Lesson Module Status

- Slides - draft
- Flash cards –
- properties – done
- page numbers - done
- 1st minute quiz – done
- Web Calendar summary –
- Web book pages - none
- Commands –
- Lab tested – done
- enlightenment script tested - done
- CCC Confer wall paper / quiz - done

CIS 90 - Lesson 4


Dennis

Sean

Christopher

Brian

Francisco

Salena

Abd


Sarah

Astitow

Mike D.

Alex

Christine

Steven

Richie

Nathan


Tony

James G.


Sergio

Anthony

Fernando

Miguel

Lars

Jennifer

Rudy


Laura P.

Nick

Juan

Jacob

Andrew


Luke

Saulius

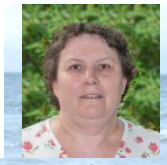

Rich

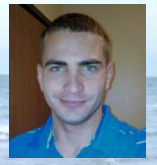
Instructor: **Rich Simms**
Dial-in: **888-450-4821**
Passcode: **761867**

Online Class Students

Edtson


James B.


Liz


Casady

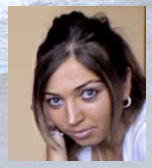
Jason

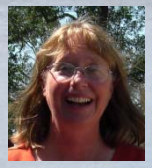
Dale

Aaron


Steve

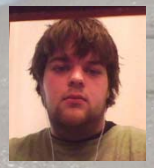

Matt

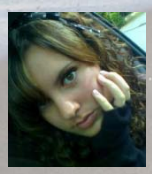

Songul

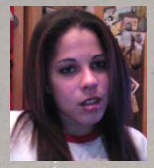

Stephanie


Victor

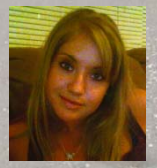
Tanya


Mike P.


Adriana


Laura S.


Olivia


Janelle

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

Quiz

Please close your books, notes, lesson materials, forum and answer these questions in the order shown:

1. What command can you use to "chat" with another user?

2. With /bin/mail, where do all messages that have been read go after using the quit command?

3. How do you forward a message with /bin/mail?

Online students,
email answers to:
risimms@cabrillo.edu

Classroom students,
email answers to:
risimms@cabrillo.edu
(or you can write your answers on a scrap piece of paper and hand it in)

[ ] Has the phone bridge been added?

[ ] Is recording on?

[ ] Does the phone bridge have the mike?

[ ] Share slides, putty (rsimms, simmsben, roddyduk), Chrome

[ ] Disable spelling on PowerPoint

The UNIX/Linux File System

Objectives	Agenda
• 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.	• Quiz • Questions from last week • The UNIX Directory Hierarchy • Navigating the file system* • File types • Viewing files • Exercise: Enlightenment • Wrap up

✵ = hands on exercise for topic

Previous material and assignment

1. Questions on previous material?

2. More Lab 3 questions?
   - I'll use check3 for grading
   - bash shell vs mail "shell"
   - Clean up duplicates before last submittal
   - mail $(ls /home/cis90)
   - mail -f, mail -f mbox, mail -f uhistory

Lab 2 Results

1) show shell	
2) type commands	xxx
3) echo variables	x
4) set TERM	xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx x
5) upper/lower case	xxxxx
6) who –g	xx
7) number of arguments	xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx
8) CR and quotes	xxxxxx
9) ; to separate commands	xxxxxxxxxx xx
10) change password	
11) uname options	xxxxxxxxxx xxxxxxxxxx xxxxxxx
12) banner	xxxxxxxxxx xx
13) finger	xxxxxxx
14) id	x
15) man	xx
16) whatis vs man –f	xxxxxxxxxx xxxx
17) tryme	
18) who –q	xxxxxxxxxx xxxxxxxxxx xx
19) man –k vs apropos	xxxxxxxxxx xxxxxxxxxx xx
20) info bash	xxx
21) google	
22) sqrt	xxxxxxxx
Q1 – input from cmd line	xxxxxxxxxx
Q2 – input from keyboard	xxxxxxx
Q3 – input from OS	xxxxx

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

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

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

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

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

banner I am fine
banner "I am fine"

Lab 2 Results

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

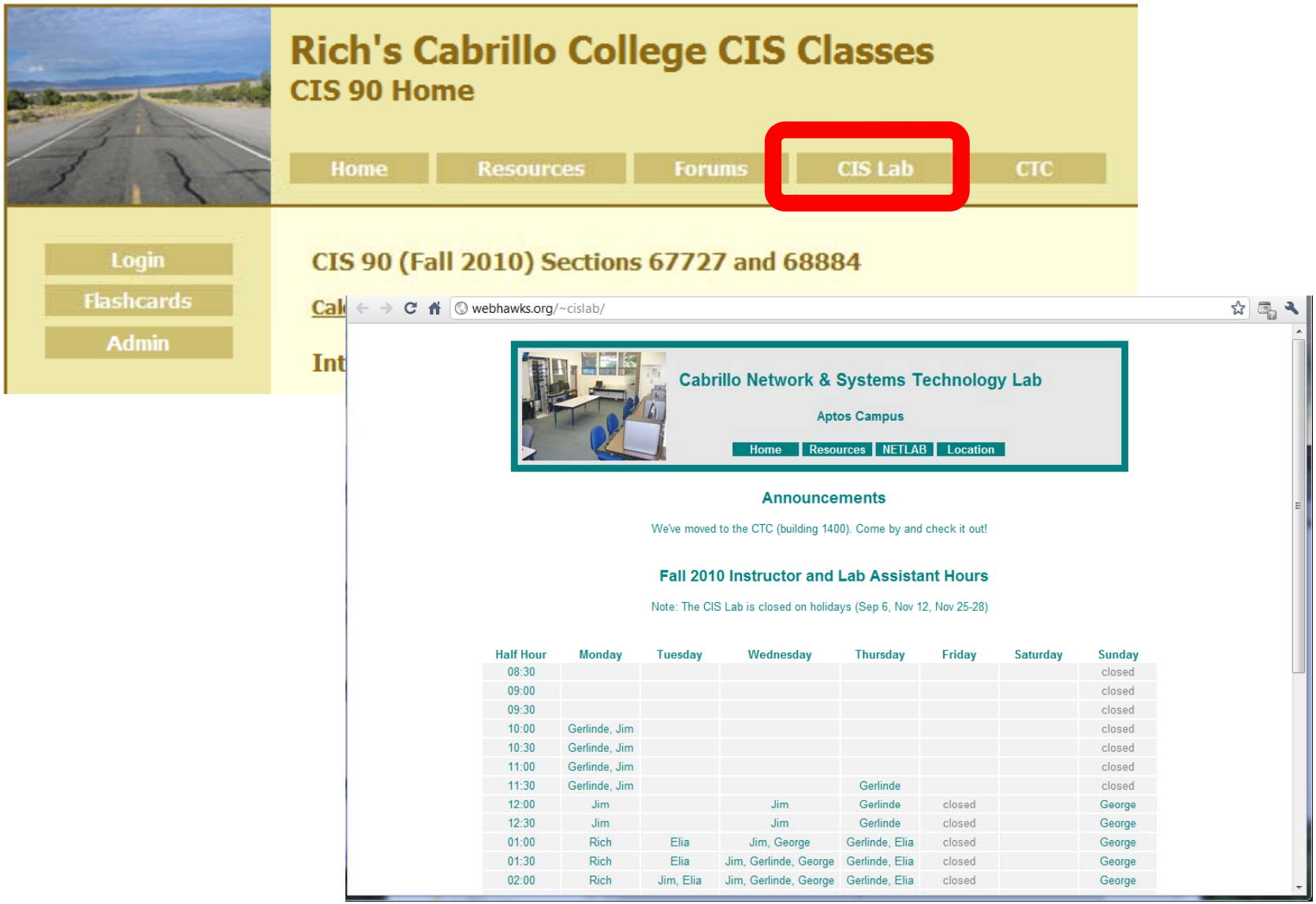
18. Use the manual pages, and the **who** command, to find out the number of users logged on.

Lab 2 Results

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.

Housekeeping

- MSDN AA - is it working?
- Graded labs placed in your home directory
- Answers to labs in /home/cis90/answers/ directory
- Lab 3 and five posts due tonight



Half Hour	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
08:30							closed
09:00							closed
09:30							closed
10:00	Gerlinde, Jim						closed
10:30	Gerlinde, Jim						closed
11:00	Gerlinde, Jim						closed
11:30	Gerlinde, Jim			Gerlinde			closed
12:00	Jim		Jim	Gerlinde	closed		George
12:30	Jim		Jim	Gerlinde	closed		George
01:00	Rich	Elia	Jim, George	Gerlinde, Elia	closed		George
01:30	Rich	Elia	Jim, Gerlinde, George	Gerlinde, Elia	closed		George
02:00	Rich	Jim, Elia	Jim, Gerlinde, George	Gerlinde, Elia	closed		George

