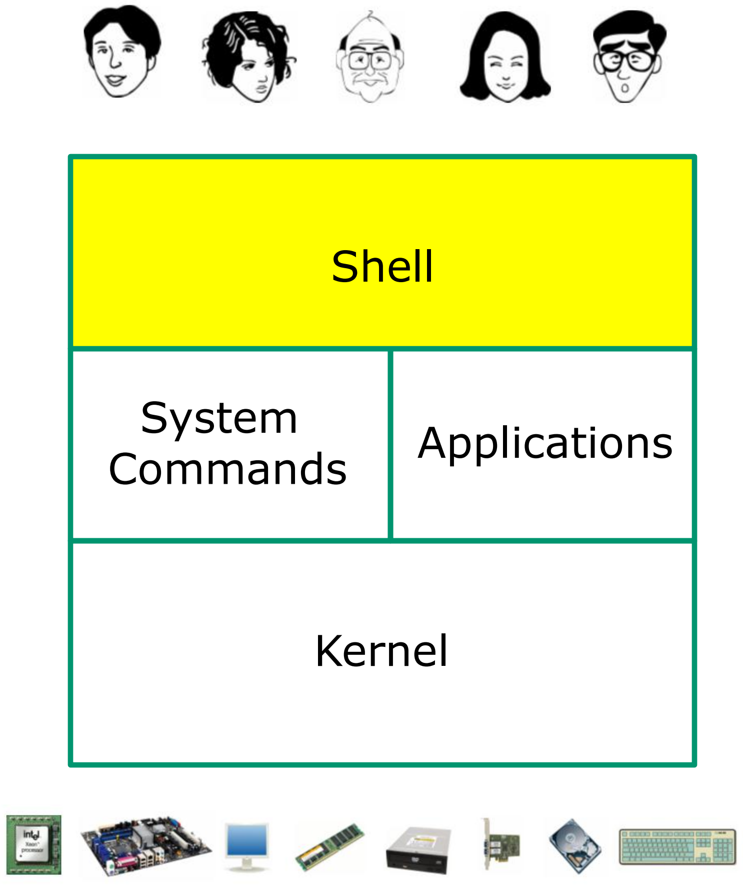
Note the use of the –d option on ls to focus on the directory itself rather than the directory contents

Class Exercise

- What is the name of your home directory?
- What is the inode number of your home directory?
- What file type is your home directory?
- What is the absolute path of your home directory?

*
metacharacter

Life of the Shell



1) Prompt

2) Parse

3) Search

4) Execute

5) Nap

6) Repeat

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

(before the selected command is even run)

*

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 lets the shell do the work instead

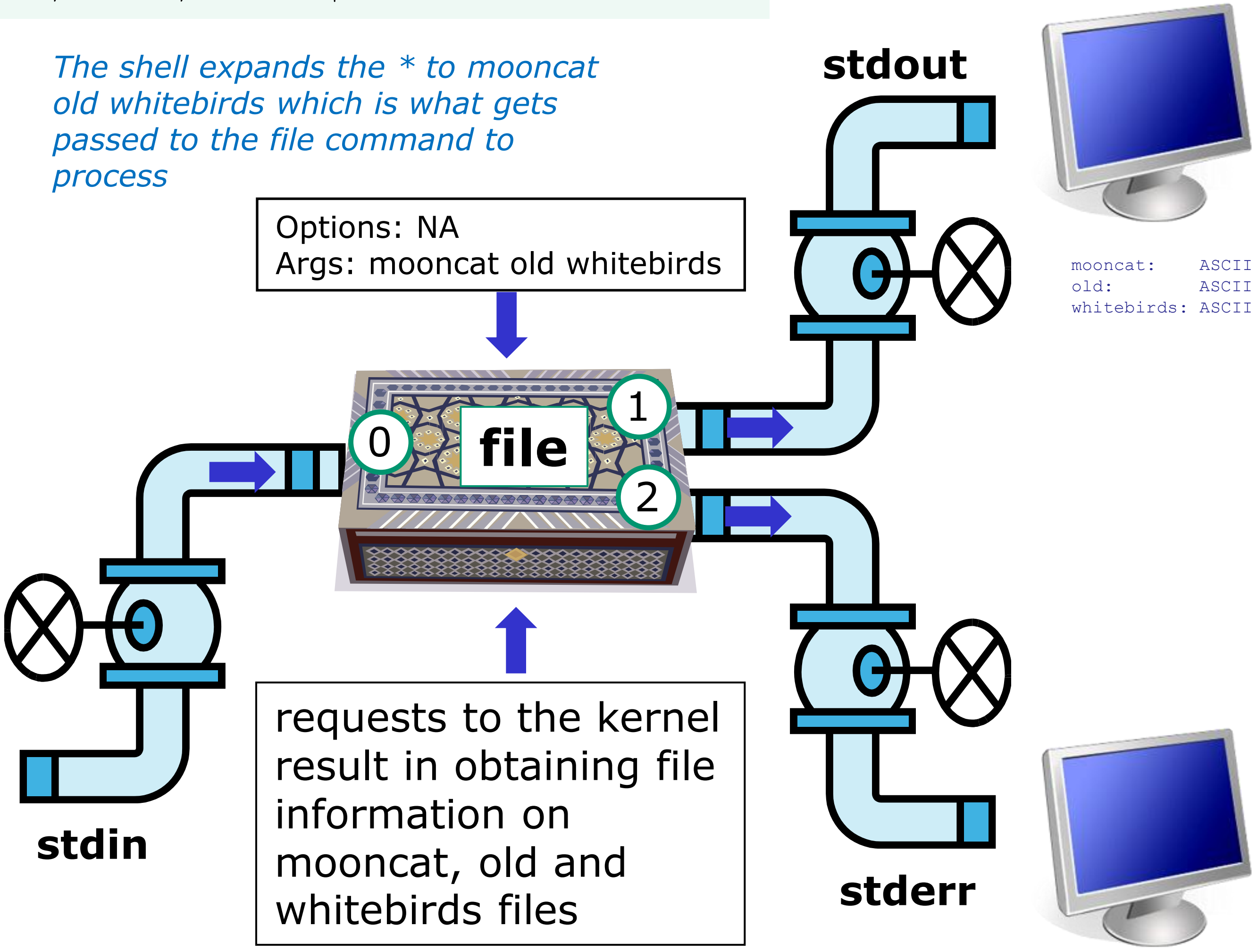
*In the second example, the shell, during the parse step, expands the * and replaces it with mooncat old whitebirds.*

*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 what gets passed to the file command to process*



```
mooncat:    ASCII English text
old:        ASCII English text
whitebirds: ASCII English text
```

requests to the kernel result in obtaining file information on mooncat, old and whitebirds files