My Lab Hours just for Thur 9/23 1-3:30

This week only

Grant Survey

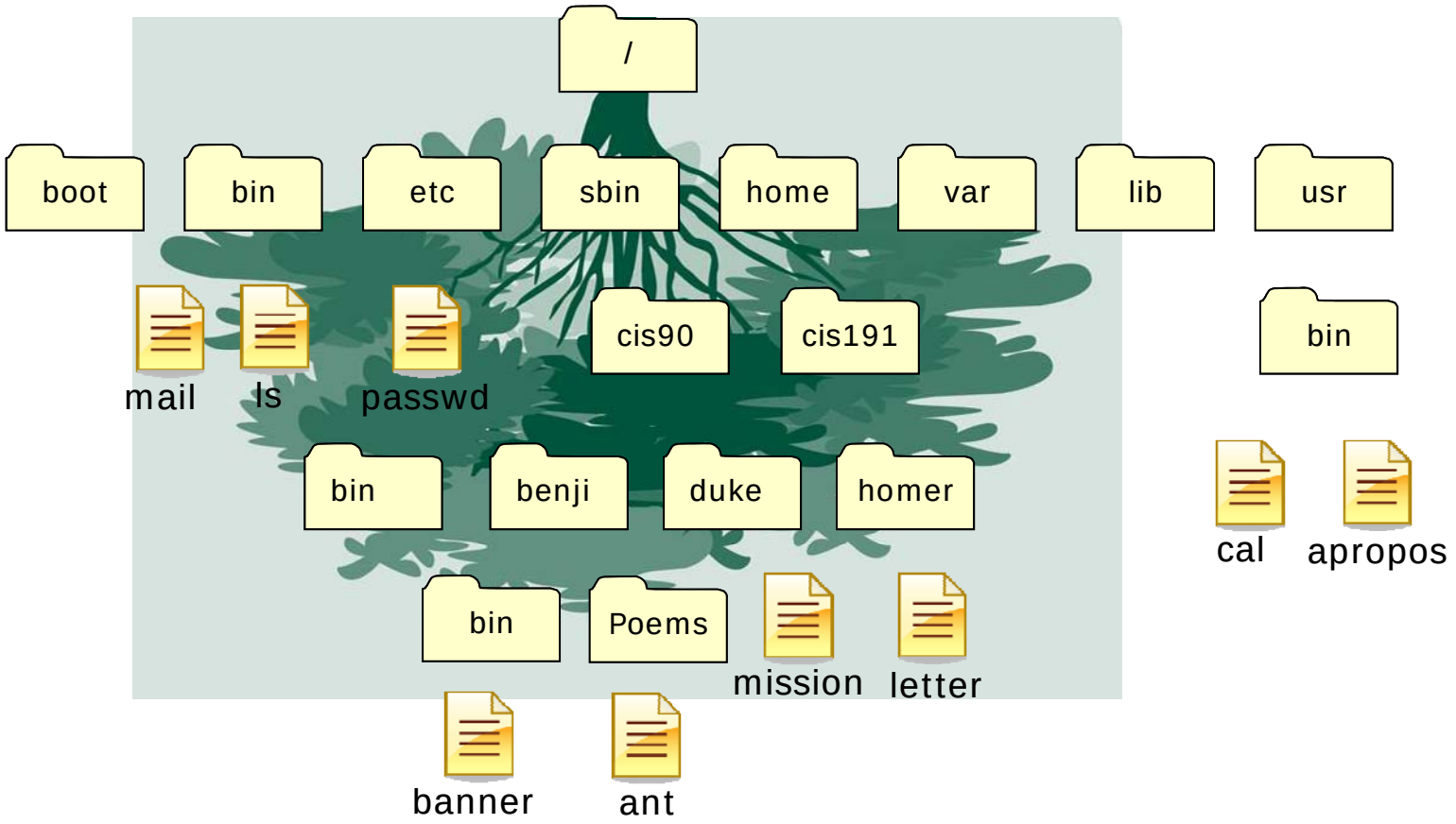
The UNIX Directory Hierarchy

UNIX File Tree

/ = root of the tree

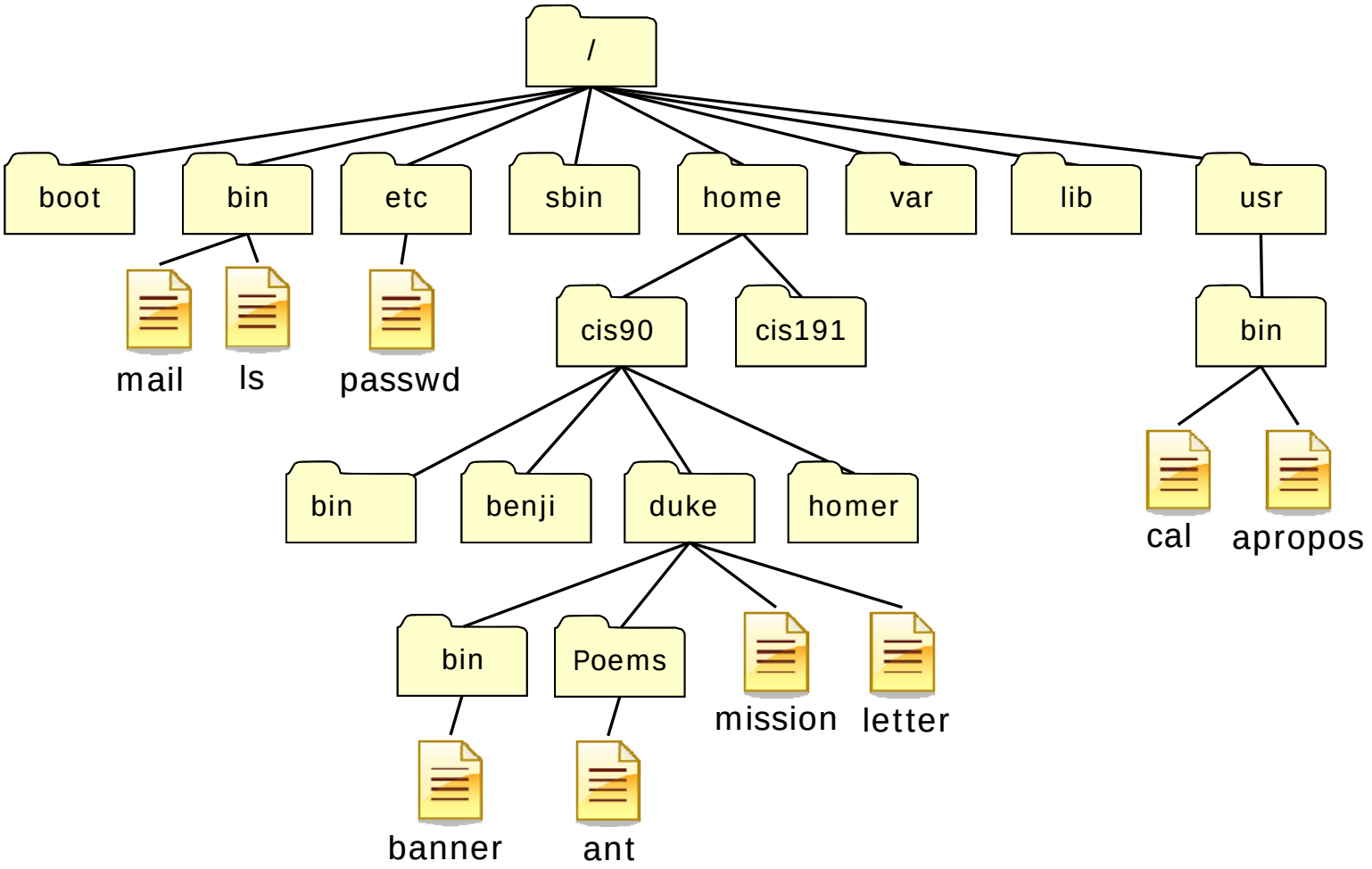
UNIX File Tree

/ = root of the tree



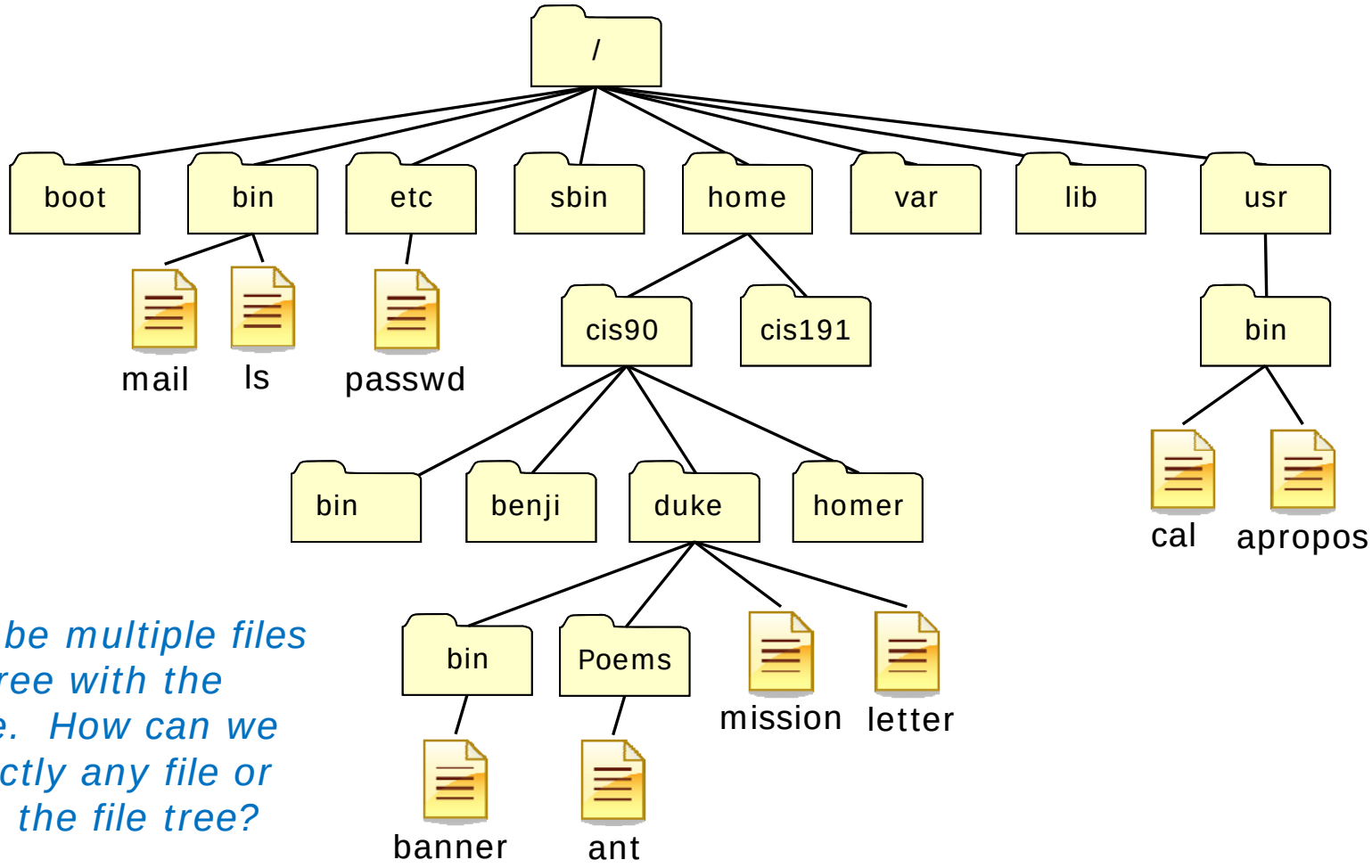
UNIX File Tree

/ = root of the tree



UNIX File Tree

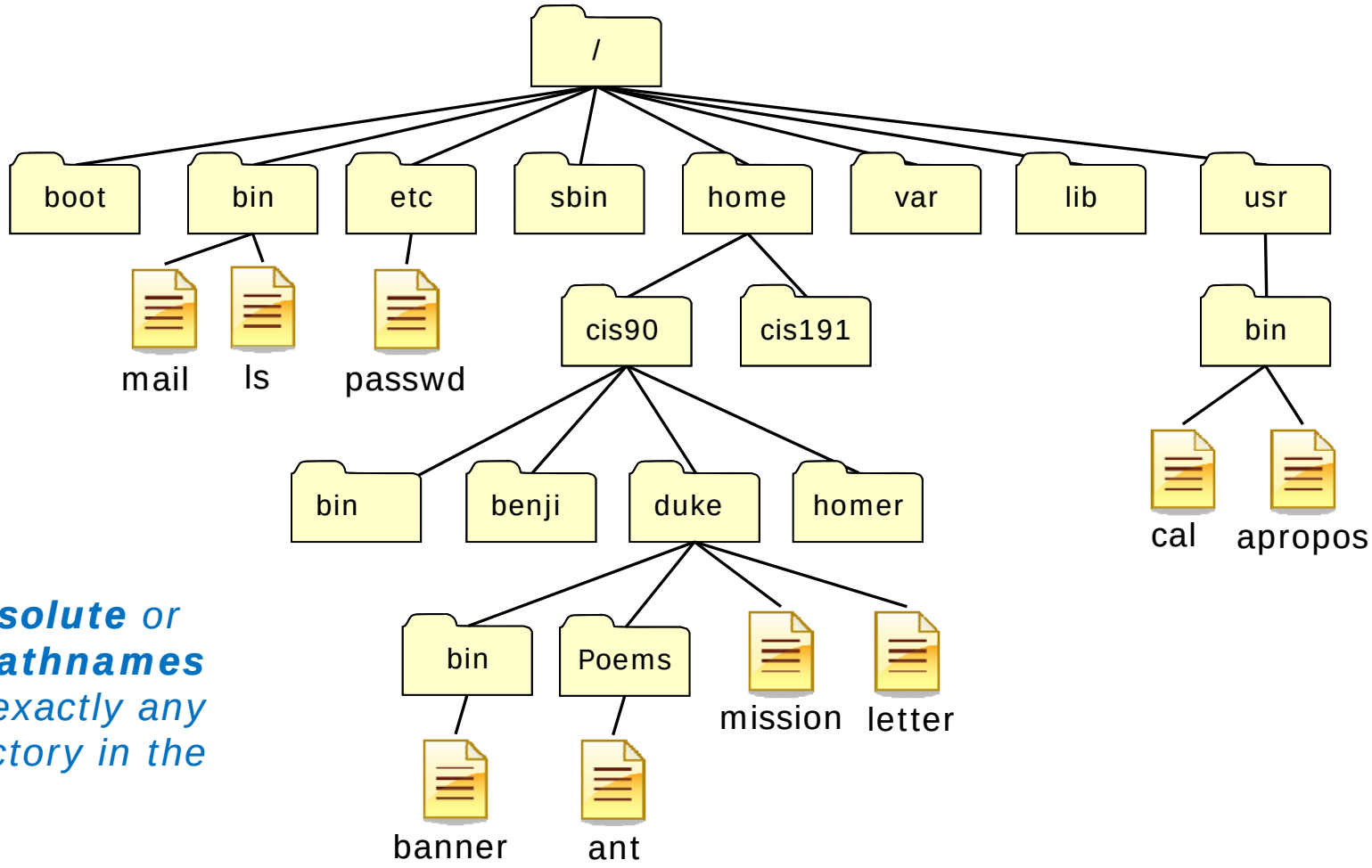
/ = root of the tree



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

UNIX File Tree

/ = root of the tree



*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 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/duke/Poems/ant	*(file)*
/boot	*(directory)*
/usr/bin/cal	*(file)*
/home/cis90/bin	*(directory)*
/bin/mail	*(file)*

Notice they all start with the /

Absolute Pathnames

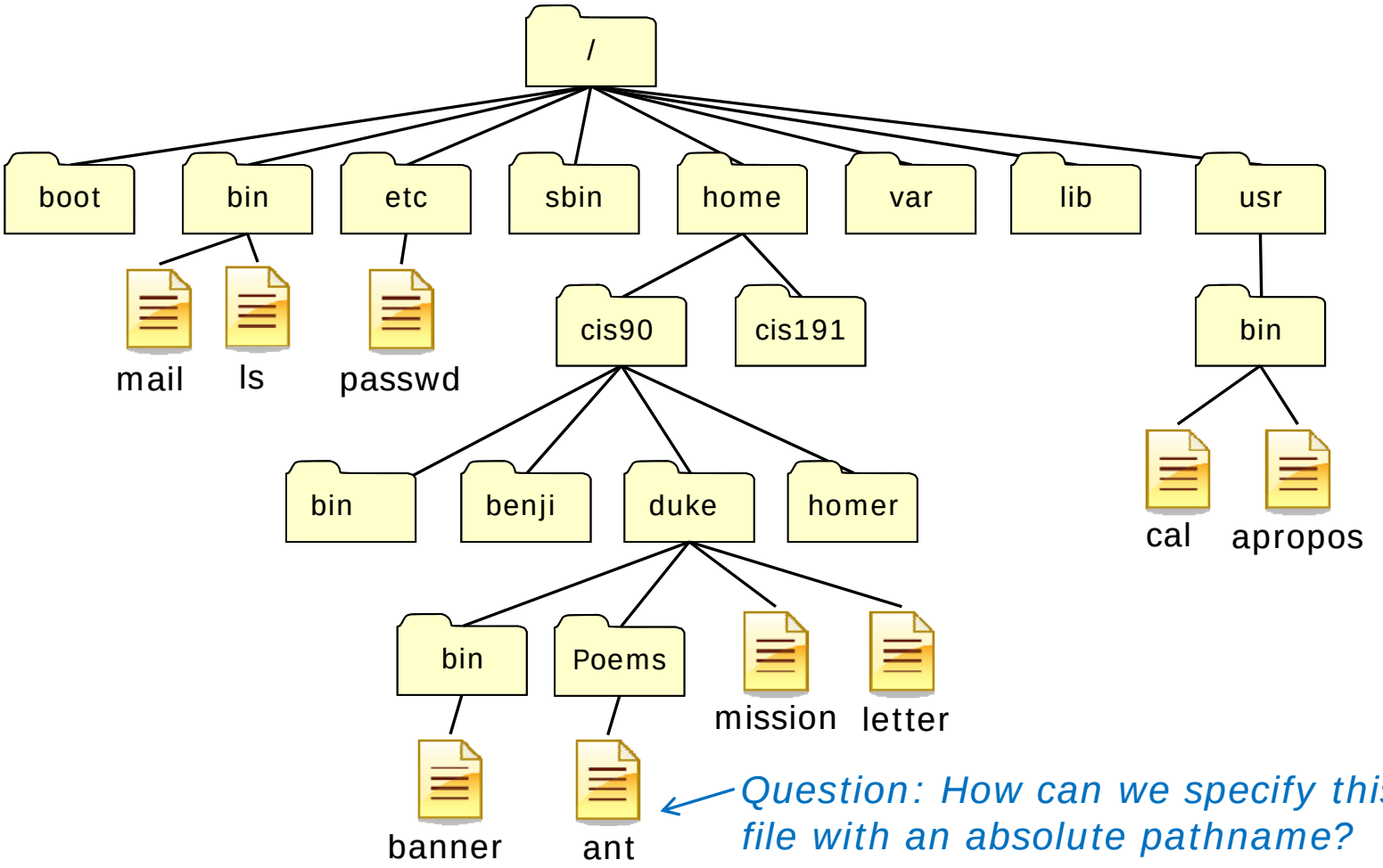
Using absolute pathnames as command arguments

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

Examples of use as command arguments:

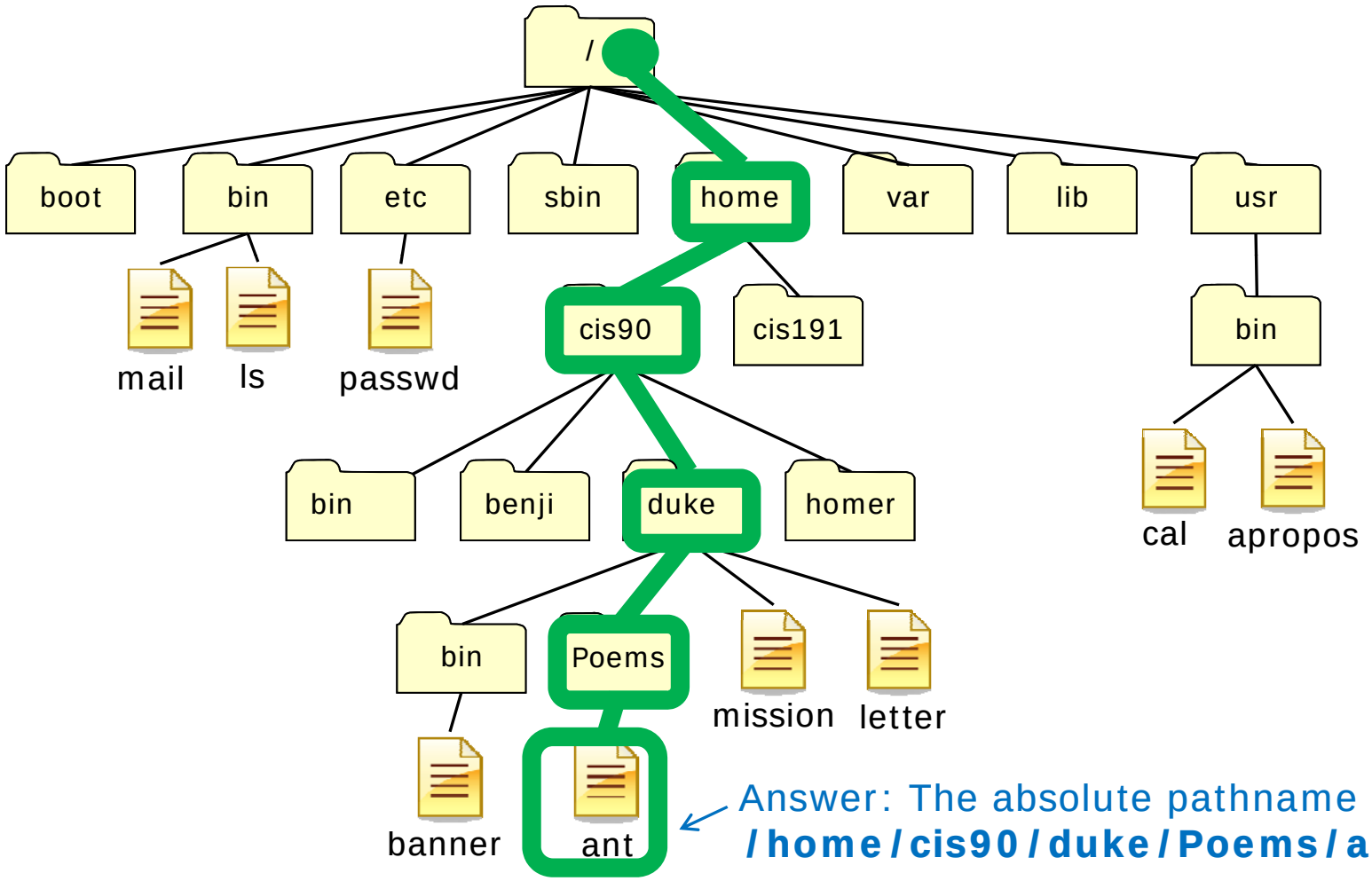
```
ls  /bin  /sbin  /usr/bin  /usr/sbin
file  /usr/bin/cal
cd  /home/cis90/bin
ls  -l  /bin/mail
cp  /etc/passwd  /home/cis90/simmsben/
```

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



Question: How can we specify this file with an absolute pathname?

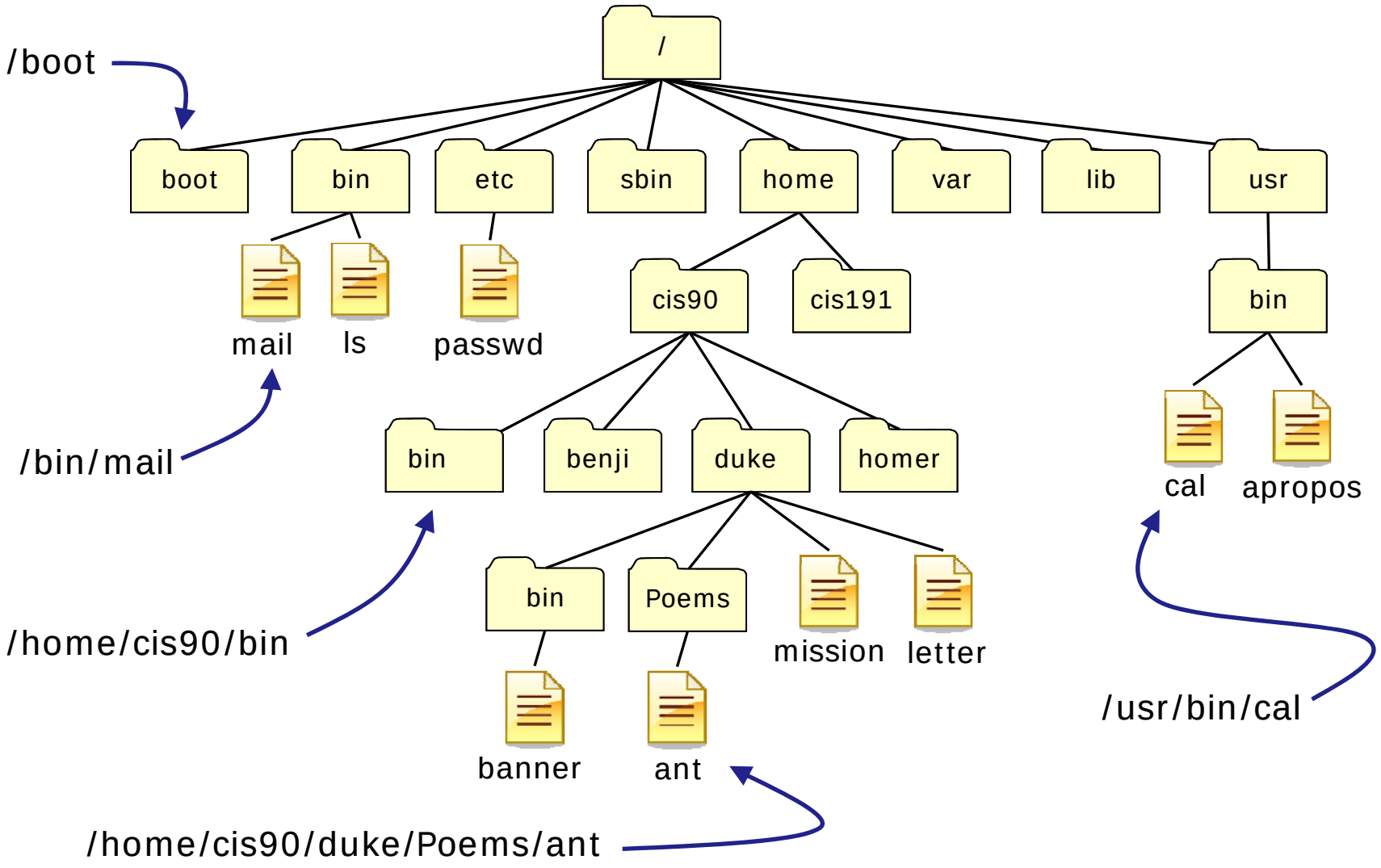
An **absolute pathname** specifies the path from the top of the tree to the target directory or file



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

Absolute Pathnames

These are all absolute pathname examples



Relative Pathnames

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

Examples:

ant	*(file)*
Poems/Shakespeare/sonnet5	*(file)*
../mission	*(file)*
../bin/	*(directory)*

Relative Pathnames

Using relative pathnames as command arguments

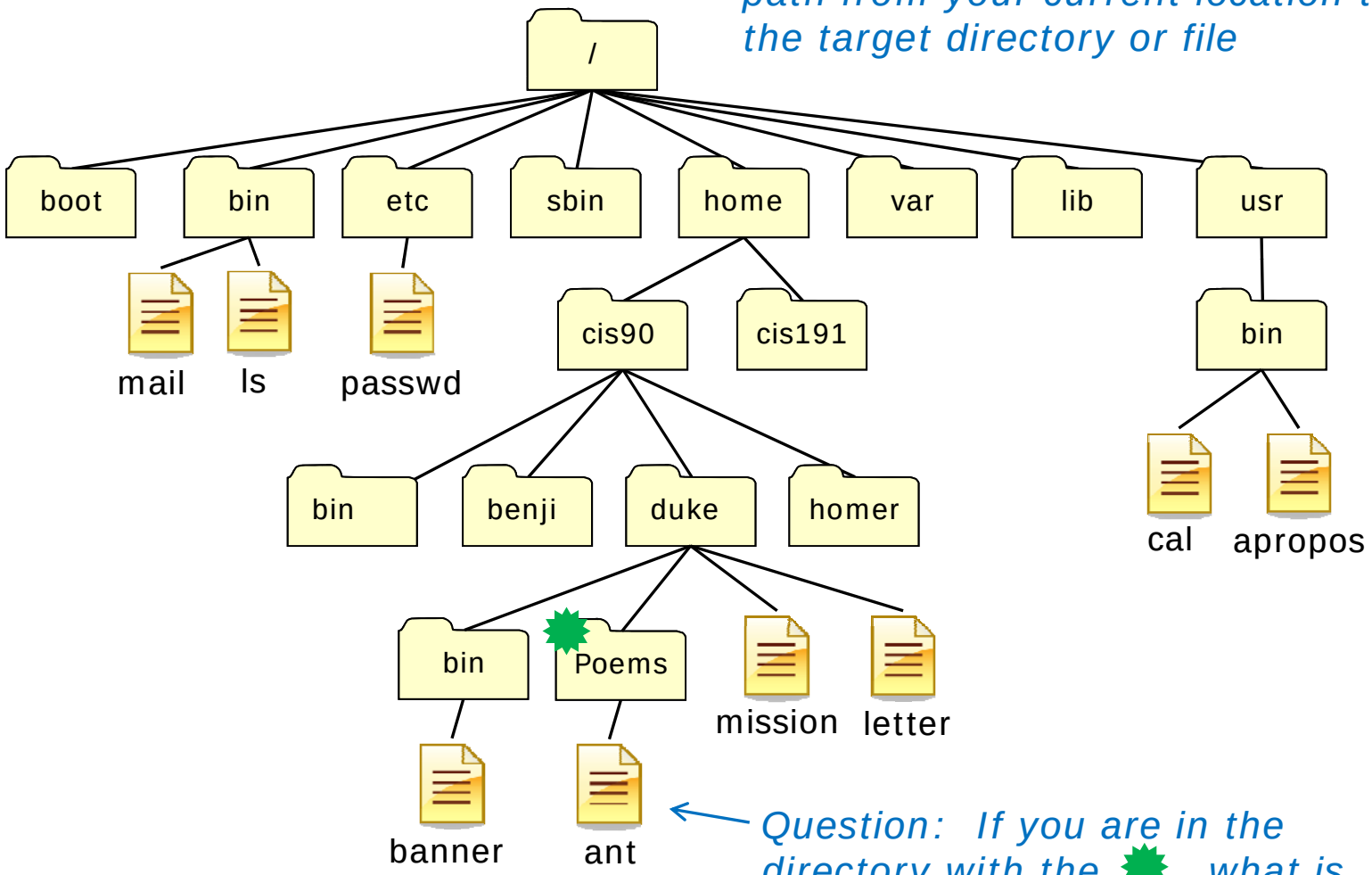
*A **relative pathname** specifies the path from your current location to the target directory or file.*

Examples of use as command arguments:

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

Relative Pathname Example

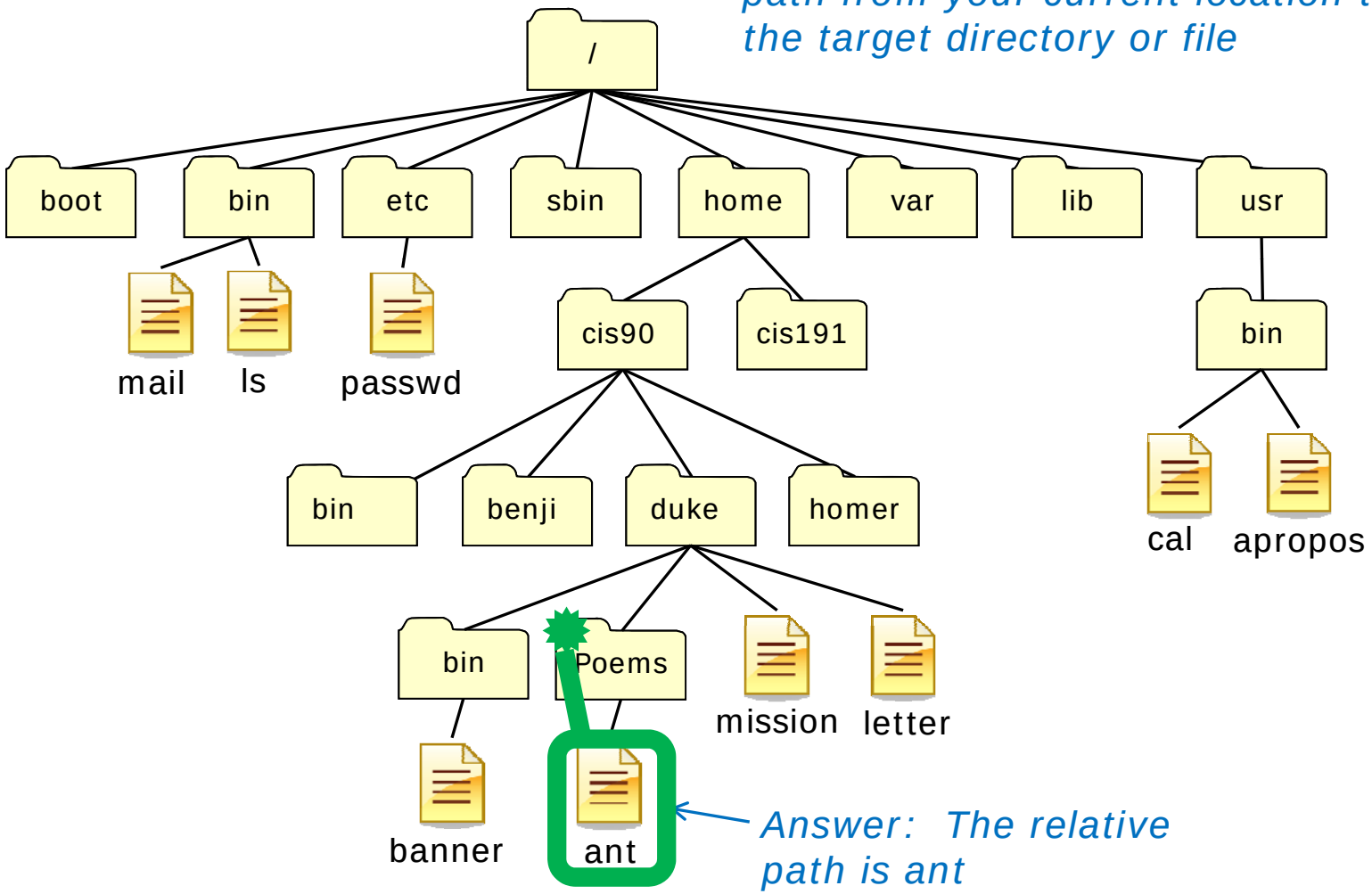
*A **relative pathname** specifies the path from your current location to the target directory or file*



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

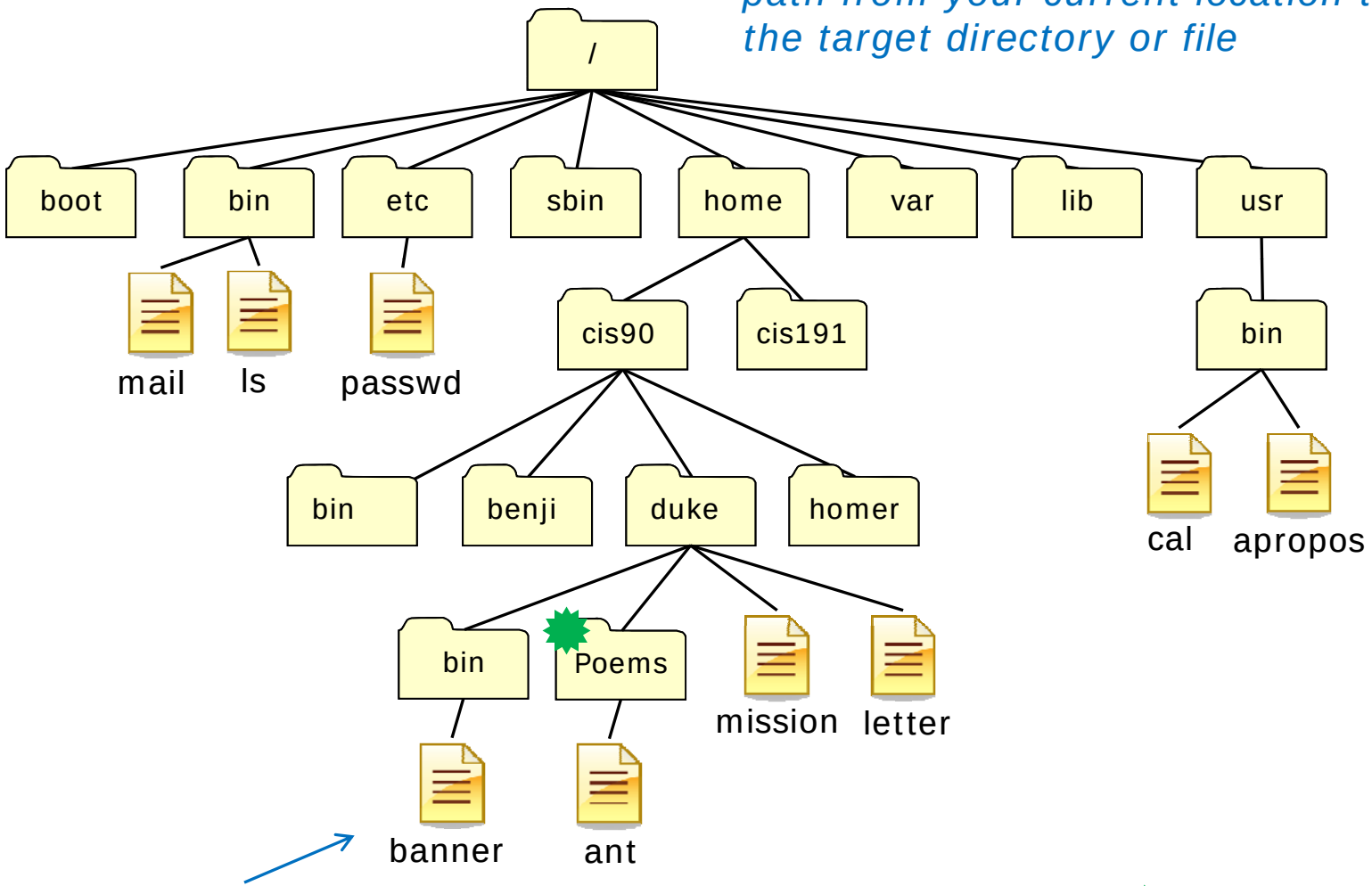
Relative Pathname Example

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



Relative Pathname Example

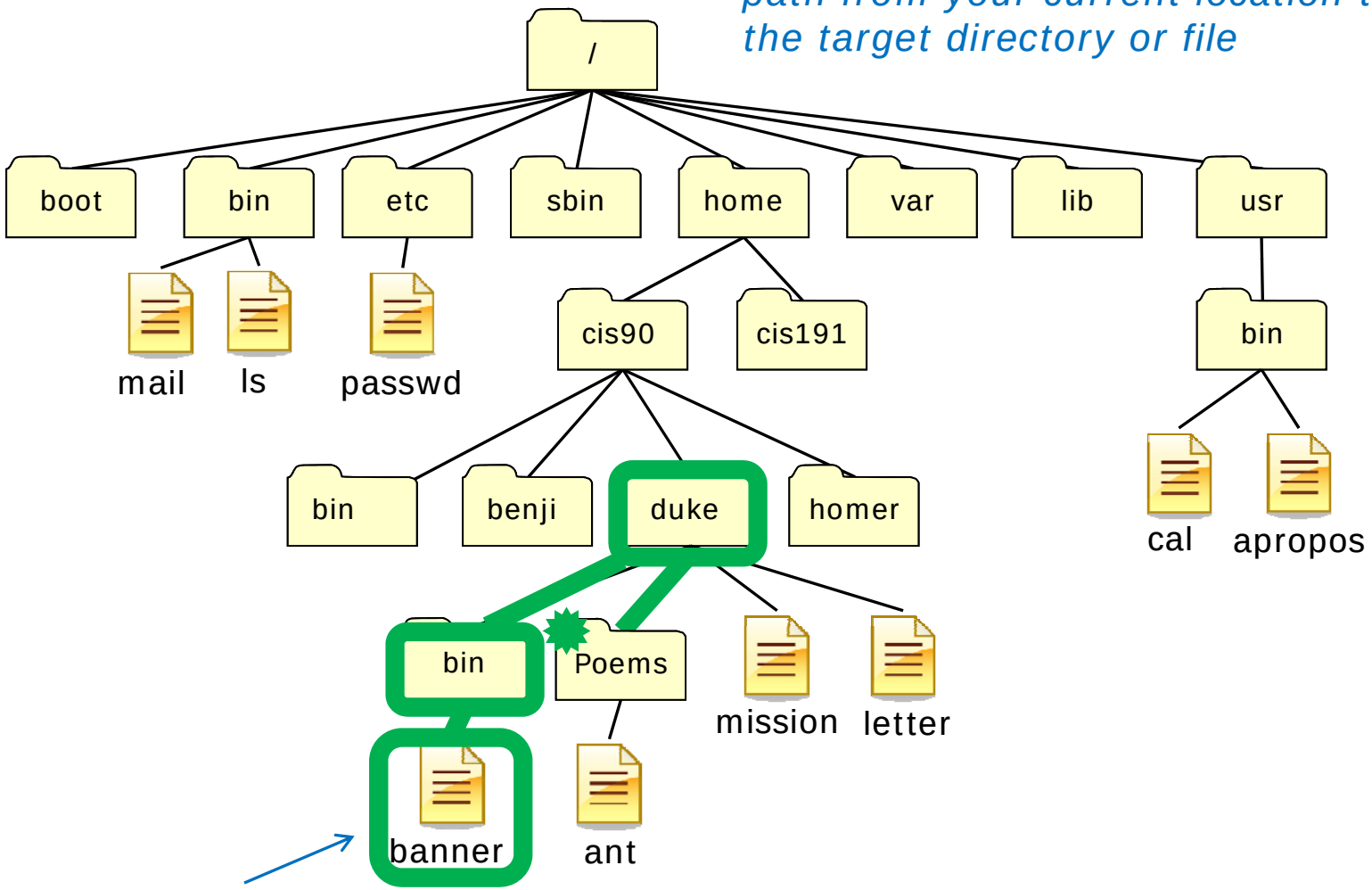
*A **relative pathname** specifies the path from your current location to the target directory or file*



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

Relative Pathname Example

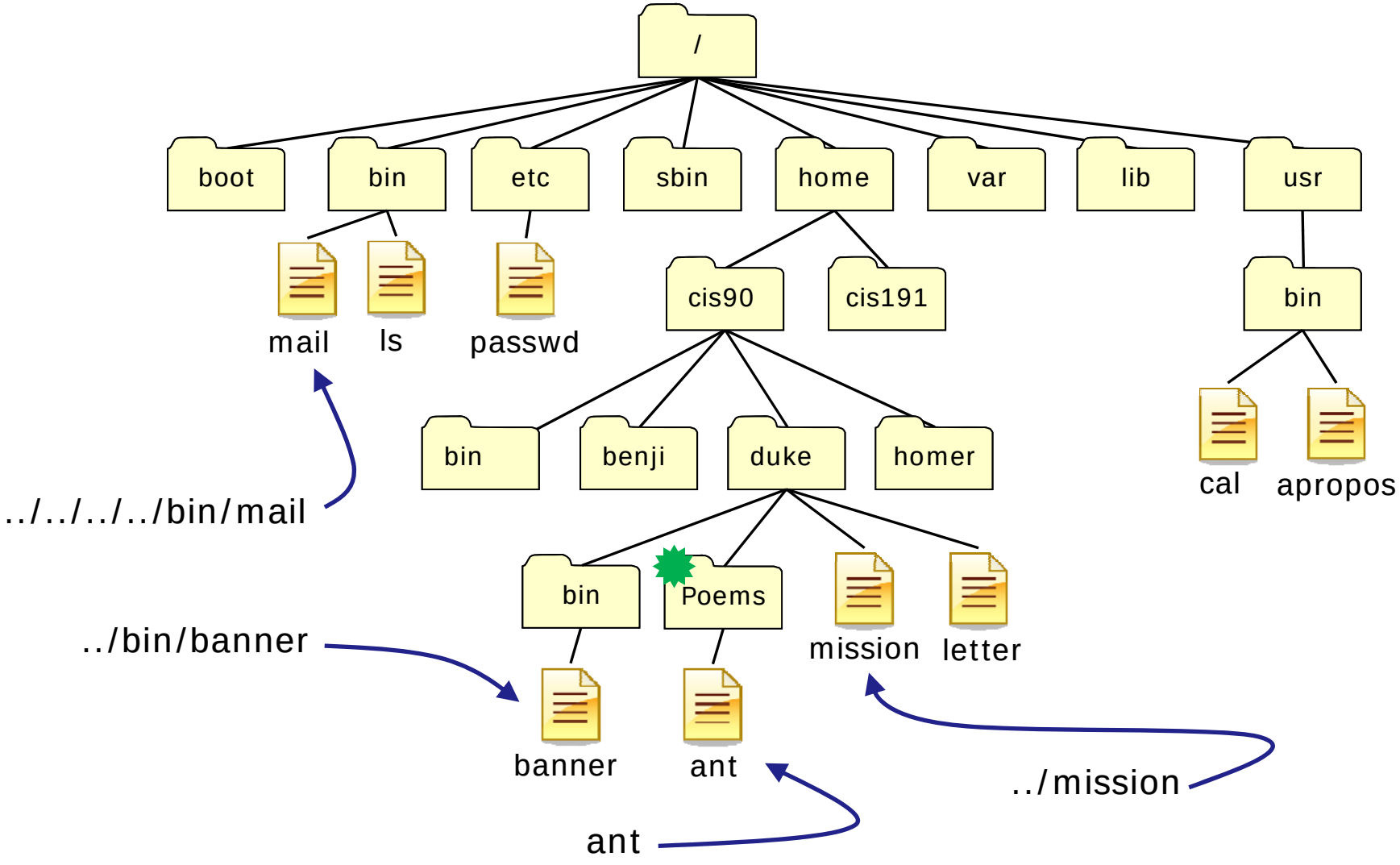
*A **relative pathname** specifies the path from your current location to the target directory or file*