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

* metacharacter

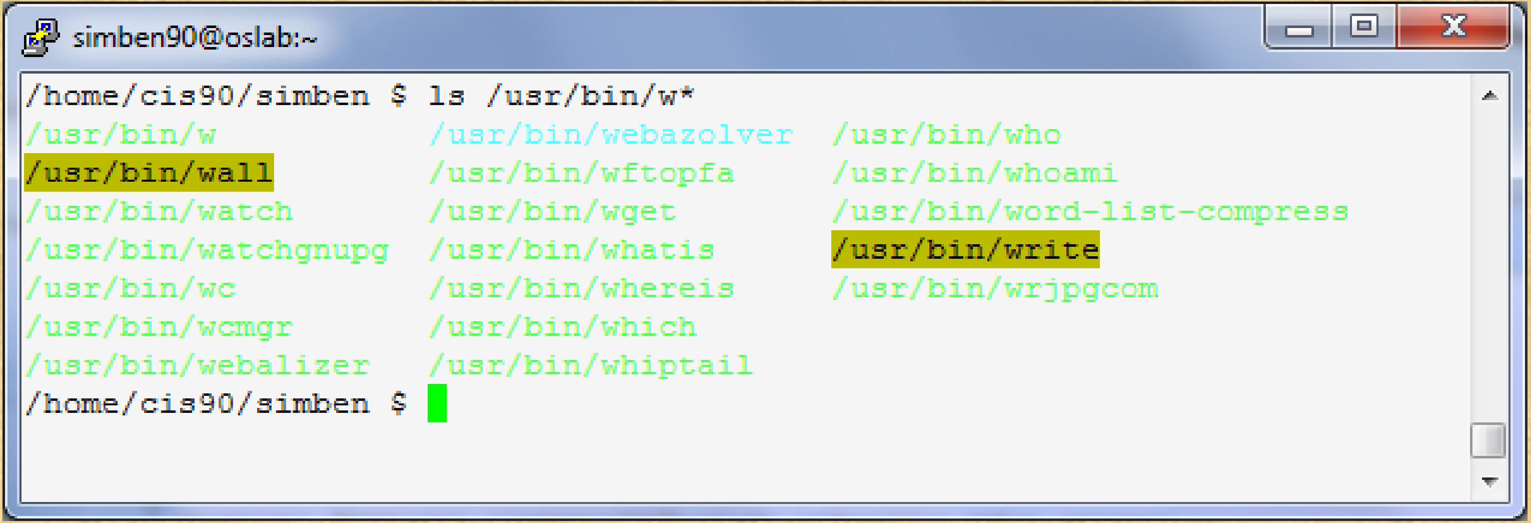
used as a *postfix* character

```
/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/simmen $ ls p*
proposal1  proposal2  proposal3
```

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

Class Activity

What commands in the /usr/bin directory starts with the letter w?



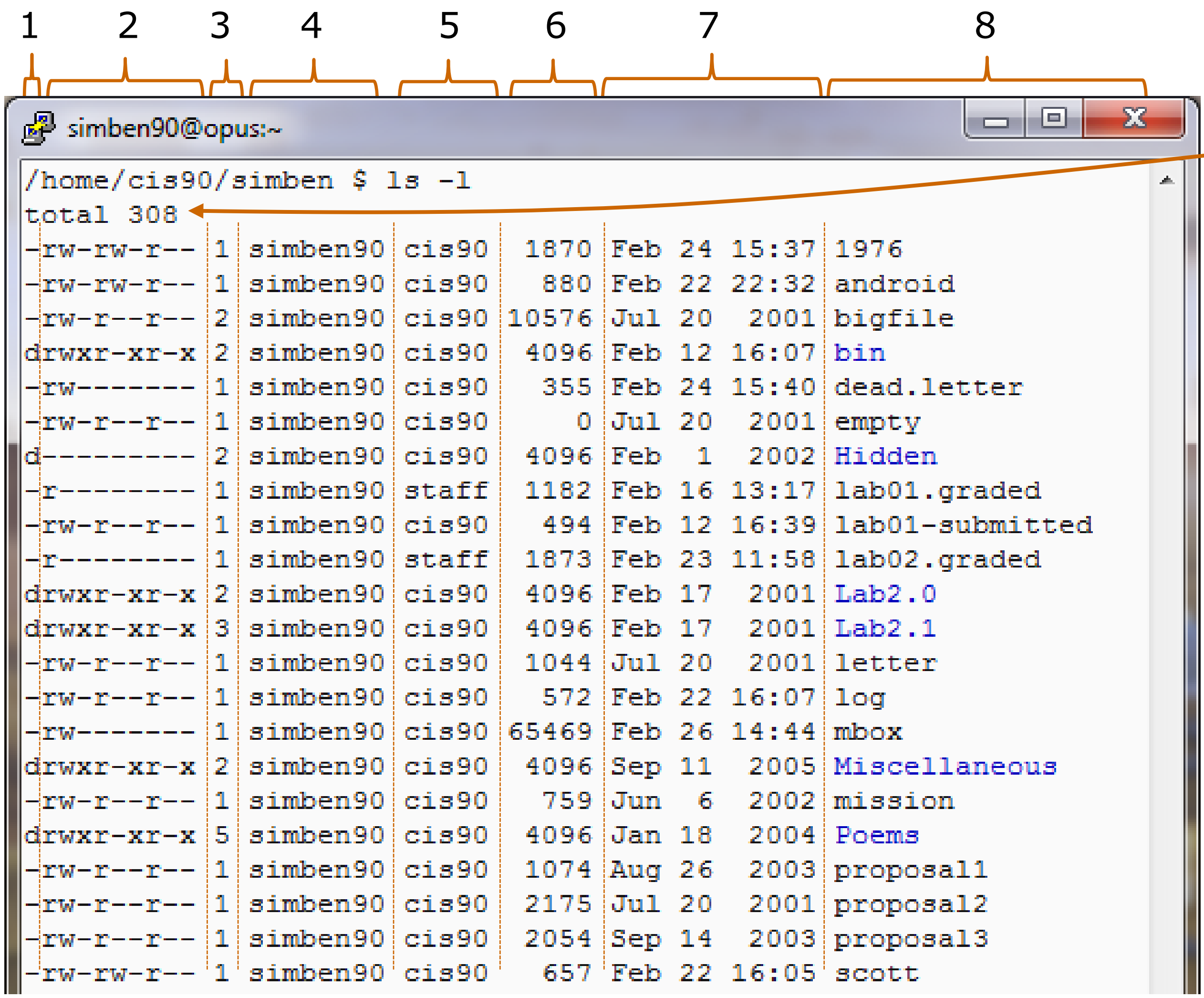
simben90@oslab:~

```
/home/cis90/simben $ ls /usr/bin/w*
/usr/bin/w              /usr/bin/webazolver  /usr/bin/who
/usr/bin/wall           /usr/bin/wftopfa     /usr/bin/whoami
/usr/bin/watch          /usr/bin/wget        /usr/bin/word-list-compress
/usr/bin/watchgnupg     /usr/bin/whatis      /usr/bin/write
/usr/bin/wc             /usr/bin/whereis     /usr/bin/wrjpgcom
/usr/bin/wcmgr          /usr/bin/which
/usr/bin/webalizer      /usr/bin/whiptail
/home/cis90/simben $
```

More on the ls command

ls command

Use the -l option for a “long listing”



```
/home/cis90/simben $ ls -l
total 308
-rw-rw-r-- 1 simben90 cis90  1870 Feb 24 15:37 1976
-rw-rw-r-- 1 simben90 cis90   880 Feb 22 22:32 android
-rw-r--r-- 2 simben90 cis90 10576 Jul 20  2001 bigfile
drwxr-xr-x 2 simben90 cis90  4096 Feb 12 16:07 bin
-rw------- 1 simben90 cis90   355 Feb 24 15:40 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  1182 Feb 16 13:17 lab01.graded
-rw-r--r-- 1 simben90 cis90   494 Feb 12 16:39 lab01-submitted
-r-------- 1 simben90 staff  1873 Feb 23 11:58 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
-rw-r--r-- 1 simben90 cis90   572 Feb 22 16:07 log
-rw------- 1 simben90 cis90 65469 Feb 26 14:44 mbox
drwxr-xr-x 2 simben90 cis90  4096 Sep 11  2005 Miscellaneous
-rw-r--r-- 1 simben90 cis90   759 Jun  6  2002 mission
drwxr-xr-x 5 simben90 cis90  4096 Jan 18  2004 Poems
-rw-r--r-- 1 simben90 cis90  1074 Aug 26  2003 proposal1
-rw-r--r-- 1 simben90 cis90  2175 Jul 20  2001 proposal2
-rw-r--r-- 1 simben90 cis90  2054 Sep 14  2003 proposal3
-rw-rw-r-- 1 simben90 cis90   657 Feb 22 16:05 scott
```

total size of all files in blocks

*On Opus,
1 block = 1024 bytes*

1. file type
   - = regular
   d = directory
   l = link
2. permissions
3. number of hard links
4. owner
5. group
6. size (in bytes)
7. last modified
8. file name

ls command

Using files vs directories as arguments

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

```
/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 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*

regular file

directories

When a file is specified, just the filename is listed

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

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

bin/:
app  banner  enlightenment  hi  I  treed  tryme  zoom

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

ls command example

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