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

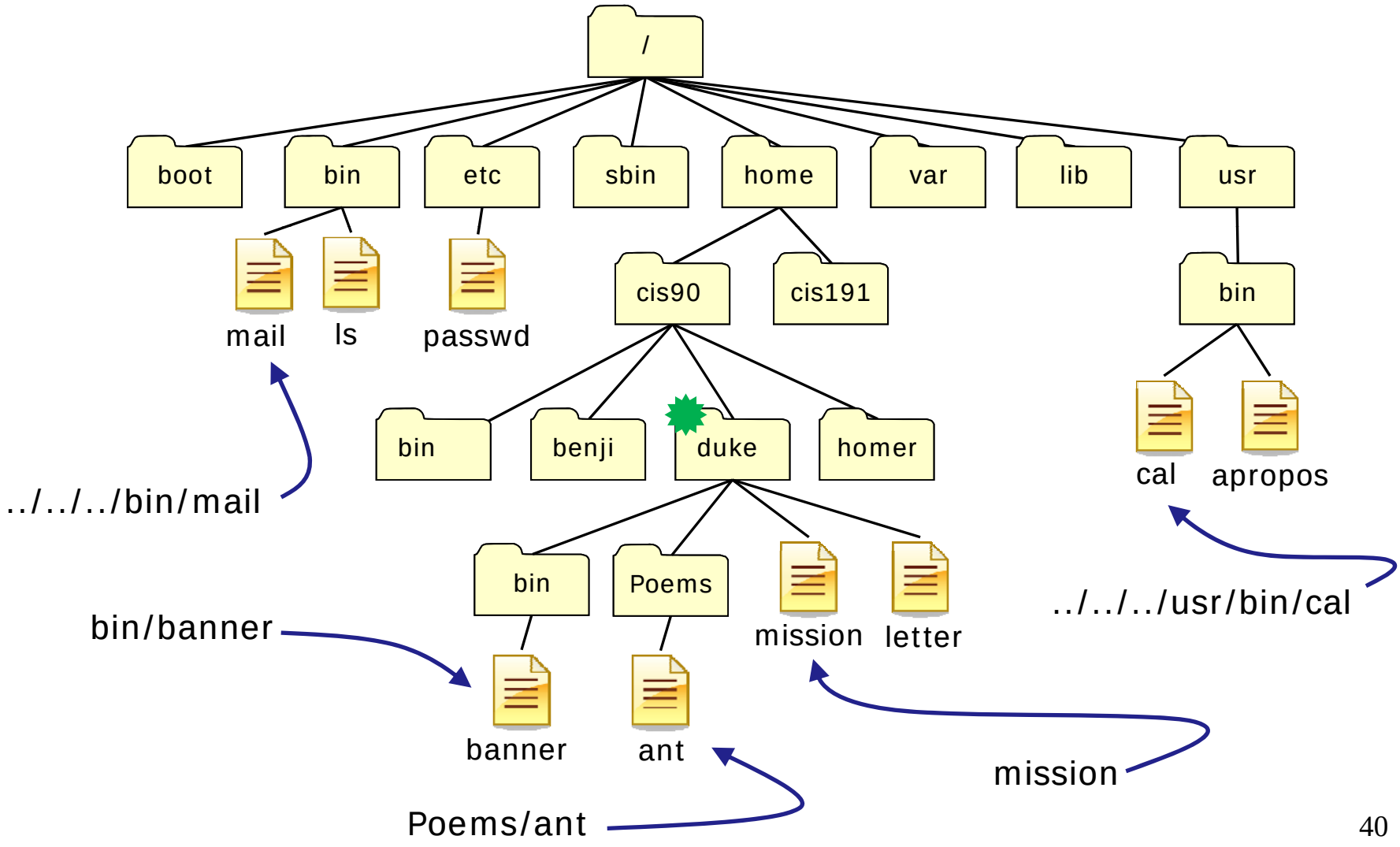
Relative Pathnames

Names that start relative to the current working directory (✹)



Relative Pathnames

Names that start relative to the current working directory (✹)



Class Exercise

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

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

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

  ls /etc/passwd

- List the letter file using a relative pathname

  ls letter

- List the letter file using an absolute pathname

  ls /home/cis90/simmsben/letter

user your home directory instead

Heads up on a future test question

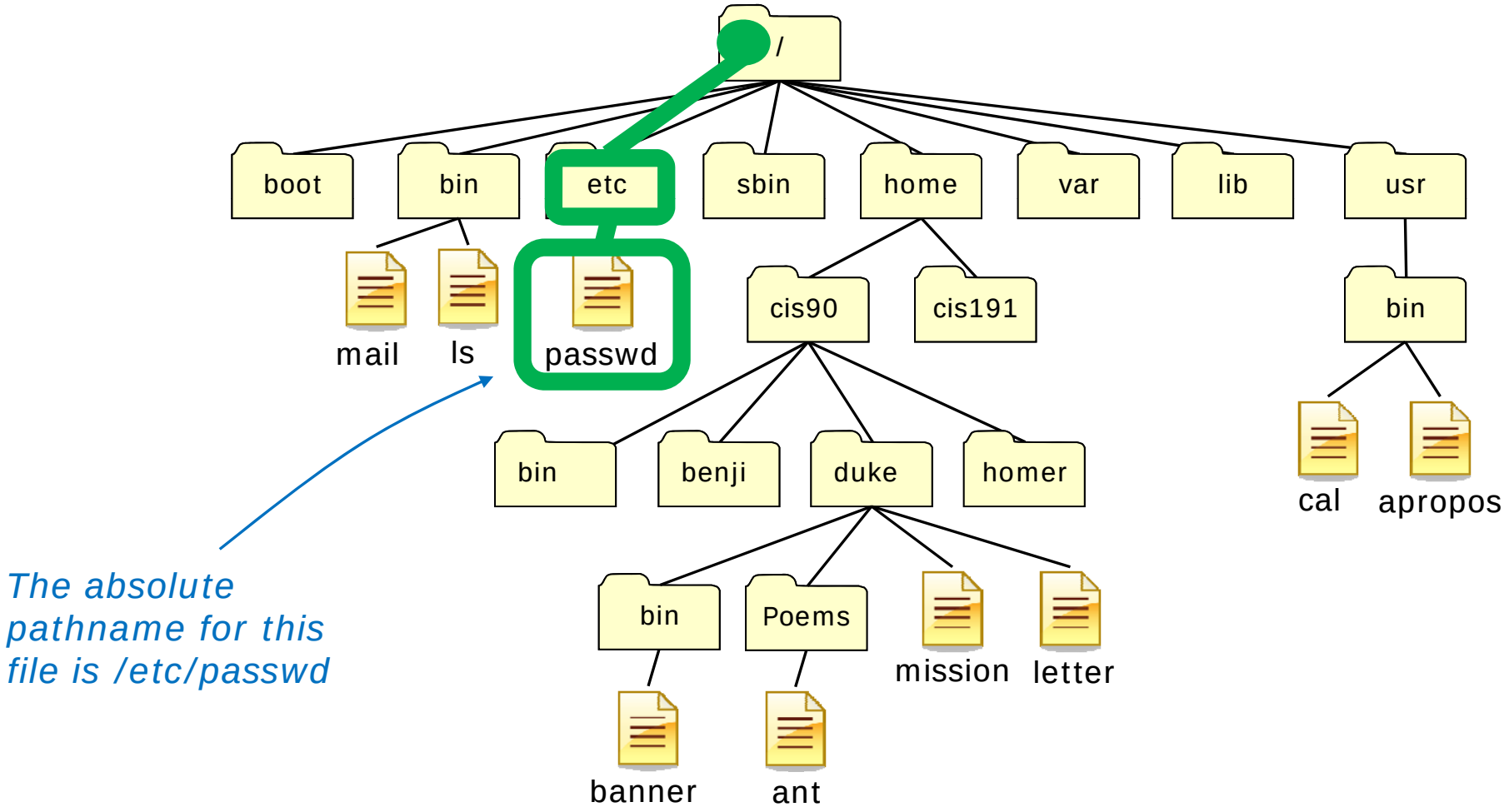
Question: What is the absolute pathname of /etc/passwd?

Answer: /etc/passwd

What is the color of Washington's white horse?

UNIX File Tree

/ = root of the tree



UNIX File Hierarchy

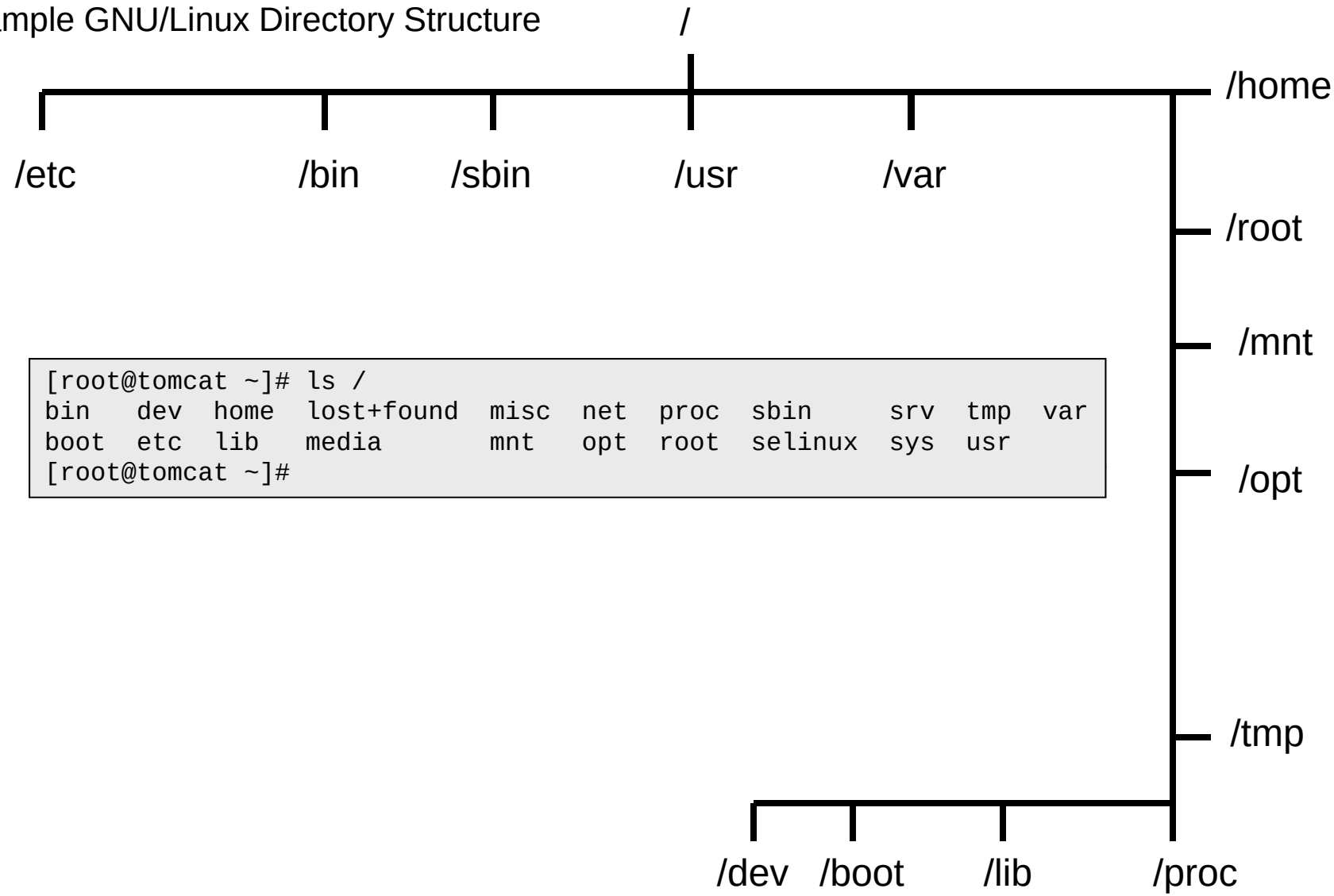
/
/bin
/boot
/dev
/etc
/home
/lib
/lost+found
/mnt
/opt
/proc
/root
/sbin
/tmp
/usr

The UNIX/Linux File System Hierarchy

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

Directory	Contents
/bin	binary files forming the commands and shells used by the system administrator and users
/boot	files used during the initial boot-up process including the kernel
/dev	device files 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

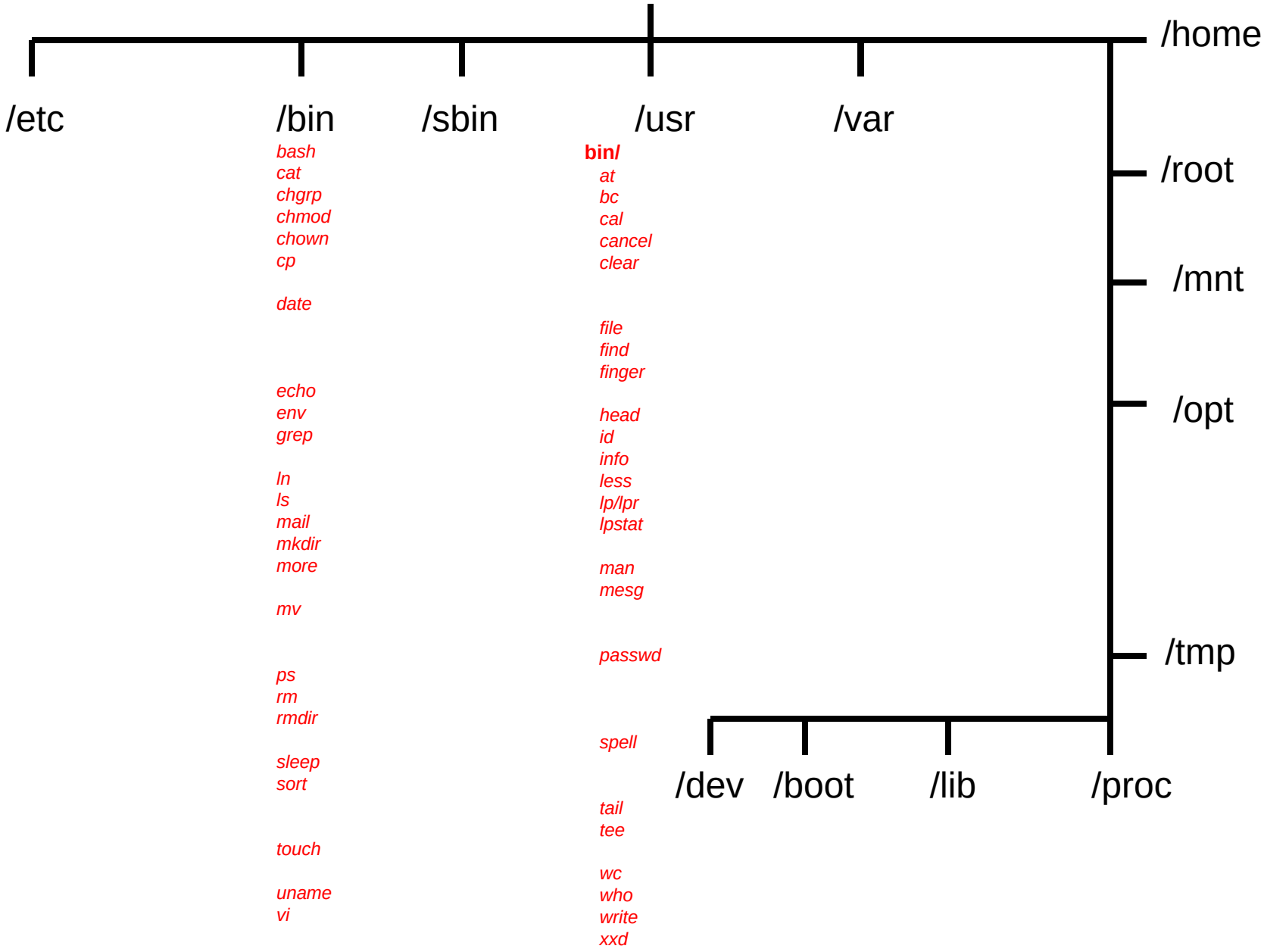
Example GNU/Linux Directory Structure




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

Example GNU/Linux Directory Structure

CIS 90 files, directories, commands

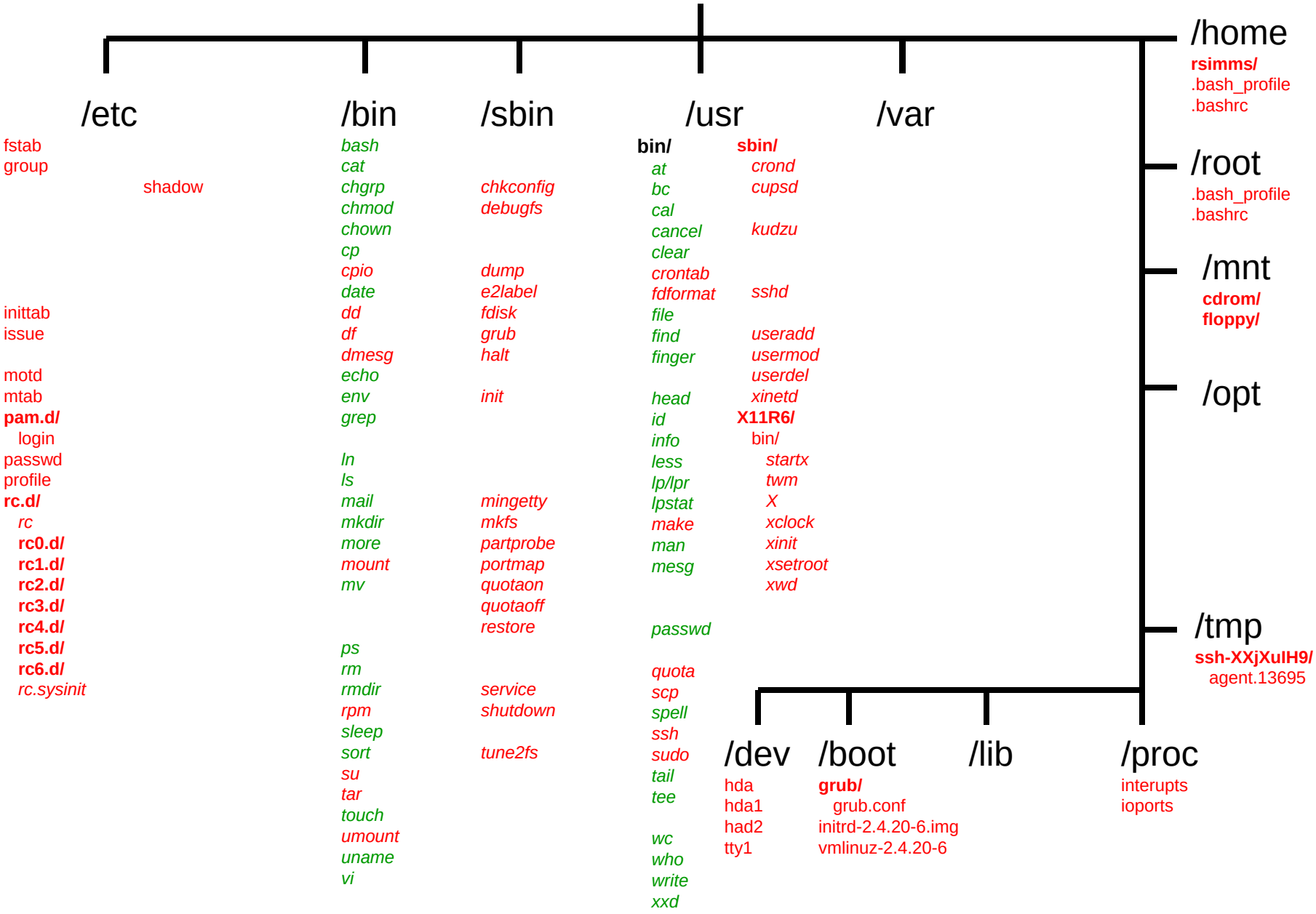


Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset

Example GNU/Linux Directory Structure

/

CIS 191 files, directories, commands

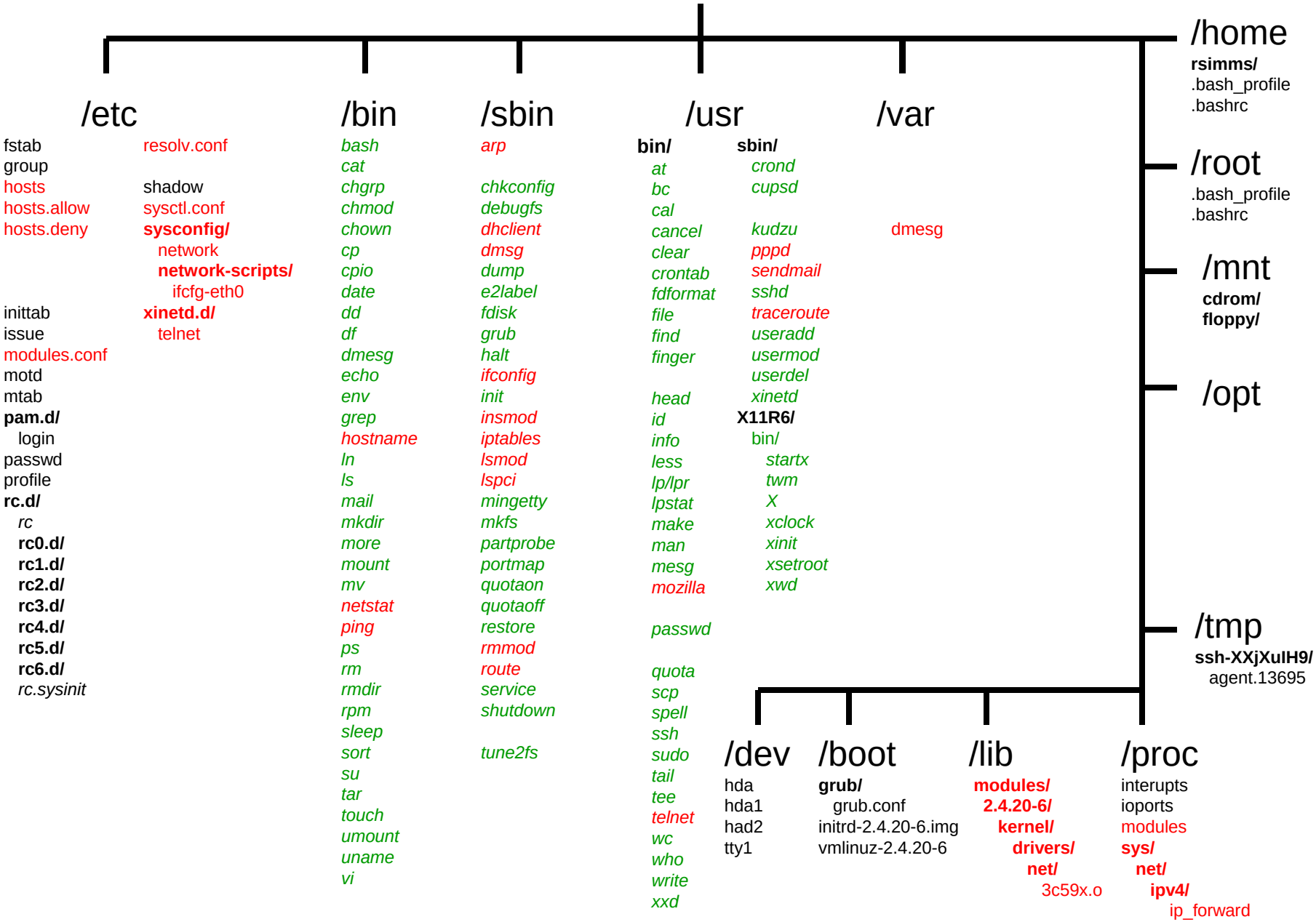


Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset

Example GNU/Linux Directory Structure

/

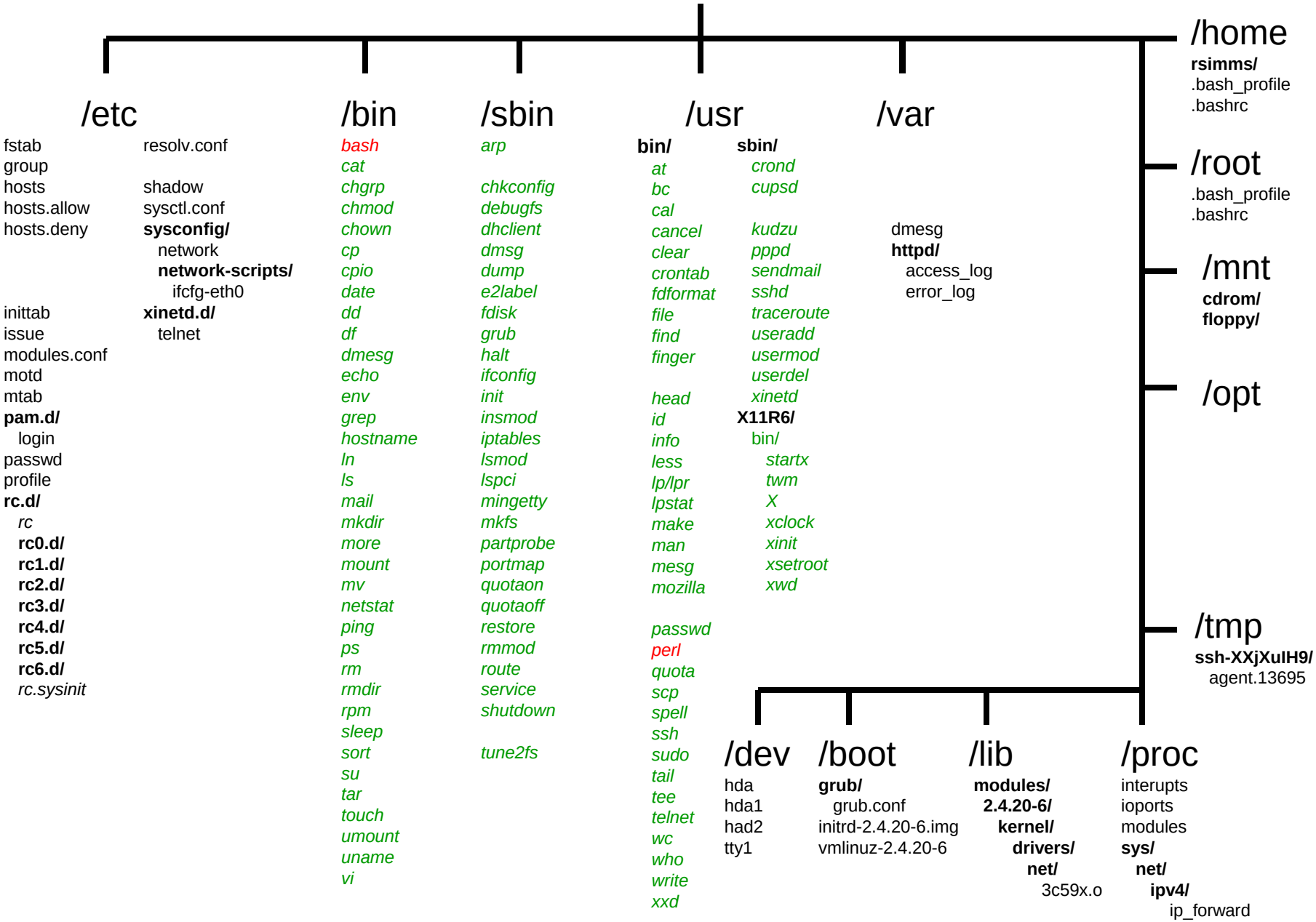
CIS 192 files, directories, commands



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

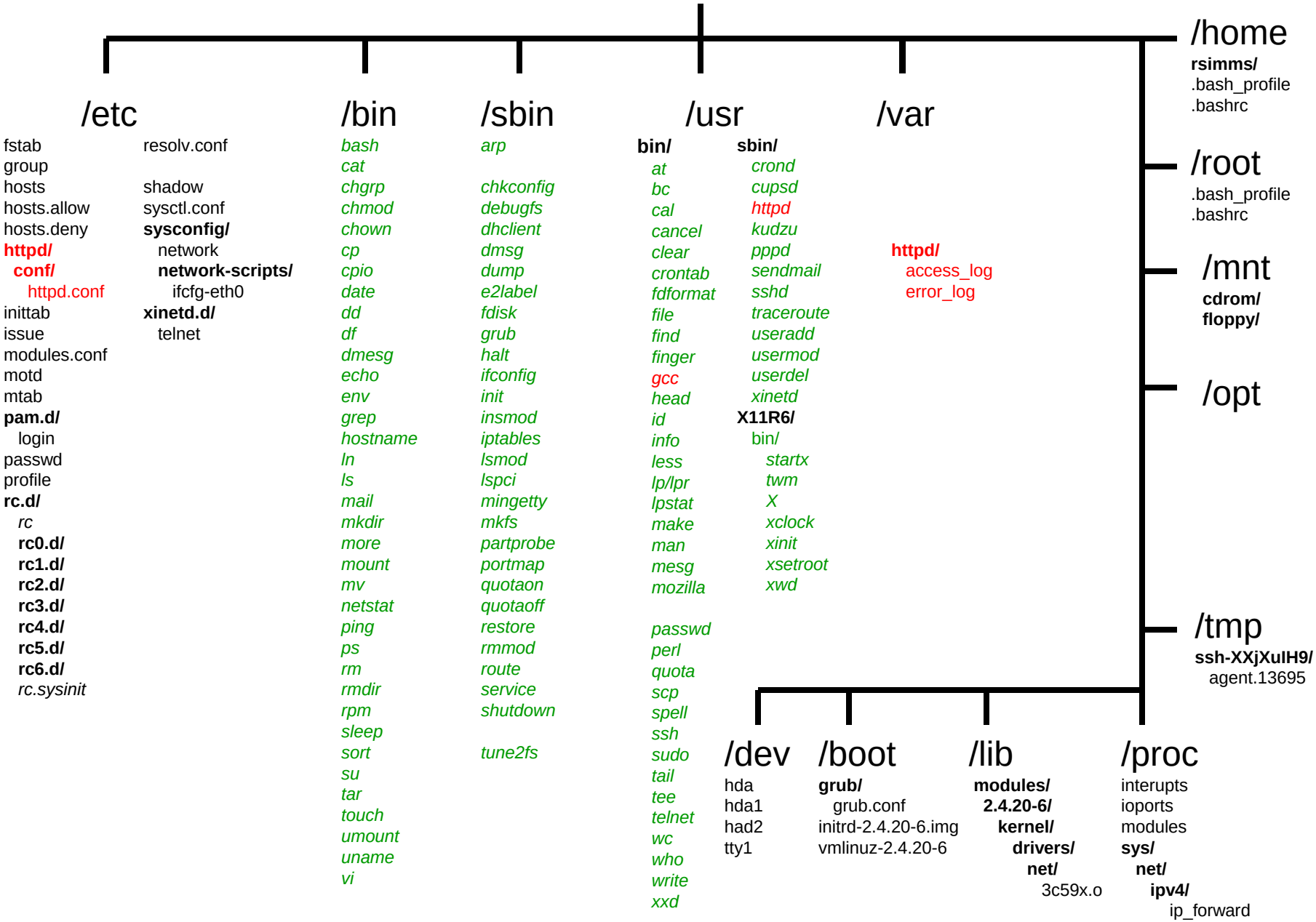


Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset shell keywords = if, then, else, case, for, while

Example GNU/Linux Directory Structure

/

CIS 164 files, directories, commands

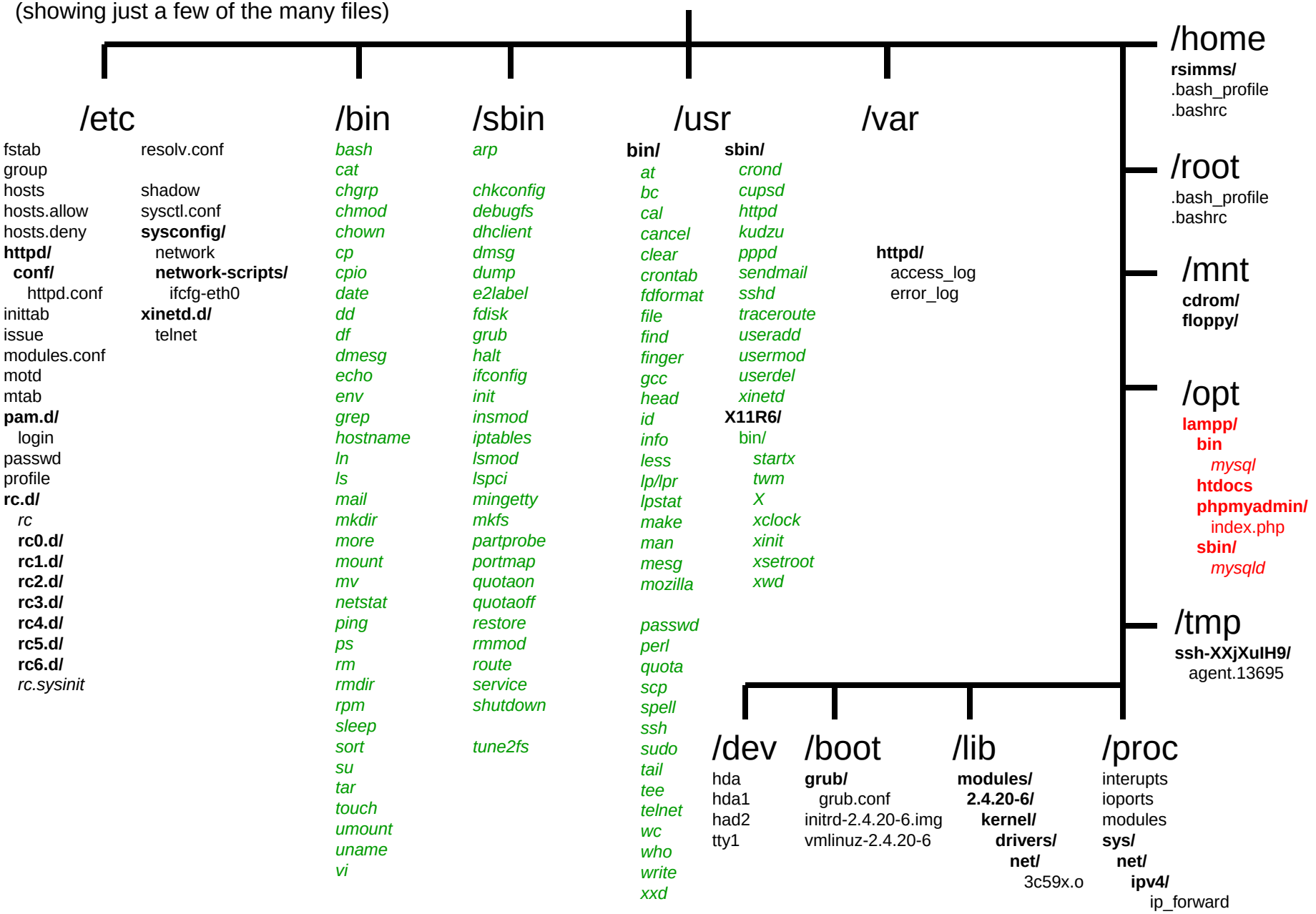


Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset shell keywords = if, then, else, case, for, while

Example GNU/Linux Directory Structure

(showing just a few of the many files)

CIS 165PH files, directories, commands

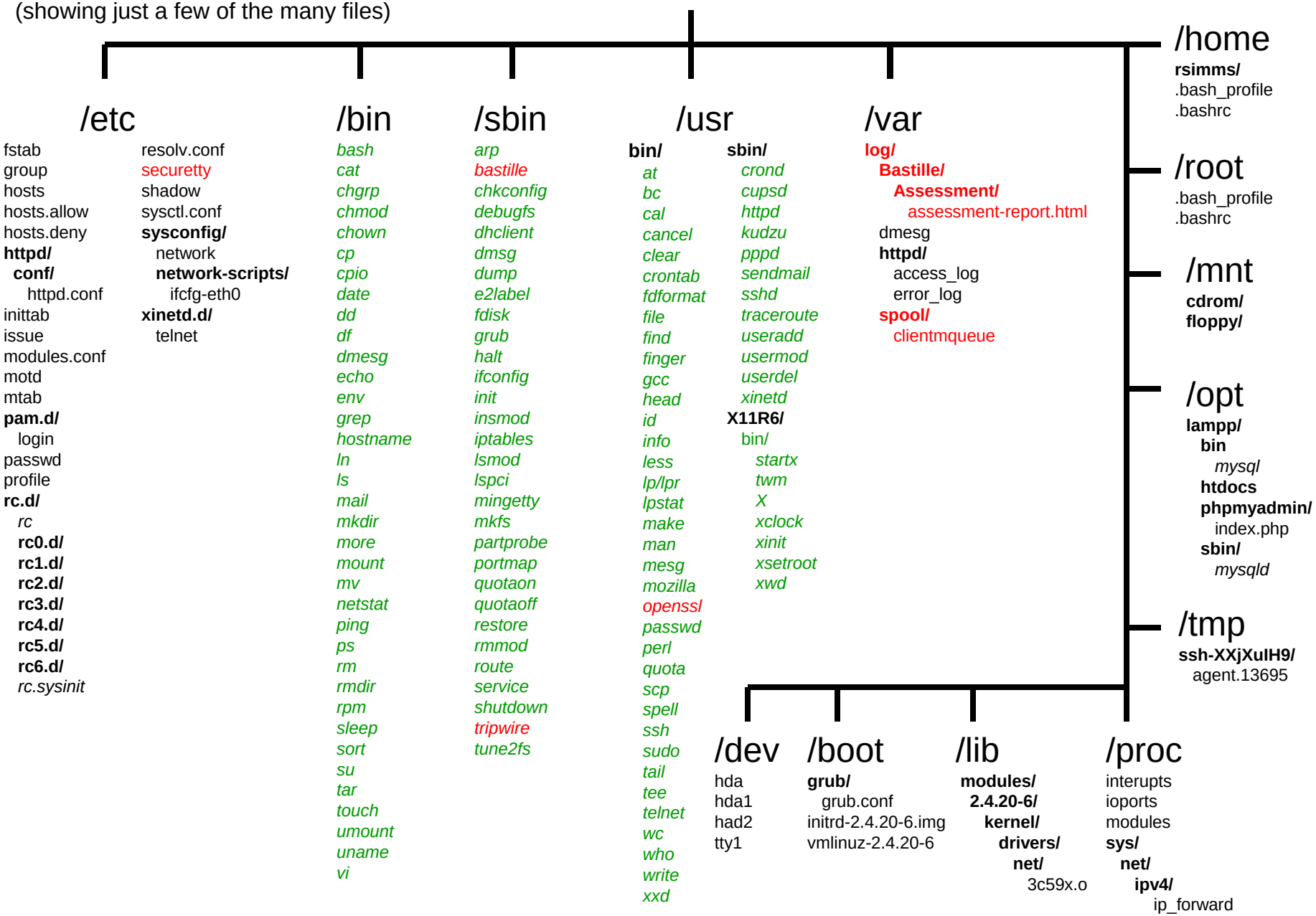


Note: shell builtins = cd, echo, exit, export, history, jobs, kill, pwd, set, type, umask, unset shell keywords = if, then, else, case, for, while