```
/home/cis90/simmsben $ ls *
bigfile  letter   proposal1  proposal3   spellk     text.fxd  what_am_i
empty    mission  proposal2  small_town  text.err   timecal

bin:
app  banner  enlightenment  hi  I  treed  tryme  zoom
ls: Hidden: Permission denied

Lab2.0:
386    A_long_name    file.9        README NAME    this_years_annual_report
afile  annual report  junk.old.bak  sTrAnGeNeSs

Lab2.1:
1.1  filename  junk  letter  more  old  Proposal3  Proposal.old  xyz

Miscellaneous:
better_town  file.dos  fruit  manpage  mystery  salad

Poems:
ant  Blake  nursery  Shakespeare  twister  Yeats
```

Files listed first

Then the contents of each directory are listed

Do you see the error message? … permission issue (more in future lessons)
Do you see the symbolic link? … in light blue (more in future lessons)

ls command

directory itself vs. contents of a directory (short listing)

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

The contents of the directory are shown

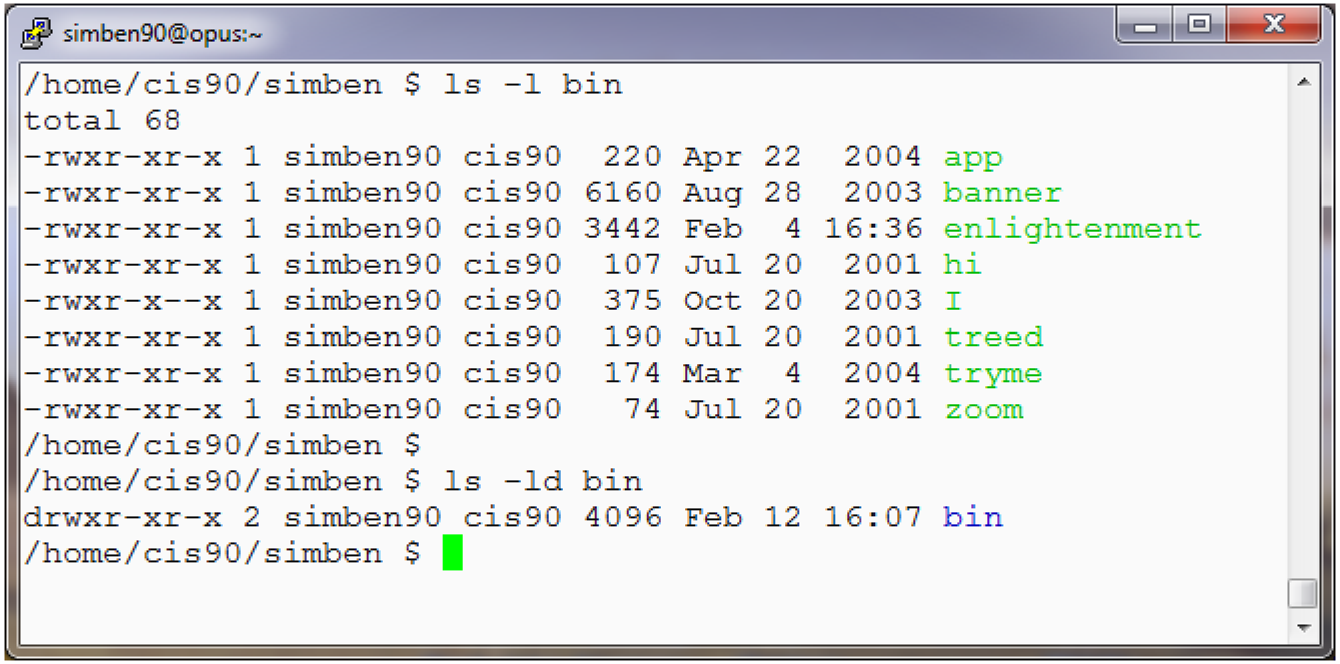
```
/home/cis90/simben $ ls -d bin
bin
```

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

directory itself vs. contents of directory (long listing)



```
/home/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
/home/cis90/simben $
/home/cis90/simben $ ls -ld bin
drwxr-xr-x 2 simben90 cis90 4096 Feb 12 16:07 bin
/home/cis90/simben $
```

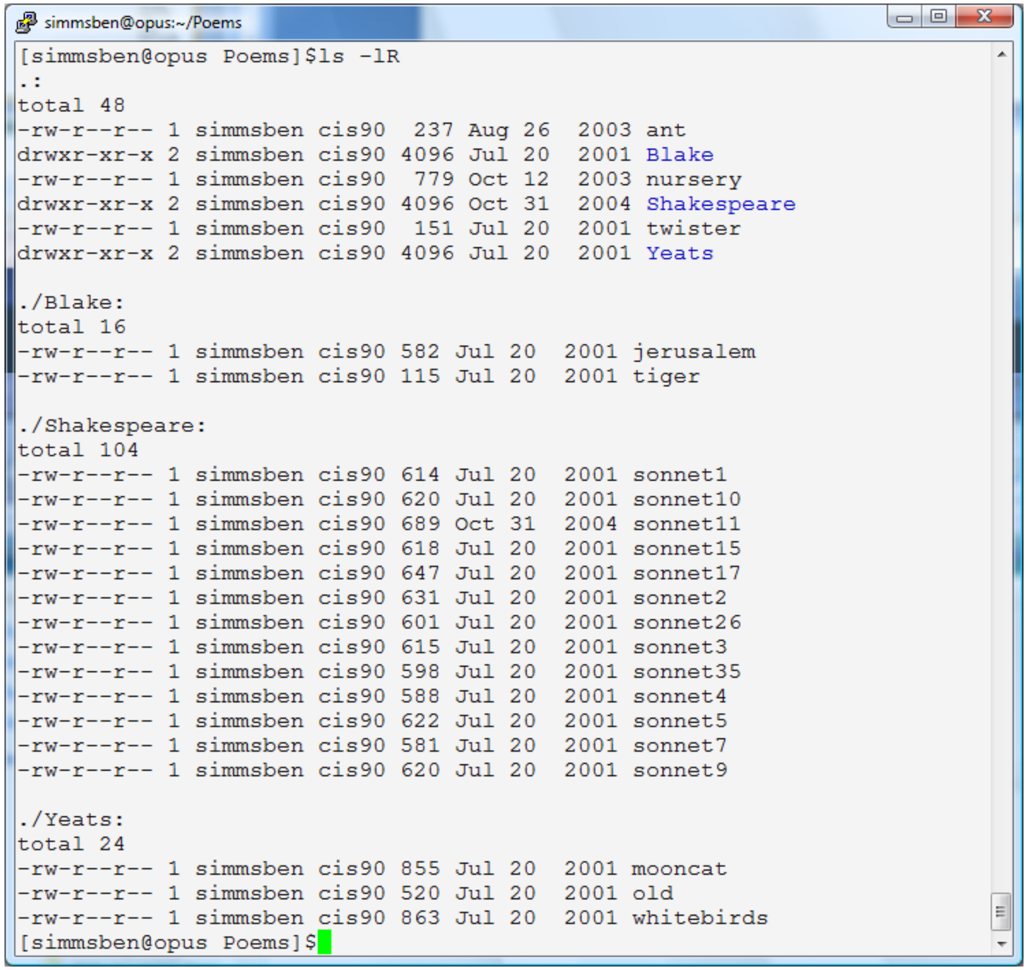
The contents of the directory are shown

The directory itself is shown with the -d option

Tip: use the -l and -d options on the ls command to get owner and permission information on directories

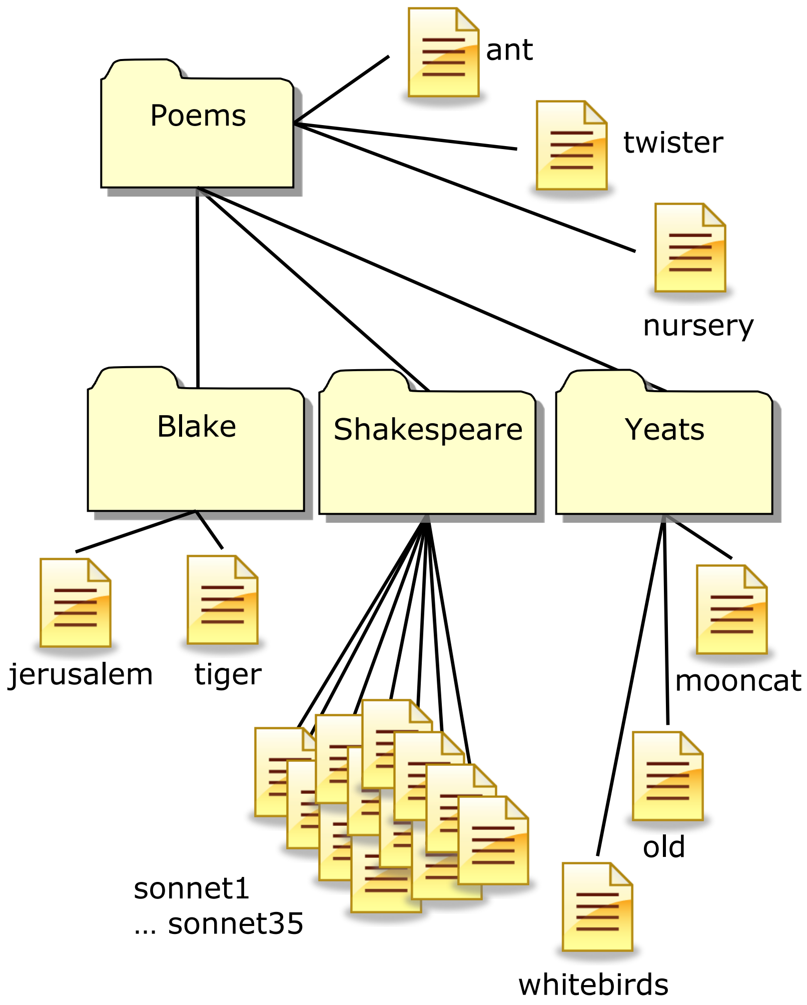
ls command

long listing (-l), recursively list subdirectories (-R)



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]$
```