Example GNU/Linux Directory Structure

(showing just a few of the many files)

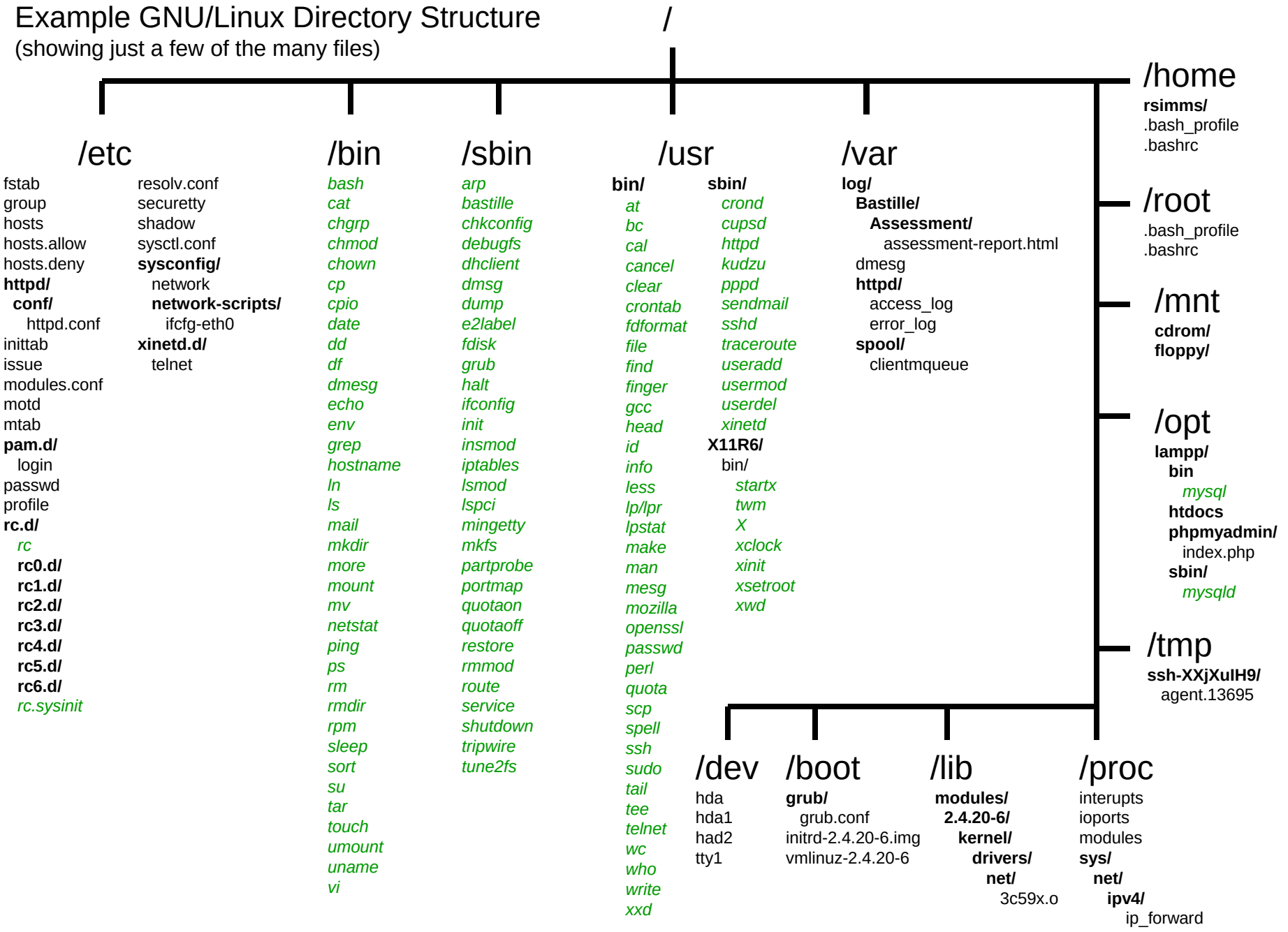
/ CIS 193 files, directories, commands



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

Example GNU/Linux Directory Structure

(showing just a few of the many files)



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

Navigating the file system

cd command

change directory

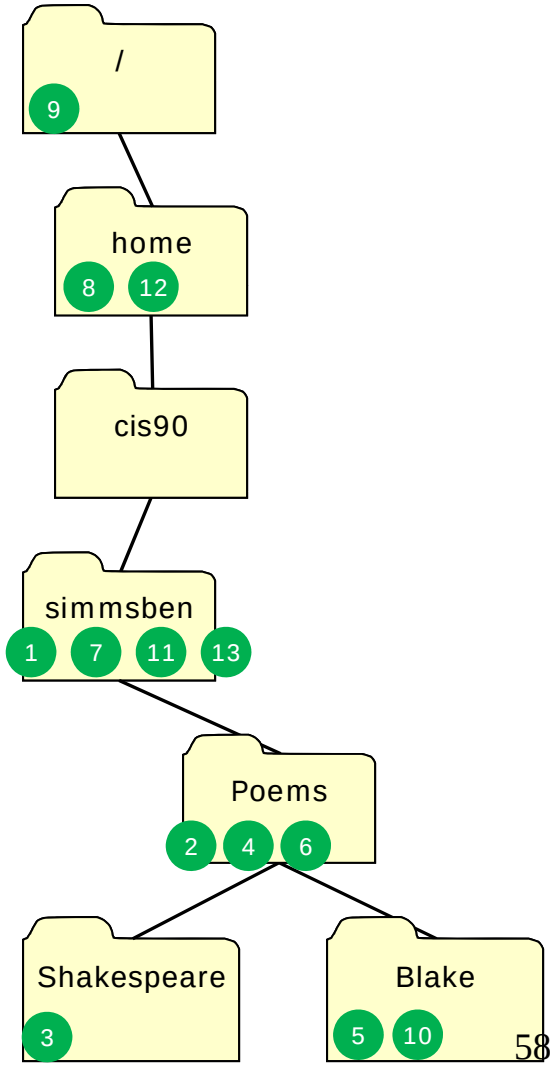
- Syntax: **cd [*directory*]** *takes a relative or absolute path*
- Changes the current working directory to the directory specified.
- Users start out in their home directory (specified in the /etc/passwd file)
- Users can always return to their home directory by typing just **cd** (with no directory specified)
- The *directory* can be:
  An absolute pathname, e.g. **cd /home/cis90/duke/Poems/ant**
  A relative pathname, e.g. **cd /Poems**
  A **..** for the parent of the current working directory, e.g. **cd ..**
  A **.** for the current directory (less used)

  If no directory is specified, you will change to your home directory
- **cd** is a Bash builtin command (part of the shell itself)
  /home/cis90/simmsben $ **type cd**
  cd is a shell builtin

cd command

change directory

```
/home/cis90/simmsben $ echo $HOME
/home/cis90/simmsben
/home/cis90/simmsben $ echo $PS1
$PWD $
1  /home/cis90/simmsben $ cd Poems/
2  /home/cis90/simmsben/Poems $ cd Shakespeare/
3  /home/cis90/simmsben/Poems/Shakespeare $ cd ..
4  /home/cis90/simmsben/Poems $ cd Blake/
5  /home/cis90/simmsben/Poems/Blake $ cd ..
6  /home/cis90/simmsben/Poems $ cd ..
7  /home/cis90/simmsben $ cd /home
8  /home $ cd ..
9  / $ cd /home/cis90/simmsben/Poems/Blake/
10 /home/cis90/simmsben/Poems/Blake $ cd
11 /home/cis90/simmsben $ cd ../../
12 /home $ cd
13 /home/cis90/simmsben $
```



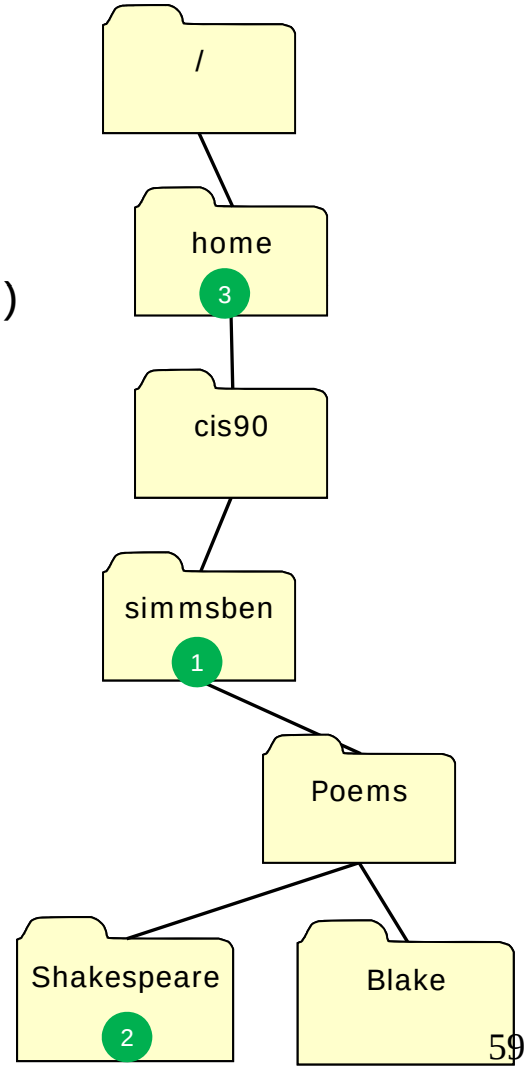
pwd command

print working directory

- Syntax: **pwd**
- Prints the current working directory.
- pwd is a BASH builtin command (part of the shell itself)

```
/home/cis90/simmsben $ type pwd
pwd is a shell builtin
```

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



Class Exercise

- Go to your home directory using just **cd** by itself
- Navigate down through the Poems directory and into the Shakespeare directory.
  - Use **ls** to list the contents of each directory after you cd into it.
  - Use **pwd** in each directory to see where you are.
- Navigate up using **cd ..** to the top of the UNIX file tree.
  - Use **ls** to list the contents of each directory after you cd into it.
  - Use **pwd** in each directory to see where you are.
- Return to your home directory using just **cd**

ls command

lists files

- Syntax: **ls [-a –i –d –l –F S -R] [*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* can be:
  An absolute pathname, e.g. **cd /home/cis90/duke/Poems/**
  A relative pathname, e.g. **cd /Poems**
  If no directory is specified, the current working directory is used.
  Can also be a filename e.g. **ls –l /etc/passwd** to show permissions
  More than one directory can be specified

- Use **man ls** to see more information.

ls command

lists files

FYI ...

- **ls** is in /bin and has been aliased to use color on tty's (not pipes)

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

- Note: the `–color=tty` added by the alias is what enables the color classifications

We will learn about aliases later in the course

ls command

lists files

```
/home/cis90/simmsben $ cd
```

cd *with no arguments take you to your home directory*

```
/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
```

ls *with no arguments shows all the files and directories in your current working directory*

black *indicates regular files,*
blue *indicates directories and*
green *indicates executables (programs or scripts)*

ls command

lists files

```
/home/cis90/simmsben $ cd
```

cd *with no arguments take you to your home directory*

```
/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*
```

ls *with –F option shows file types using symbols*

no symbol *indicates regular files,*
/ *indicates directories and*
******* *indicates executables (programs or scripts)*

ls command

lists files

```
/home/cis90/simmsben $ cd
```

cd with no arguments take 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 $
```

. is "this directory" or "here",
.. is the parent directory

ls command

lists files

```
/home/cis90/simmsben $ cd
```

cd *with no arguments take you to your home directory*

```
/home/cis90/simmsben $ ls -i
105056 bigfile  102566 Lab2.1         102608 proposal1   102613 text.err
102542 bin      102576 letter         102609 proposal2   102614 text.fxd
102551 empty    102577 Miscellaneous  102610 proposal3   102615 timecal
102552 Hidden   102582 mission        102611 small_town  102616 what_am_i
102555 Lab2.0   102584 Poems          102612 spellk
```

ls *with –i option shows inode numbers*

Class Exercise

- Go to your home directory using just **cd**
- Use the ls command with different option combinations
  - **ls**
  - **ls –a**
  - **ls –i**
  - **ls –F**
  - **ls -aiF**

Class Exercise

- Type: **ls -3**
  Which program generated the error message?

- Type: **lx**
  Which program generated the error message?

Metacharacters

* (filename expansion character)

Matches:

- all non-hidden files in the current directory when used alone
- zero or more characters when used as **prefix**, infix or postfix

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

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

Metacharacters

* (filename expansion character)

Matches:

- all non-hidden files in the current directory when used alone
- zero or more characters when used as prefix, infix or **postfix**

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

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

Metacharacters

* (filename expansion character)

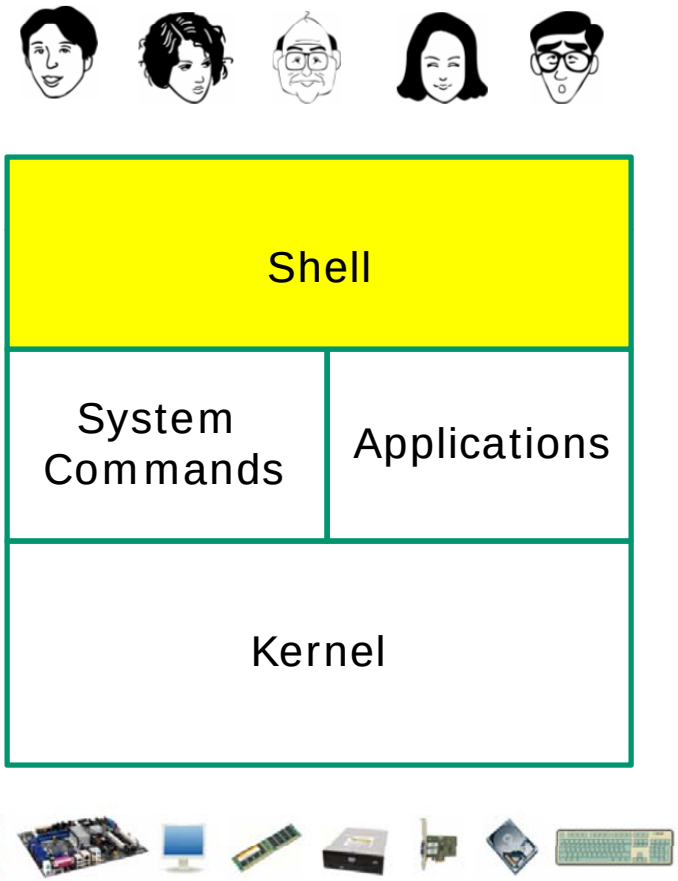
Matches:

- all non-hidden files in the current directory when used alone
- zero or more characters when used as prefix, **infix** or postfix

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

am matches all file names containing "am"*

Life of the Shell



1) Prompt
2) Parse
3) Search
4) Execute
5) Nap
6) Repeat

Metacharacters are processed during the Parse step

ls command

Using files and directories as arguments

```
/home/cis90/simmsben $ 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/simmsben $ ls bigfile
bigfile
```

Specify a file, just the file name is listed

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

Specify a directory, then the contents of the directory are listed

ls command

specifying multiple directories

Three directories are specified

```
/home/cis90/simmsben $ ls Poems/ bin/ Miscellaneous/
bin/:
app  banner  enlightenment  hi  I  treed  tryme  zoom

Miscellaneous/:
better_town  file.dos  fruit  manpage  mystery  salad

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

*Note the mystery file is **light blue** ... it is a symbolic link to another file*

ls command

specifying multiple files and directories

```
/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

/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        READNAME     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
```

Regular files listed first

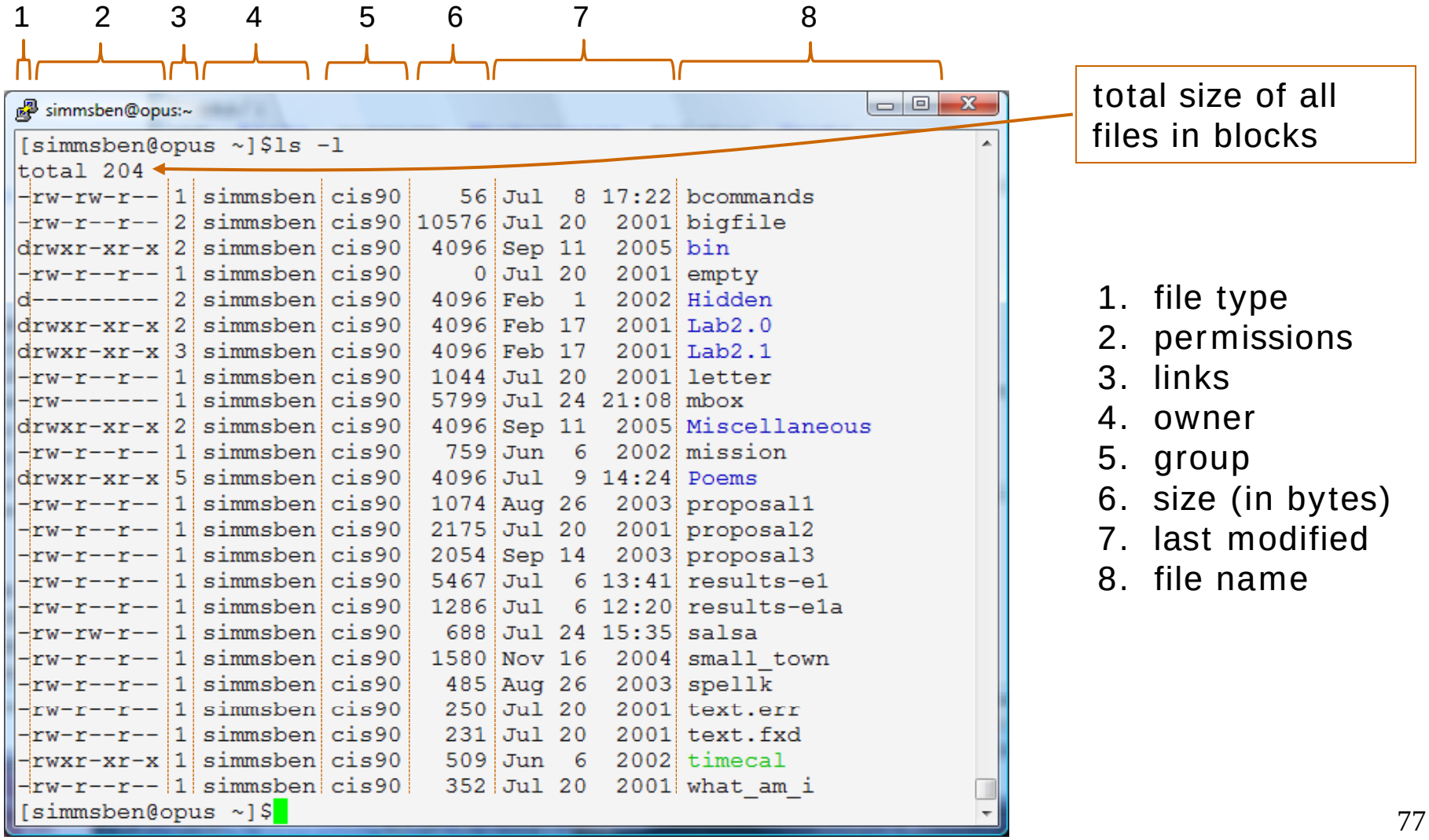
*Followed by each directory **expanded** to show that directory's files*

Class Exercise

- Go to your home directory using just **cd**
- Use the **ls** command with different arguments
  - **ls**
  - **ls p***
  - **ls mission**
  - **ls poems**
  - **ls Poems**
  - **ls Poems/Yeats**
  - **ls P*/Sh*/son***
  - **ls mission Lab2.0**
  - **ls *n**
  - **ls *.*** *(this is NOT like the DOS command!)*

ls command

long listing (-l)



1. file type
2. permissions
3. links
4. owner
5. group
6. size (in bytes)
7. last modified
8. file name

ls command

long listing (-l), sorted by size (-S)

simmsben@opus:~

```
[simmsben@opus ~]$ls -lS
total 204
-rw-r--r-- 2 simmsben cis90 10576 Jul 20  2001 bigfile
-rw------- 1 simmsben cis90  5799 Jul 24 21:08 mbox
-rw-r--r-- 1 simmsben cis90  5467 Jul  6 13:41 results-e1
drwxr-xr-x 2 simmsben cis90  4096 Sep 11  2005 bin
d--------- 2 simmsben cis90  4096 Feb  1  2002 Hidden
drwxr-xr-x 2 simmsben cis90  4096 Feb 17  2001 Lab2.0
drwxr-xr-x 3 simmsben cis90  4096 Feb 17  2001 Lab2.1
drwxr-xr-x 2 simmsben cis90  4096 Sep 11  2005 Miscellaneous
drwxr-xr-x 5 simmsben cis90  4096 Jul  9 14:24 Poems
-rw-r--r-- 1 simmsben cis90  2175 Jul 20  2001 proposal2
-rw-r--r-- 1 simmsben cis90  2054 Sep 14  2003 proposal3
-rw-r--r-- 1 simmsben cis90  1580 Nov 16  2004 small_town
-rw-r--r-- 1 simmsben cis90  1286 Jul  6 12:20 results-e1a
-rw-r--r-- 1 simmsben cis90  1074 Aug 26  2003 proposal1
-rw-r--r-- 1 simmsben cis90  1044 Jul 20  2001 letter
-rw-r--r-- 1 simmsben cis90   759 Jun  6  2002 mission
-rw-rw-r-- 1 simmsben cis90   688 Jul 24 15:35 salsa
-rwxr-xr-x 1 simmsben cis90   509 Jun  6  2002 timecal
-rw-r--r-- 1 simmsben cis90   485 Aug 26  2003 spellk
-rw-r--r-- 1 simmsben cis90   352 Jul 20  2001 what_am_i
-rw-r--r-- 1 simmsben cis90   250 Jul 20  2001 text.err
-rw-r--r-- 1 simmsben cis90   231 Jul 20  2001 text.fxd
-rw-rw-r-- 1 simmsben cis90    56 Jul  8 17:22 bcommands
-rw-r--r-- 1 simmsben cis90     0 Jul 20  2001 empty
[simmsben@opus ~]$
```

ls command

just the directory itself (-d)

```
/home/cis90/simmsben $ ls -l bin
total 68
-rwxr-xr-x 1 simmsben cis90  220 Apr 22  2004 app
-rwxr-xr-x 1 simmsben cis90 6160 Aug 28  2003 banner
-rwxr-xr-x 1 simmsben cis90 3388 Sep 11  2005 enlightenment
-rwxr-xr-x 1 simmsben cis90  107 Jul 20  2001 hi
-rwxr-x--x 1 simmsben cis90  375 Oct 20  2003 I
-rwxr-xr-x 1 simmsben cis90  190 Jul 20  2001 treed
-rwxr-xr-x 1 simmsben cis90  174 Mar  4  2004 tryme
-rwxr-xr-x 1 simmsben cis90   74 Jul 20  2001 zoom


/home/cis90/simmsben $ ls -ld bin
drwxr-xr-x 2 simmsben cis90 4096 Sep 11  2005 bin
/home/cis90/simmsben $
```

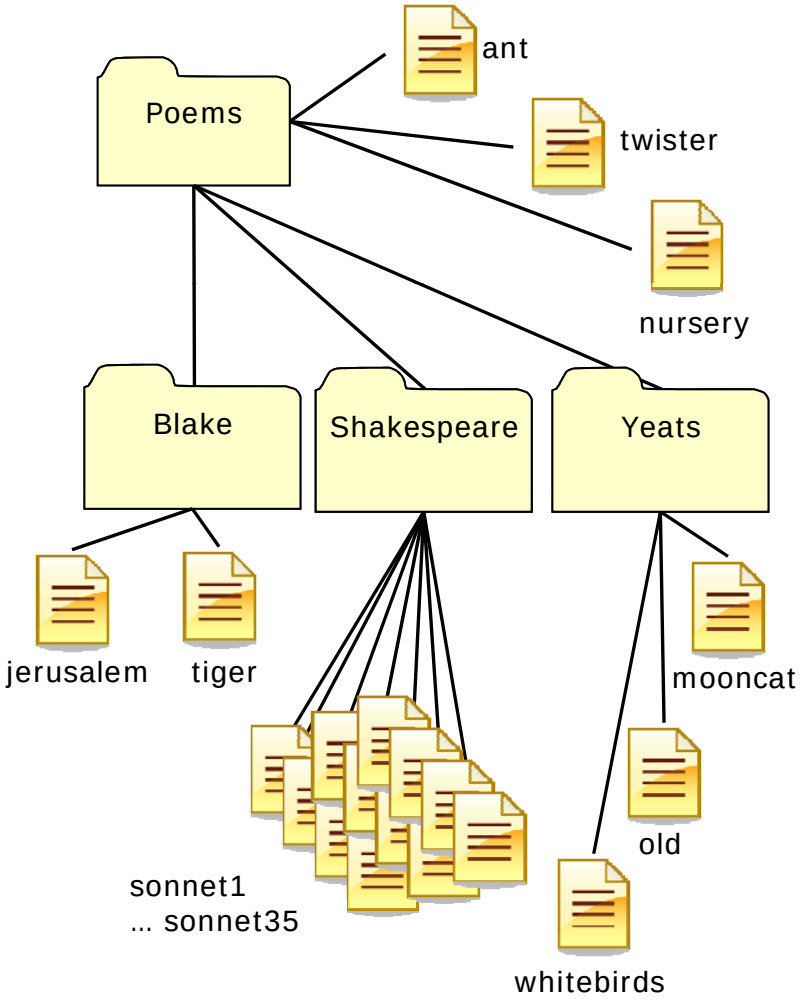
*Use the **d** option to get long listing information about the directory itself. Without the **d** the directory contents are listed instead.*

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]$
```

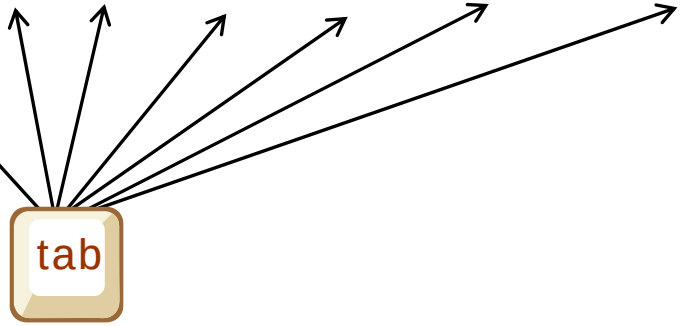


bash shell tip

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/simmsben/Poems/Shakespeare/
```

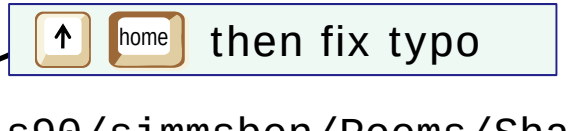


bash shell tip

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



Class Exercise

- Do a long listing of your home directory with **ls -l**
- Do long listing the files in the Poems directory with **ls –l Poems/**
- Do a long listing on the Poems directory itself with **ls –ld Poems/**
- Do long listing of all the Shakespeare sonnets using an absolute pathname (hint: use tab completion)

  ls -l /home/cis90/simmsben/Poems/Shakespeare/

  Use your own username

- Do long listing of all the Shakespeare sonnets using a relative pathname (hint: use tab completion)

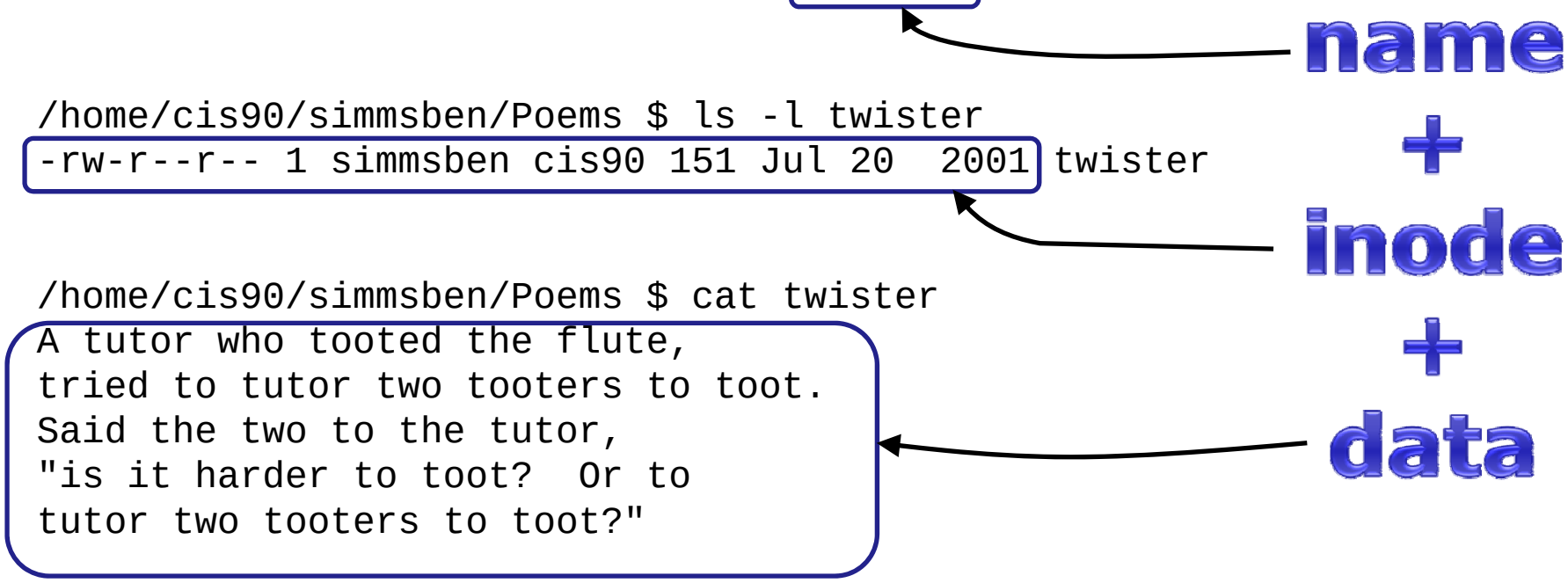
  ls –l Poems/Shakespeare/

- Do long listing of your home directory including inodes with **ls -li**

File Types

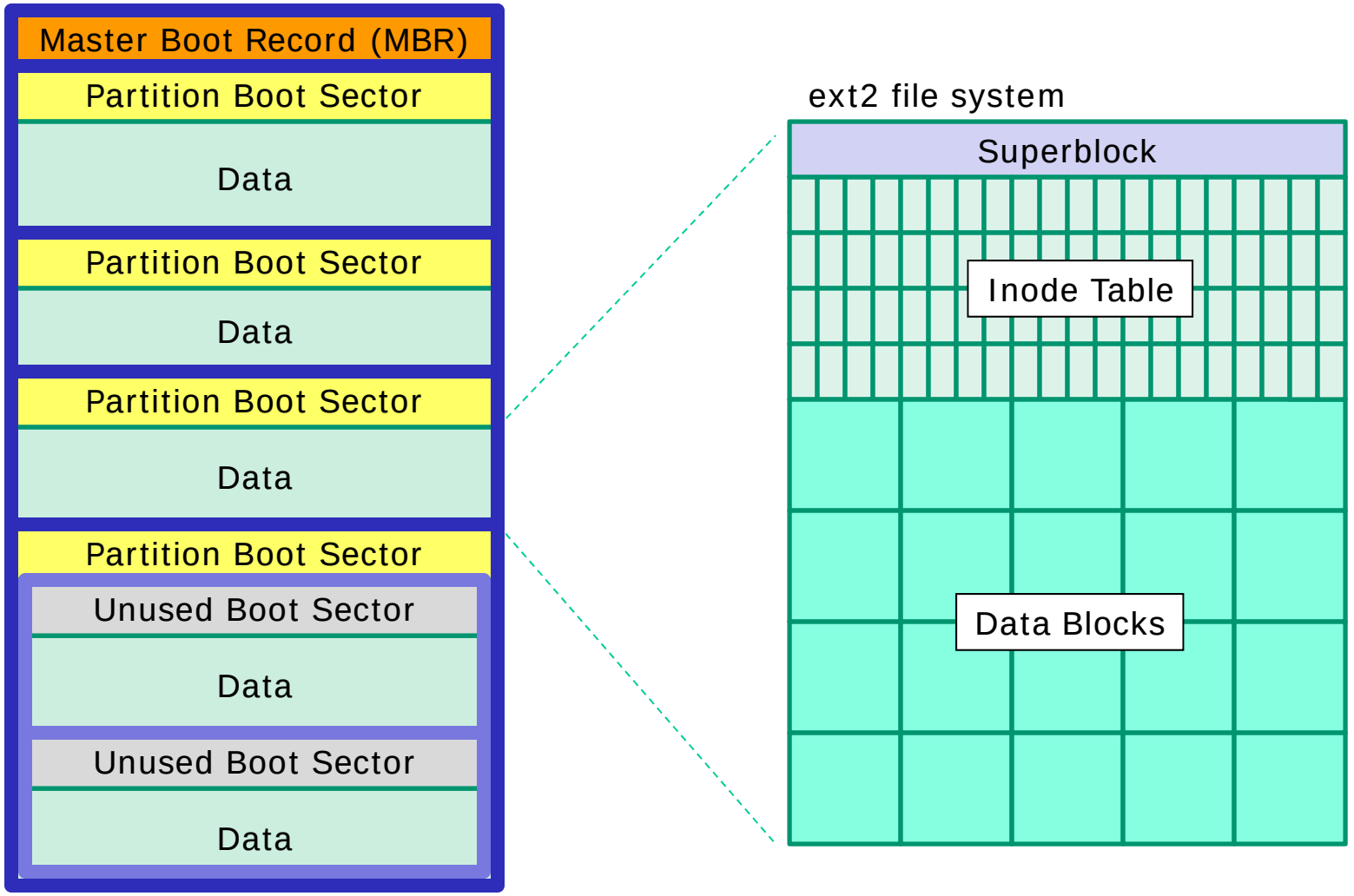
UNIX Files
The three elements of a file

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

/home/cis90/simmsben/Poems $ ls -l twister
-rw-r--r-- 1 simmsben cis90 151 Jul 20  2001 twister