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 –ilR

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?"
```

name

+

inode

+

data

Class Exercise
Enlightenment

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

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/*username*	The home directory for CIS 90 student *username (without the 90)*
/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 paths?

3) What are the three elements of a UNIX file?

Backup

Review

Parsing & Command Syntax

Shell prints this to prompt user to enter a command

Shell parses this command line

$\underbrace{\text{Prompt}}$ $\underbrace{\text{Command Options Arguments Redirection}}$

Examples

```
/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
```

Options *modify the behavior of the command*

Arguments *are what the command works upon*

Redirection *is covered later in the course*

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

Lab 2 Results

4. Set the TERM environment variable to "dumb", and execute the **clear** command. What does it do? Use **echo $TERM** to see the new setting. Set TERM back to "vt100" or "ansi" What happens?

TERM="dumb"
TERM="ansi"

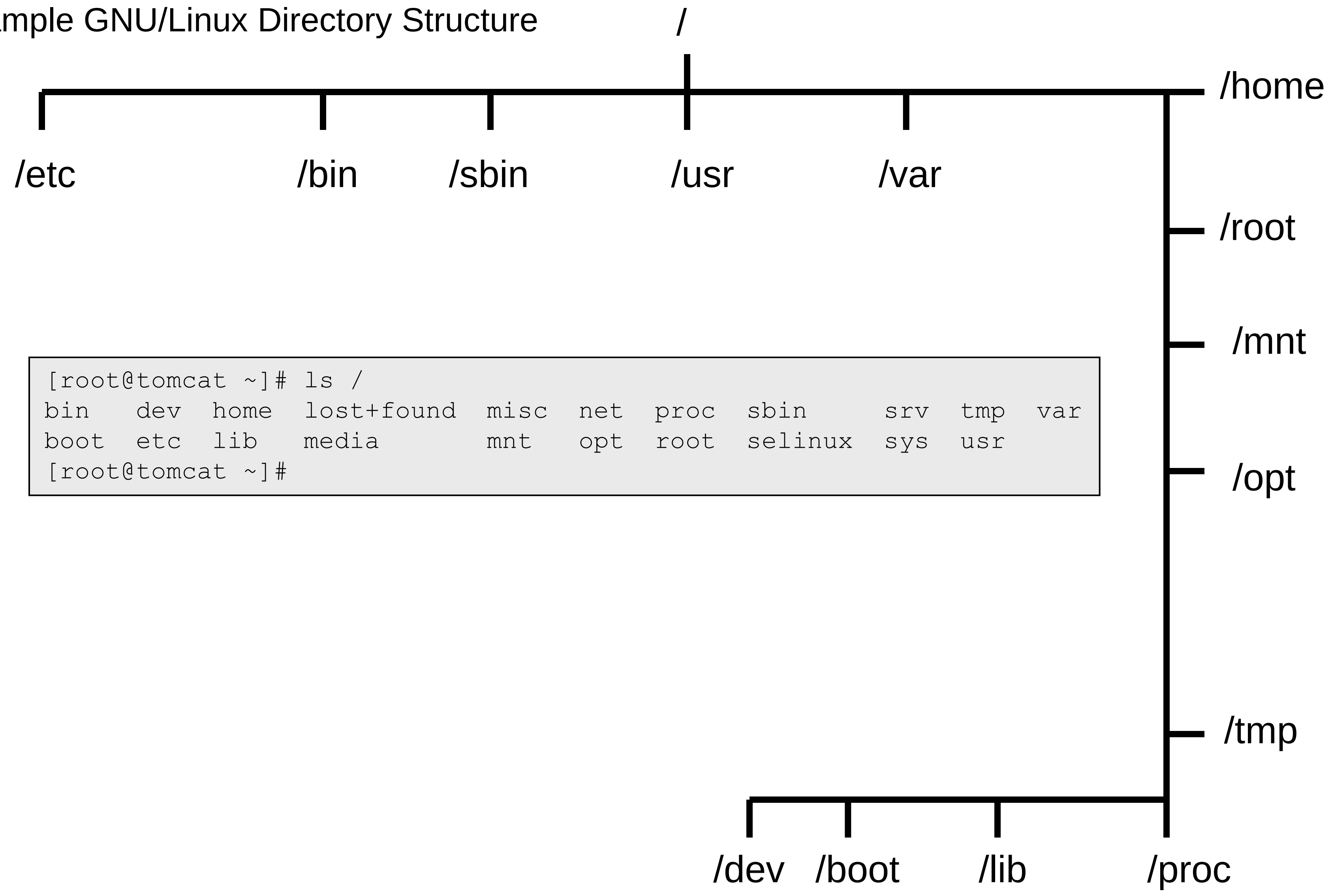
Set the TERM environment variable back to "xterm" which is what it was when you logged in.

Lab 2 Results

12. What is the difference in output between the following two commands?

```
banner I am fine
banner "I am fine"
```

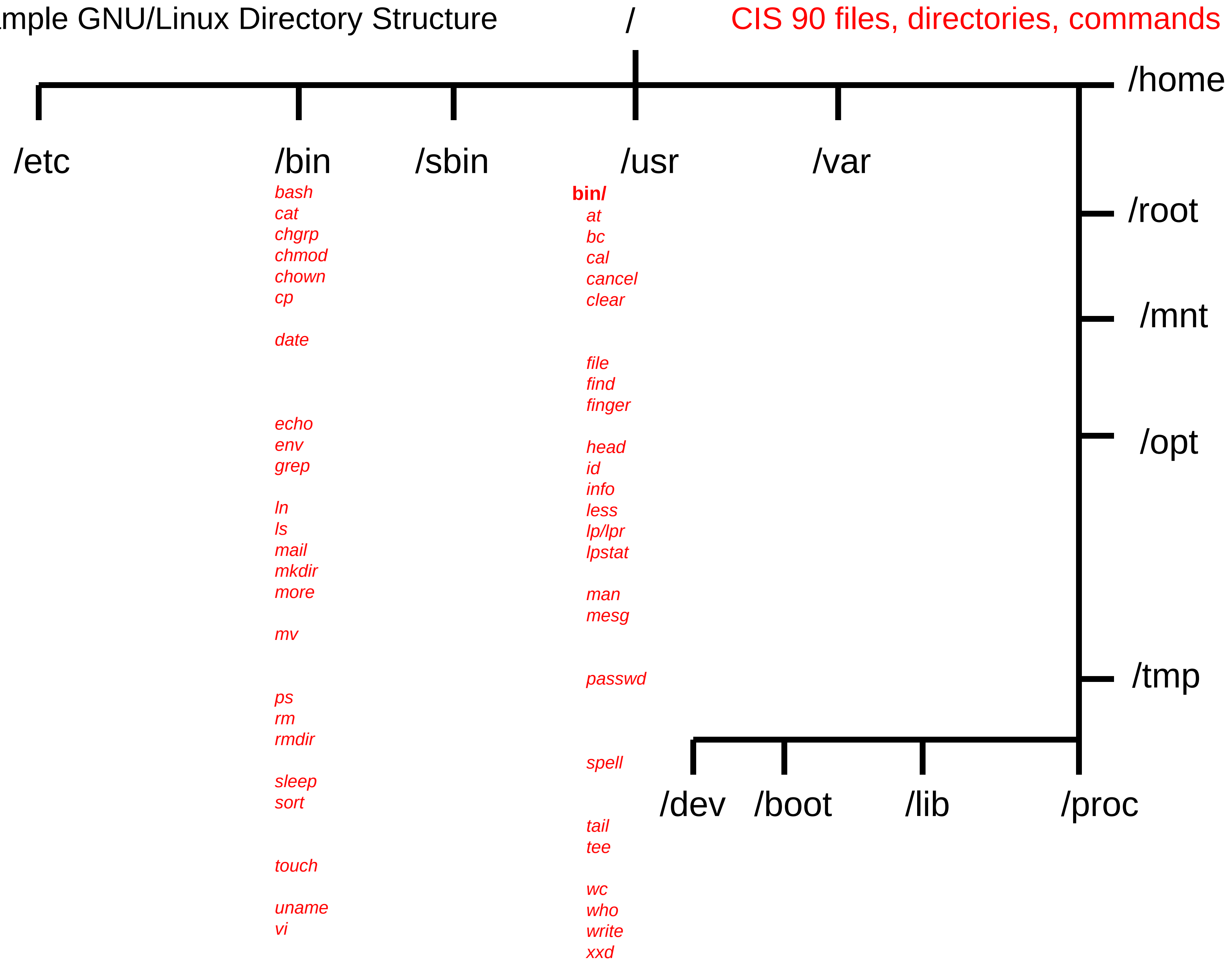
Example GNU/Linux Directory Structure



```
[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

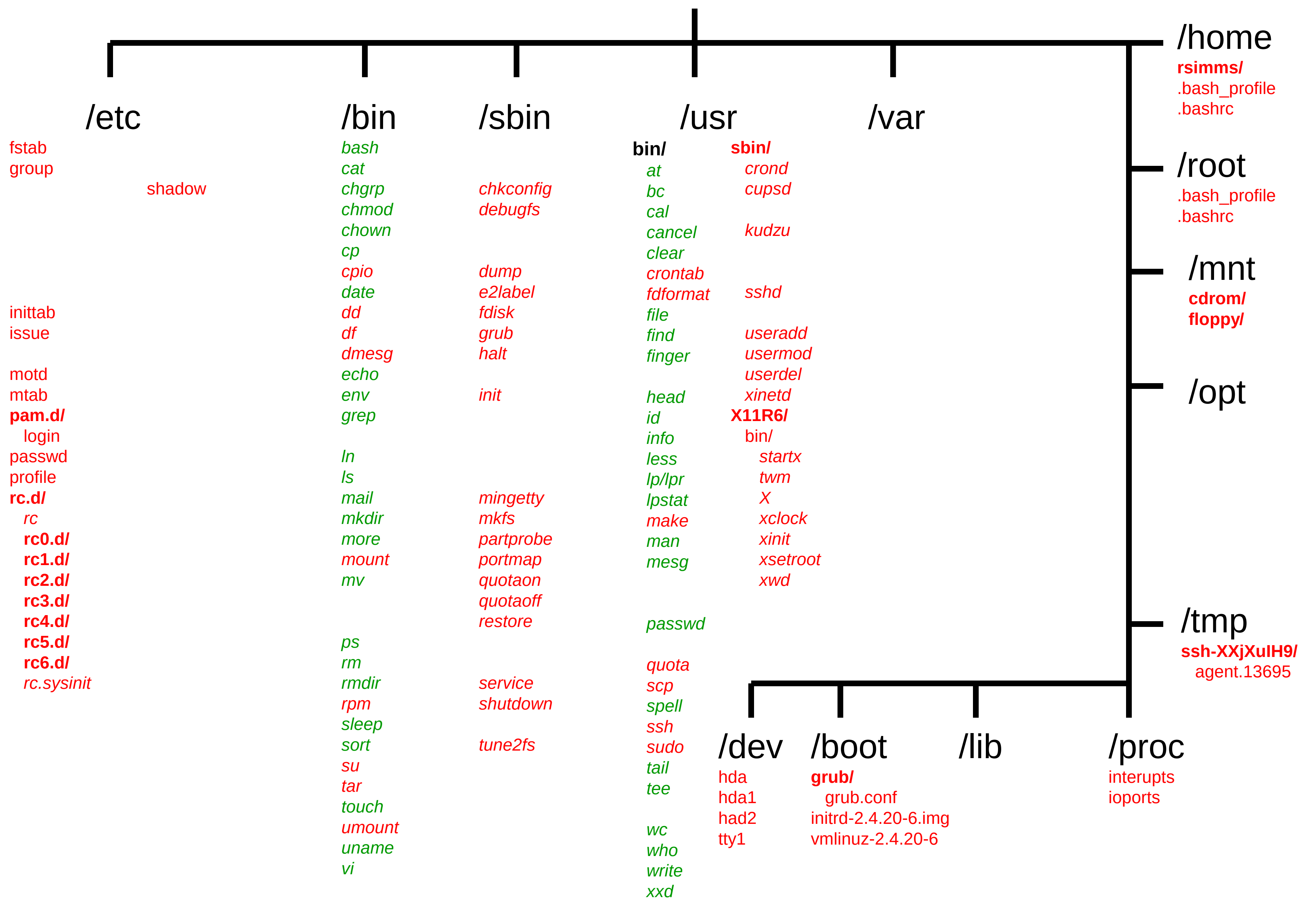
CIS 90 files, directories, commands



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



/

- /etc
  - fstab
  - group
  - shadow
  - inittab
  - issue
  - motd
  - mtab
  - **pam.d/**
    - login
  - passwd
  - profile
  - **rc.d/**
    - *rc*
    - **rc0.d/**
    - **rc1.d/**
    - **rc2.d/**
    - **rc3.d/**
    - **rc4.d/**
    - **rc5.d/**
    - **rc6.d/**
    - *rc.sysinit*
- /bin
  - *bash*
  - *cat*
  - *chgrp*
  - *chmod*
  - *chown*
  - *cp*
  - *cpio*
  - *date*
  - *dd*
  - *df*
  - *dmesg*
  - *echo*
  - *env*
  - *grep*
  - *ln*
  - *ls*
  - *mail*
  - *mkdir*
  - *more*
  - *mount*
  - *mv*
  - *ps*
  - *rm*
  - *rmdir*
  - *rpm*
  - *sleep*
  - *sort*
  - *su*
  - *tar*
  - *touch*
  - *umount*
  - *uname*
  - *vi*
- /sbin
  - *chkconfig*
  - *debugfs*
  - *dump*
  - *e2label*
  - *fdisk*
  - *grub*
  - *halt*
  - *init*
  - *mingetty*
  - *mkfs*
  - *partprobe*
  - *portmap*
  - *quotaon*
  - *quotaoff*
  - *restore*
  - *service*
  - *shutdown*
  - *tune2fs*
- /usr
  - **bin/**
    - *at*
    - *bc*
    - *cal*
    - *cancel*
    - *clear*
    - *crontab*
    - *fdformat*
    - *file*
    - *find*
    - *finger*
    - *head*
    - *id*
    - *info*
    - *less*
    - *lp/lpr*
    - *lpstat*
    - *make*
    - *man*
    - *mesg*
    - *passwd*
    - *quota*
    - *scp*
    - *spell*
    - *ssh*
    - *sudo*
    - *tail*
    - *tee*
    - *wc*
    - *who*
    - *write*
    - *xxd*
  - **sbin/**
    - *crond*
    - *cupsd*
    - *kudzu*
    - *sshd*
    - *useradd*
    - *usermod*
    - *userdel*
    - *xinetd*
  - **X11R6/**
    - bin/
      - *startx*
      - *twm*
      - *X*
      - *xclock*
      - *xinit*
      - *xsetroot*
      - *xwd*
- /var
- /home
  - **rsimms/**
    - .bash_profile
    - .bashrc
- /root
  - .bash_profile
  - .bashrc
- /mnt
  - **cdrom/**
  - **floppy/**
- /opt
- /tmp
  - **ssh-XXjXuIH9/**
    - agent.13695
- /dev
  - hda
  - hda1
  - had2
  - tty1
- /boot
  - **grub/**
    - grub.conf
  - initrd-2.4.20-6.img
  - vmlinuz-2.4.20-6
- /lib
- /proc
  - interupts
  - ioports

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

Example GNU/Linux Directory Structure

/

CIS 192 files, directories, commands

/etc

- fstab
- group
- hosts
- hosts.allow
- hosts.deny
- inittab
- issue
- modules.conf
- motd
- mtab
- **pam.d/**
  - login
- passwd
- profile
- **rc.d/**
  - *rc*
  - **rc0.d/**
  - **rc1.d/**
  - **rc2.d/**
  - **rc3.d/**
  - **rc4.d/**
  - **rc5.d/**
  - **rc6.d/**
  - *rc.sysinit*
- resolv.conf
- shadow
- sysctl.conf
- **sysconfig/**
  - network
  - **network-scripts/**
    - ifcfg-eth0
- **xinetd.d/**
  - telnet

/bin

bash, cat, chgrp, chmod, chown, cp, cpio, date, dd, df, dmesg, echo, env, grep, hostname, ln, ls, mail, mkdir, more, mount, mv, netstat, ping, ps, rm, rmdir, rpm, sleep, sort, su, tar, touch, umount, uname, vi

/sbin

arp, chkconfig, debugfs, dhclient, dmsg, dump, e2label, fdisk, grub, halt, ifconfig, init, insmod, iptables, lsmod, lspci, mingetty, mkfs, partprobe, portmap, quotaon, quotaoff, restore, rmmod, route, service, shutdown, tune2fs

/usr

- **bin/**
  - *at, bc, cal, cancel, clear, crontab, fdformat, file, find, finger, head, id, info, less, lp/lpr, lpstat, make, man, mesg, mozilla, passwd, quota, scp, spell, ssh, sudo, tail, tee, telnet, wc, who, write, xxd*
- **sbin/**
  - *crond, cupsd, kudzu, pppd, sendmail, sshd, traceroute, useradd, usermod, userdel, xinetd*
- **X11R6/**
  - bin/
    - *startx, twm, X, xclock, xinit, xsetroot, xwd*

/var

- dmesg

/home

- **rsimms/**
  - .bash_profile
  - .bashrc

/root

- .bash_profile
- .bashrc

/mnt

- **cdrom/**
- **floppy/**

/opt

/tmp

- **ssh-XXjXuIH9/**
  - agent.13695

/dev

- hda
- hda1
- had2
- tty1

/boot

- **grub/**
  - grub.conf
- initrd-2.4.20-6.img
- vmlinuz-2.4.20-6

/lib

- **modules/**
  - **2.4.20-6/**
    - **kernel/**
      - **drivers/**
        - **net/**
          - 3c59x.o

/proc

- interupts
- ioports
- modules
- **sys/**
  - **net/**
    - **ipv4/**
      - ip_forward

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

Example GNU/Linux Directory Structure

/

CIS 130 files, directories, commands

/etc

- fstab
- group
- hosts
- hosts.allow
- hosts.deny
- inittab
- issue
- modules.conf
- motd
- mtab
- **pam.d/**
  - login
- passwd
- profile
- **rc.d/**
  - *rc*
  - **rc0.d/**
  - **rc1.d/**
  - **rc2.d/**
  - **rc3.d/**
  - **rc4.d/**
  - **rc5.d/**
  - **rc6.d/**
  - *rc.sysinit*
- resolv.conf
- shadow
- sysctl.conf
- **sysconfig/**
  - network
  - **network-scripts/**
    - ifcfg-eth0
- **xinetd.d/**
  - telnet

/bin

bash, *cat*, *chgrp*, *chmod*, *chown*, *cp*, *cpio*, *date*, *dd*, *df*, *dmesg*, *echo*, *env*, *grep*, *hostname*, *ln*, *ls*, *mail*, *mkdir*, *more*, *mount*, *mv*, *netstat*, *ping*, *ps*, *rm*, *rmdir*, *rpm*, *sleep*, *sort*, *su*, *tar*, *touch*, *umount*, *uname*, *vi*

/sbin

arp, *chkconfig*, *debugfs*, *dhclient*, *dmsg*, *dump*, *e2label*, *fdisk*, *grub*, *halt*, *ifconfig*, *init*, *insmod*, *iptables*, *lsmod*, *lspci*, *mingetty*, *mkfs*, *partprobe*, *portmap*, *quotaon*, *quotaoff*, *restore*, *rmmod*, *route*, *service*, *shutdown*, *tune2fs*

/usr

- **bin/**
  - *at*, *bc*, *cal*, *cancel*, *clear*, *crontab*, *fdformat*, *file*, *find*, *finger*, *head*, *id*, *info*, *less*, *lp/lpr*, *lpstat*, *make*, *man*, *mesg*, *mozilla*, *passwd*, *perl*, *quota*, *scp*, *spell*, *ssh*, *sudo*, *tail*, *tee*, *telnet*, *wc*, *who*, *write*, *xxd*
- **sbin/**
  - *crond*, *cupsd*, *kudzu*, *pppd*, *sendmail*, *sshd*, *traceroute*, *useradd*, *usermod*, *userdel*, *xinetd*
- **X11R6/**
  - bin/
    - *startx*, *twm*, *X*, *xclock*, *xinit*, *xsetroot*, *xwd*

/var

- dmesg
- **httpd/**
  - access_log
  - error_log

/home

- **rsimms/**
  - .bash_profile
  - .bashrc

/root

- .bash_profile
- .bashrc

/mnt

- **cdrom/**
- **floppy/**

/opt

/tmp

- **ssh-XXjXuIH9/**
  - agent.13695

/dev

- hda
- hda1
- had2
- tty1

/boot

- **grub/**
  - grub.conf
- initrd-2.4.20-6.img
- vmlinuz-2.4.20-6

/lib

- **modules/**
  - **2.4.20-6/**
    - **kernel/**
      - **drivers/**
        - **net/**
          - 3c59x.o

/proc

- interupts
- ioports
- modules
- **sys/**
  - **net/**
    - **ipv4/**
      - ip_forward

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

/etc

- fstab
- group
- hosts
- hosts.allow
- hosts.deny
- **httpd/**
  - **conf/**
    - httpd.conf
- inittab
- issue
- modules.conf
- motd
- mtab
- **pam.d/**
  - login
- passwd
- profile
- **rc.d/**
  - *rc*
  - **rc0.d/**
  - **rc1.d/**
  - **rc2.d/**
  - **rc3.d/**
  - **rc4.d/**
  - **rc5.d/**
  - **rc6.d/**
  - *rc.sysinit*
- resolv.conf
- shadow
- sysctl.conf
- **sysconfig/**
  - network
  - **network-scripts/**
    - ifcfg-eth0
- **xinetd.d/**
  - telnet

/bin

bash, *cat*, *chgrp*, *chmod*, *chown*, *cp*, *cpio*, *date*, *dd*, *df*, *dmesg*, *echo*, *env*, *grep*, *hostname*, *ln*, *ls*, *mail*, *mkdir*, *more*, *mount*, *mv*, *netstat*, *ping*, *ps*, *rm*, *rmdir*, *rpm*, *sleep*, *sort*, *su*, *tar*, *touch*, *umount*, *uname*, *vi*

/sbin

arp

chkconfig, *debugfs*, *dhclient*, *dmsg*, *dump*, *e2label*, *fdisk*, *grub*, *halt*, *ifconfig*, *init*, *insmod*, *iptables*, *lsmod*, *lspci*, *mingetty*, *mkfs*, *partprobe*, *portmap*, *quotaon*, *quotaoff*, *restore*, *rmmod*, *route*, *service*, *shutdown*

tune2fs

/usr

- **bin/**
  - *at*, *bc*, *cal*, *cancel*, *clear*, *crontab*, *fdformat*, *file*, *find*, *finger*, *gcc*, *head*, *id*, *info*, *less*, *lp/lpr*, *lpstat*, *make*, *man*, *mesg*, *mozilla*
  - *passwd*, *perl*, *quota*, *scp*, *spell*, *ssh*, *sudo*, *tail*, *tee*, *telnet*, *wc*, *who*, *write*, *xxd*
- **sbin/**
  - *crond*, *cupsd*, *httpd*, *kudzu*, *pppd*, *sendmail*, *sshd*, *traceroute*, *useradd*, *usermod*, *userdel*, *xinetd*
- **X11R6/**
  - bin/
    - *startx*, *twm*, *X*, *xclock*, *xinit*, *xsetroot*, *xwd*

/var

- **httpd/**
  - access_log
  - error_log

/home

- **rsimms/**
- .bash_profile
- .bashrc

/root

- .bash_profile
- .bashrc

/mnt

- **cdrom/**
- **floppy/**

/opt

/tmp

- **ssh-XXjXuIH9/**
  - agent.13695

/dev

- hda
- hda1
- had2
- tty1

/boot

- **grub/**
  - grub.conf
- initrd-2.4.20-6.img
- vmlinuz-2.4.20-6

/lib

- **modules/**
  - **2.4.20-6/**
    - **kernel/**
      - **drivers/**
        - **net/**
          - 3c59x.o

/proc

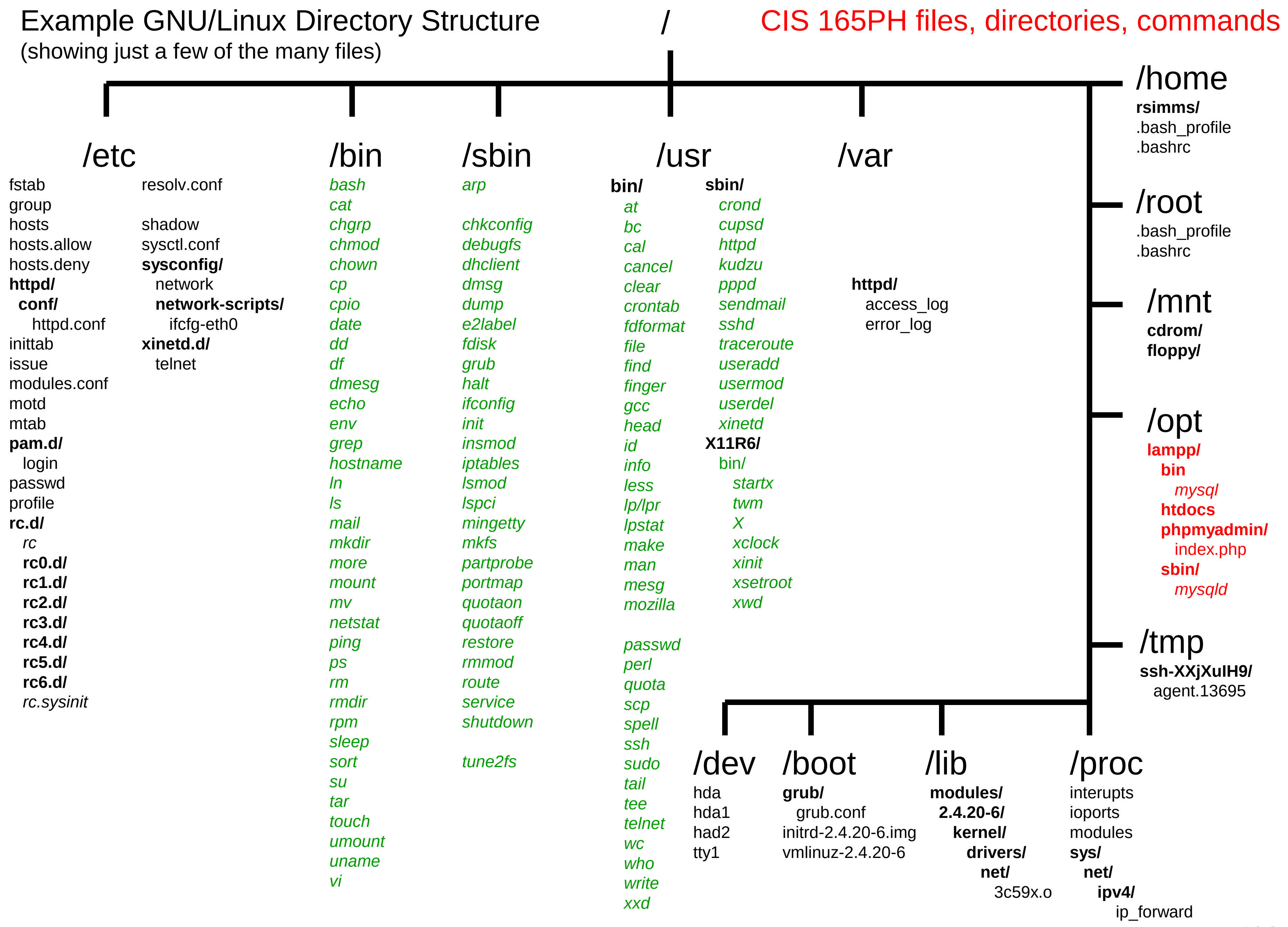
- interupts
- ioports
- modules
- **sys/**
  - **net/**
    - **ipv4/**
      - ip_forward

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



```
/

/etc
fstab                resolv.conf
group
hosts                shadow
hosts.allow          sysctl.conf
hosts.deny           sysconfig/
httpd/                 network
  conf/                network-scripts/
    httpd.conf           ifcfg-eth0
inittab              xinetd.d/
issue                  telnet
modules.conf
motd
mtab
pam.d/
  login
passwd
profile
rc.d/
  rc
  rc0.d/
  rc1.d/
  rc2.d/
  rc3.d/
  rc4.d/
  rc5.d/
  rc6.d/
  rc.sysinit

/bin
bash cat chgrp chmod chown cp cpio date dd df dmesg echo env grep
hostname ln ls mail mkdir more mount mv netstat ping ps rm rmdir rpm
sleep sort su tar touch umount uname vi

/sbin
arp
chkconfig debugfs dhclient dmsg dump e2label fdisk grub halt ifconfig
init insmod iptables lsmod lspci mingetty mkfs partprobe portmap
quotaon quotaoff restore rmmod route service shutdown
tune2fs

/usr
bin/                 sbin/
  at                   crond
  bc                   cupsd
  cal                  httpd
  cancel               kudzu
  clear                pppd
  crontab              sendmail
  fdformat             sshd
  file                 traceroute
  find                 useradd
  finger               usermod
  gcc                  userdel
  head                 xinetd
  id                 X11R6/
  info                 bin/
  less                   startx
  lp/lpr                 twm
  lpstat                 X
  make                   xclock
  man                    xinit
  mesg                   xsetroot
  mozilla                xwd

  passwd
  perl
  quota
  scp
  spell
  ssh
  sudo
  tail
  tee
  telnet
  wc
  who
  write
  xxd

/var
httpd/
  access_log
  error_log

/home
rsimms/
.bash_profile
.bashrc

/root
.bash_profile
.bashrc

/mnt
cdrom/
floppy/

/opt
lampp/
  bin
    mysql
  htdocs
  phpmyadmin/
    index.php
  sbin/
    mysqld

/tmp
ssh-XXjXuIH9/
  agent.13695

/dev
hda
hda1
had2
tty1

/boot
grub/
  grub.conf
initrd-2.4.20-6.img
vmlinuz-2.4.20-6

/lib
modules/
  2.4.20-6/
    kernel/
      drivers/
        net/
          3c59x.o

/proc
interupts
ioports
modules
sys/
  net/
    ipv4/
      ip_forward
```

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

```
/etc
fstab               resolv.conf
group               securetty
hosts               shadow
hosts.allow         sysctl.conf
hosts.deny          sysconfig/
httpd/                network
  conf/               network-scripts/
    httpd.conf          ifcfg-eth0
inittab             xinetd.d/
issue                 telnet
modules.conf
motd
mtab
pam.d/
  login
passwd
profile
rc.d/
  rc
  rc0.d/
  rc1.d/
  rc2.d/
  rc3.d/
  rc4.d/
  rc5.d/
  rc6.d/
  rc.sysinit
```

```
/bin
bash
cat
chgrp
chmod
chown
cp
cpio
date
dd
df
dmesg
echo
env
grep
hostname
ln
ls
mail
mkdir
more
mount
mv
netstat
ping
ps
rm
rmdir
rpm
sleep
sort
su
tar
touch
umount
uname
vi
```

```
/sbin
arp
bastille
chkconfig
debugfs
dhclient
dmsg
dump
e2label
fdisk
grub
halt
ifconfig
init
insmod
iptables
lsmod
lspci
mingetty
mkfs
partprobe
portmap
quotaon
quotaoff
restore
rmmod
route
service
shutdown
tripwire
tune2fs
```

```
/usr
bin/          sbin/
  at            crond
  bc            cupsd
  cal           httpd
  cancel        kudzu
  clear         pppd
  crontab       sendmail
  fdformat      sshd
  file          traceroute
  find          useradd
  finger        usermod
  gcc           userdel
  head          xinetd
  id          X11R6/
  info          bin/
  less            startx
  lp/lpr          twm
  lpstat          X
  make            xclock
  man             xinit
  mesg            xsetroot
  mozilla         xwd
  openssl
  passwd
  perl
  quota
  scp
  spell
  ssh
  sudo
  tail
  tee
  telnet
  wc
  who
  write
  xxd
```

```
/var
log/
  Bastille/
    Assessment/
      assessment-report.html
  dmesg
  httpd/
    access_log
    error_log
  spool/
    clientmqueue
```

```
/home
rsimms/
.bash_profile
.bashrc
```

```
/root
.bash_profile
.bashrc
```

```
/mnt
cdrom/
floppy/
```

```
/opt
lampp/
  bin
    mysql
  htdocs
  phpmyadmin/
    index.php
  sbin/
    mysqld
```

```
/tmp
ssh-XXjXuIH9/
  agent.13695
```

```
/dev
hda
hda1
had2
tty1
```

```
/boot
grub/
  grub.conf
initrd-2.4.20-6.img
vmlinuz-2.4.20-6
```

```
/lib
modules/
  2.4.20-6/
    kernel/
      drivers/
        net/
          3c59x.o
```

```
/proc
interupts
ioports
modules
sys/
  net/
    ipv4/
      ip_forward
```

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)

/

/etc

```
fstab              resolv.conf
group              securetty
hosts              shadow
hosts.allow        sysctl.conf
hosts.deny         sysconfig/
httpd/               network
  conf/              network-scripts/
    httpd.conf         ifcfg-eth0
inittab            xinetd.d/
issue                telnet
modules.conf
motd
mtab
pam.d/
  login
passwd
profile
rc.d/
  rc
  rc0.d/
  rc1.d/
  rc2.d/
  rc3.d/
  rc4.d/
  rc5.d/
  rc6.d/
  rc.sysinit
```

/bin

```
bash
cat
chgrp
chmod
chown
cp
cpio
date
dd
df
dmesg
echo
env
grep
hostname
ln
ls
mail
mkdir
more
mount
mv
netstat
ping
ps
rm
rmdir
rpm
sleep
sort
su
tar
touch
umount
uname
vi
```

/sbin

```
arp
bastille
chkconfig
debugfs
dhclient
dmsg
dump
e2label
fdisk
grub
halt
ifconfig
init
insmod
iptables
lsmod
lspci
mingetty
mkfs
partprobe
portmap
quotaon
quotaoff
restore
rmmod
route
service
shutdown
tripwire
tune2fs
```

/usr

```
bin/           sbin/
  at             crond
  bc             cupsd
  cal            httpd
  cancel         kudzu
  clear          pppd
  crontab        sendmail
  fdformat       sshd
  file           traceroute
  find           useradd
  finger         usermod
  gcc            userdel
  head           xinetd
  id           X11R6/
  info           bin/
  less             startx
  lp/lpr           twm
  lpstat           X
  make             xclock
  man              xinit
  mesg             xsetroot
  mozilla          xwd
  openssl
  passwd
  perl
  quota
  scp
  spell
  ssh
  sudo
  tail
  tee
  telnet
  wc
  who
  write
  xxd
```

/var

```
log/
  Bastille/
    Assessment/
      assessment-report.html
  dmesg
  httpd/
    access_log
    error_log
spool/
  clientmqueue
```

/home

```
rsimms/
.bash_profile
.bashrc
```

/root

```
.bash_profile
.bashrc
```

/mnt

```
cdrom/
floppy/
```

/opt

```
lampp/
  bin
    mysql
  htdocs
  phpmyadmin/
    index.php
  sbin/
    mysqld
```

/tmp

```
ssh-XXjXuIH9/
  agent.13695
```

/dev

```
hda
hda1
had2
tty1
```

/boot

```
grub/
  grub.conf
initrd-2.4.20-6.img
vmlinuz-2.4.20-6
```

/lib

```
modules/
  2.4.20-6/
    kernel/
      drivers/
        net/
          3c59x.o
```

/proc

```
interupts
ioports
modules
sys/
  net/
    ipv4/
      ip_forward
```

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

Class Field Trip

1) boot
   - The kernel

2) etc
   - Apache web configuration file
   - motd
   - passwd

3) var
   - mail
   - www

4) home
   - Student home directories
   - depot
   - bin