/home/cis90/simmsben/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?"
```

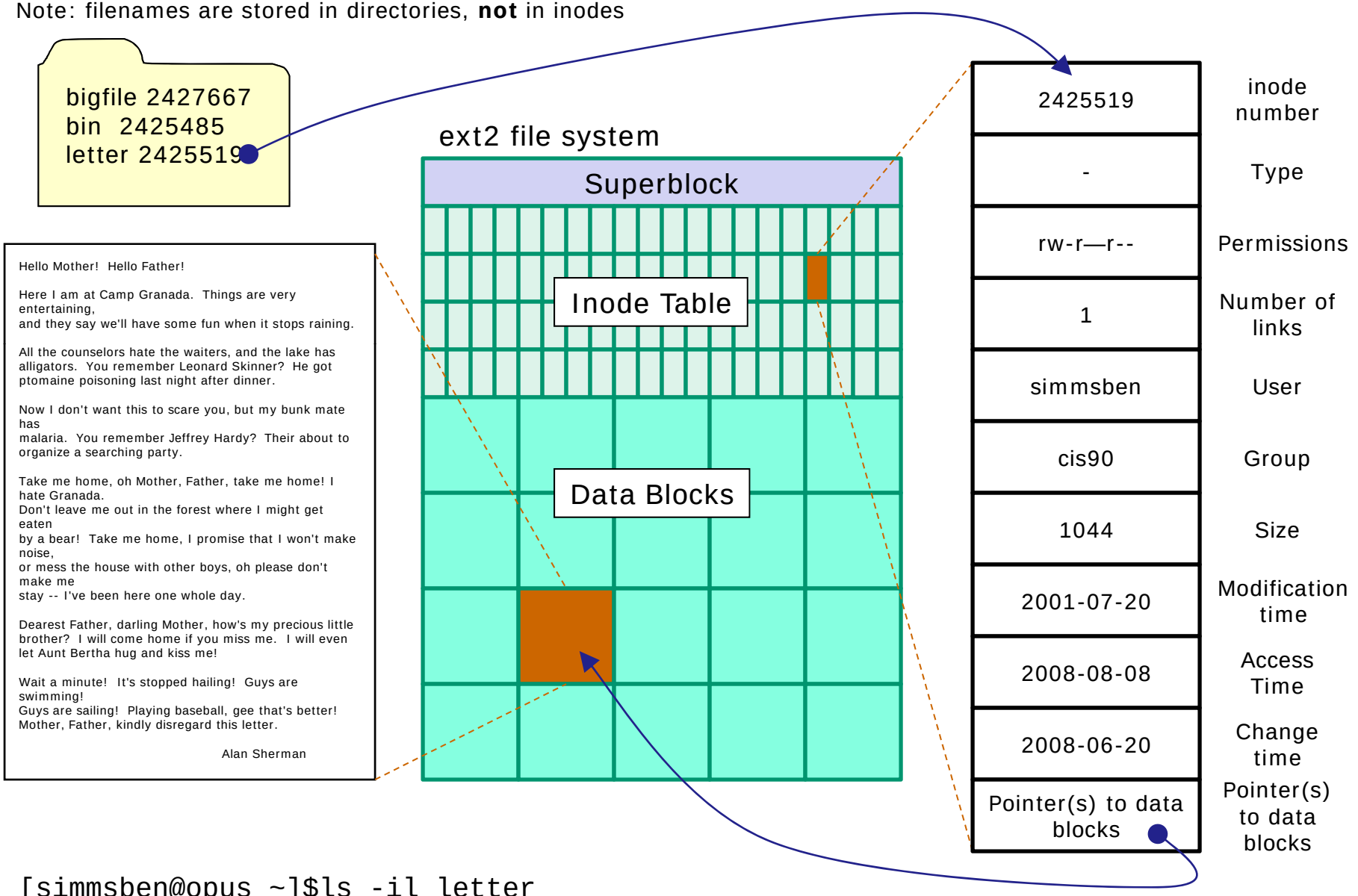


File Systems

Linux



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



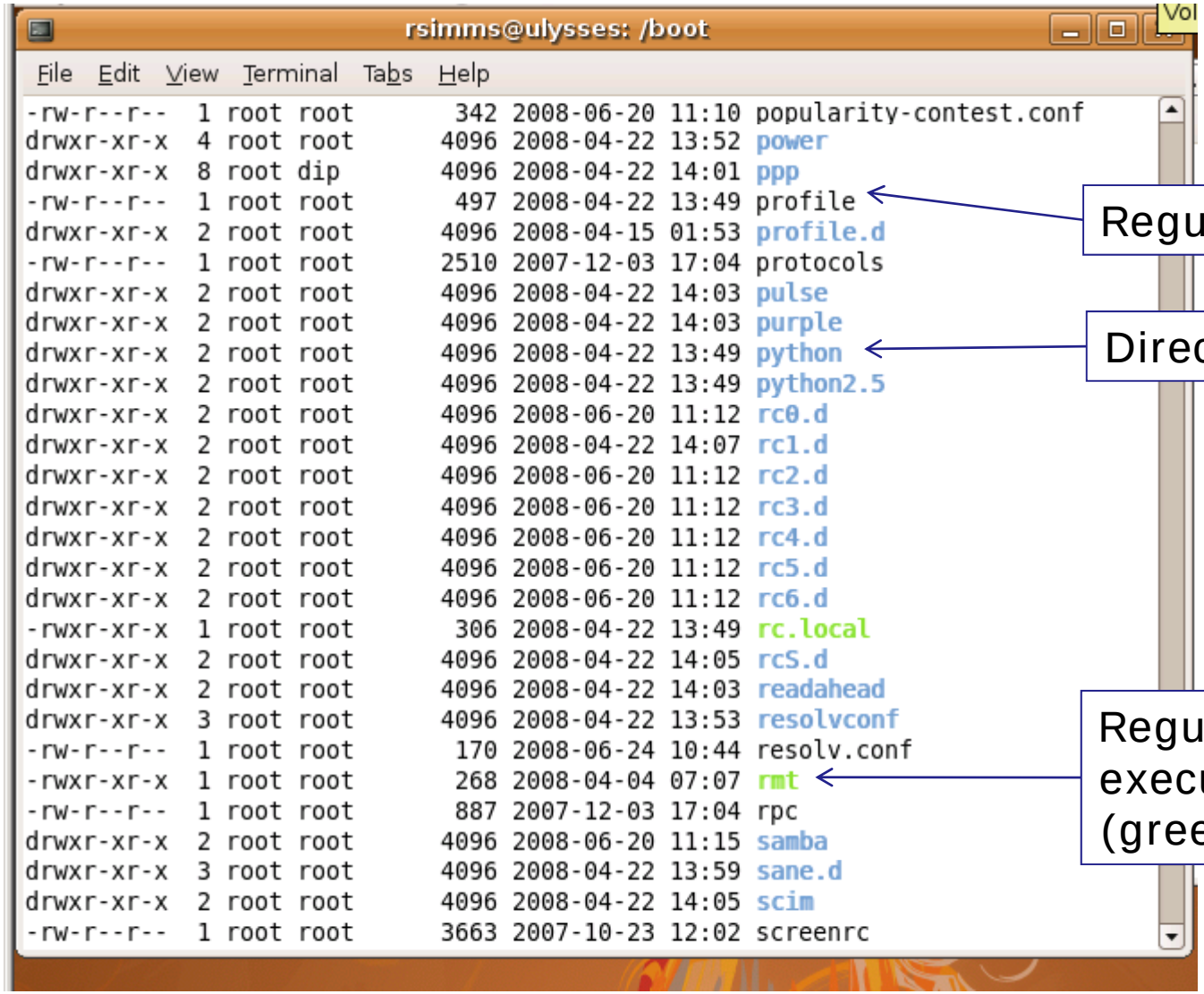

```
[simmsben@opus ~]$ls -il letter
2425519 -rw-r--r-- 1 simmsben cis90 1044 Jul 20  2001 letter
```

File Types and Commands

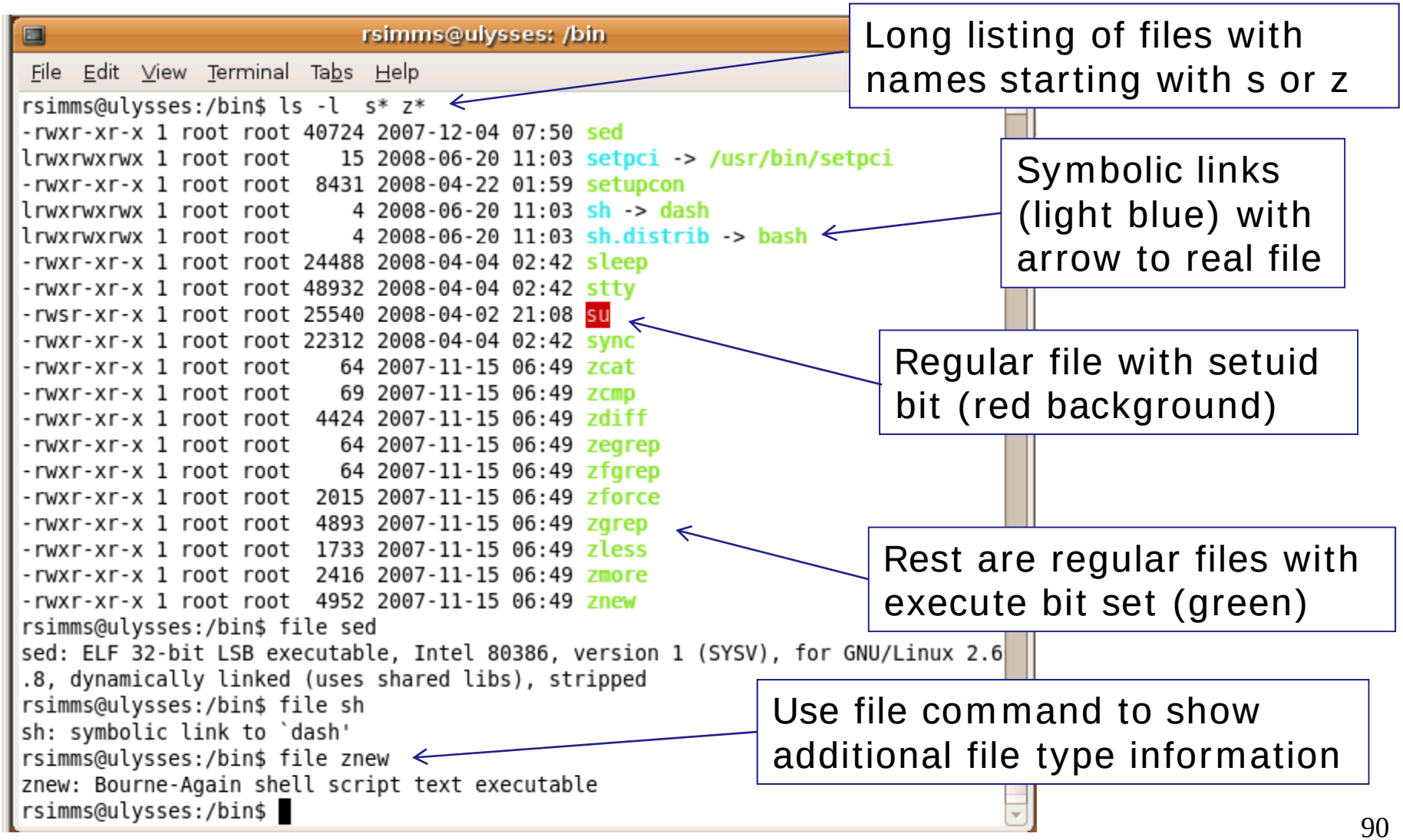
Long listing code (ls –l)	Type	How to make one
d	directory	mkdir
-	regular • Programs • Text • Data (binary)	touch
l	symbolic link	ln -s
c	character special device files	mknod
b	block special device files	mknod

Note: Other files types includes sockets (s) and named pipes (p)

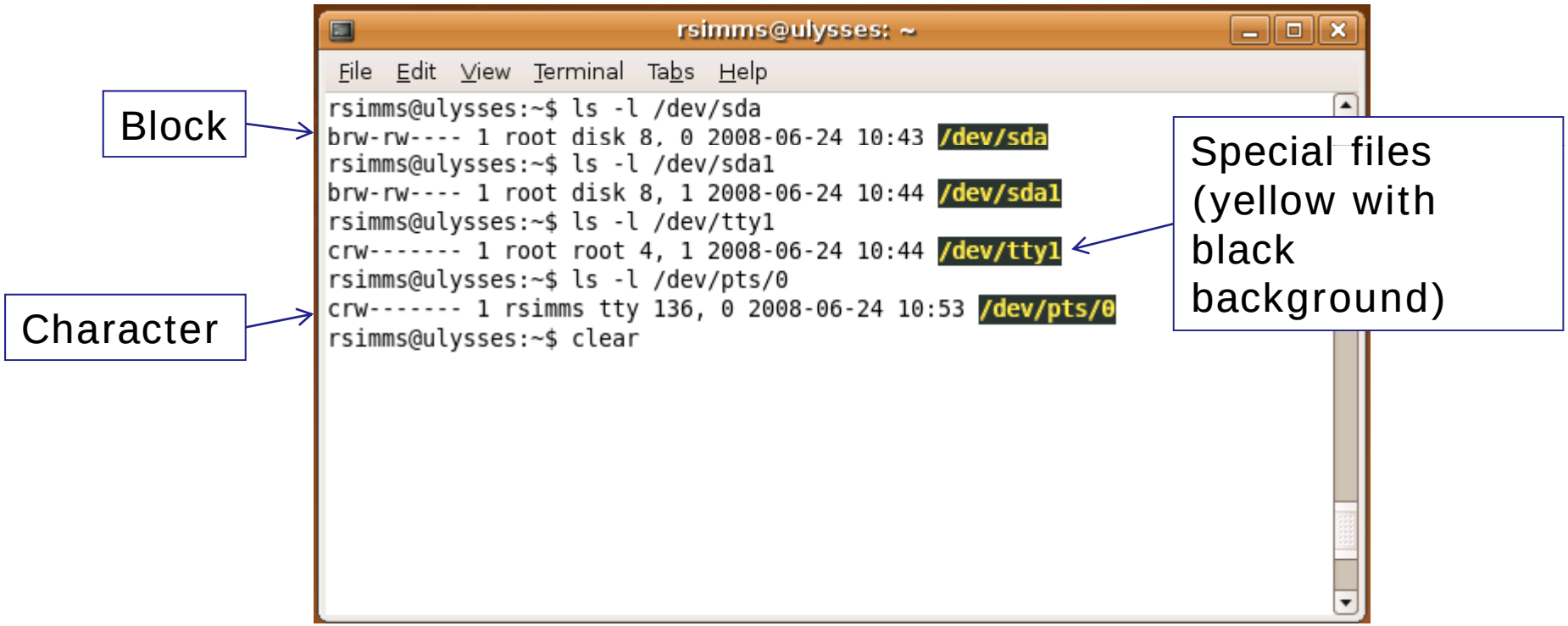
Various Types of Files (found in /etc)



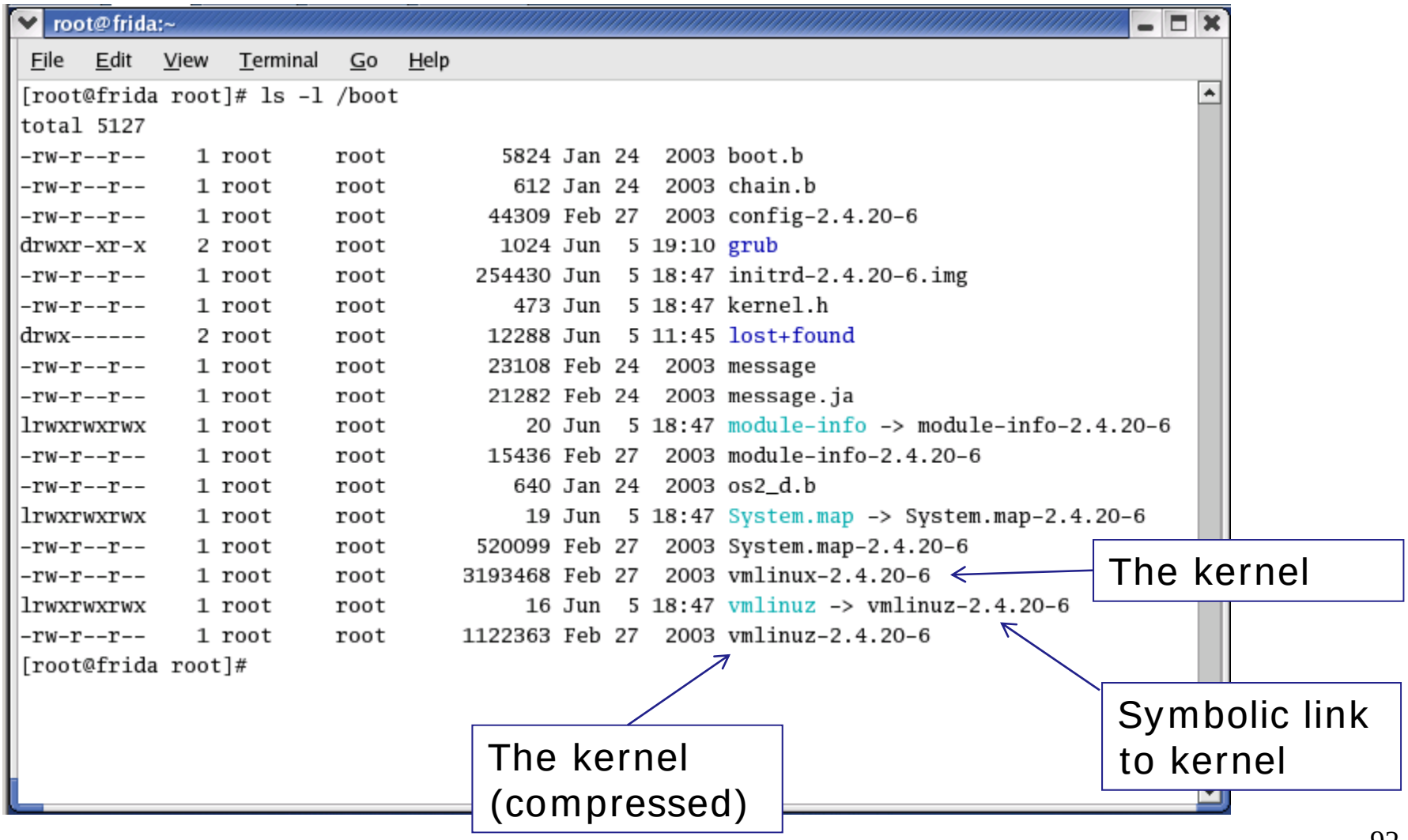
Various Types of Files (found in /bin)



Various Types of Files (found in /dev)



/boot (Red Hat 9)



Class Exercise

- Do a long listing of the /bin directory
  - *Who owns the vi command?*
  - *What size is the sleep command?*
  - *What file does the symbolic link tcptraceroute point to?*
  - *When was the file touch last modified?*
- Do a long listing of the /etc directory
  - *Is yum a directory or a file?*
  - *What are two ways you can tell if yum is a directory or a file?*

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.

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

file command

determines the type of file

```
rsimms@opus:~/work/examples/filetypes
[rsimms@opus filetypes]$ ls -l
total 156
-rw-r--r-- 1 rsimms cis191  8983 Aug  5 07:57 Adjective.frm
-rw-rw-rw- 1 rsimms cis191  5976 Aug  5 07:57 Adjective.MYD
-rw-rw-rw- 1 rsimms cis191  2048 Aug  5 07:57 Adjective.MYI
-rw-r--r-- 1 rsimms cis191 10240 Aug  4 18:13 backup.tar
-rw------- 1 rsimms users    191 Aug  5 08:10 bash_profile
crw-r--r-- 1 rsimms cis191  5, 1 Aug  5 08:03 console
-rwx------ 1 rsimms cis191  4846 Aug  4 18:08 cprog
lrwxrwxrwx 1 rsimms users      5 Aug  5 08:07 go-cprog -> cprog
-rw-r--r-- 1 rsimms cis191   119 Aug  4 17:55 letter
-rw------- 1 rsimms users   2968 Aug  5 08:08 mbox
-rw-r--r-- 1 rsimms cis191 34611 Aug  5 07:59 rich-260x216.jpg
-rwxr-xr-x 1 rsimms cis191   445 Aug  4 17:56 runit
brw-r--r-- 1 rsimms cis191  8, 0 Aug  5 08:04 sda
drwxr-xr-x 2 rsimms cis191  4096 Aug  4 17:57 travel
[rsimms@opus filetypes]$
```

Not all regular files are text files

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

```
rsimms@opus:~/work/examples/filetypes
[rsimms@opus 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
bash_profile:     ASCII English text
console:          character special (5/1)
cprog:            ELF 32-bit LSB executable, Intel 80386, version
 1 (SYSV), for GNU/Linux 2.2.5, dynamically linked (uses shared l
ibs), for GNU/Linux 2.2.5, not stripped
go-cprog:         symbolic link to `cprog'
letter:           ASCII English text
mbox:             ASCII mail text
rich-260x216.jpg: JPEG image data, JFIF standard 1.02
runit:            Bourne shell script text executable
sda:              block special (8/0)
travel:           directory
[rsimms@opus filetypes]$
```

wc command

count words, lines, and bytes

simmsben@opus:~/Poems/Blake

```
/home/cis90/simmsben/Poems/Blake $ cat tiger
Tiger, Tiger burning bright
In the forest of the night,
What immortal hand or eye
Dare frame thy fearful symmetry?
/home/cis90/simmsben/Poems/Blake $ wc -l tiger
4 tiger
/home/cis90/simmsben/Poems/Blake $ wc -w tiger
20 tiger
/home/cis90/simmsben/Poems/Blake $ wc tiger
  4  20 115 tiger
/home/cis90/simmsben/Poems/Blake $ ls -l tiger
-rw-r--r-- 1 simmsben cis90 115 Jul 20  2001 tiger
/home/cis90/simmsben/Poems/Blake $ wc -l *
 27 jerusalem
  4 tiger
 31 total
/home/cis90/simmsben/Poems/Blake $
```

Number of lines

Number of words

Number of lines, words, bytes

tiger file has 4 lines, 20 words and 115 bytes

Class Exercise

Navigate with cd, pwd, and ls

- Navigate to your Blake directory
- Use the file command on tiger: **file tiger**
  - *Is tiger a binary or ASCII text file?*
- Using one command only, with a relative path, get expanded type information on the sonnet files in your Shakespeare directory

  file ../Shakespeare/*
- Navigate to your Shakespeare directory
- Print the sonnet3 files with **cat sonnet3**

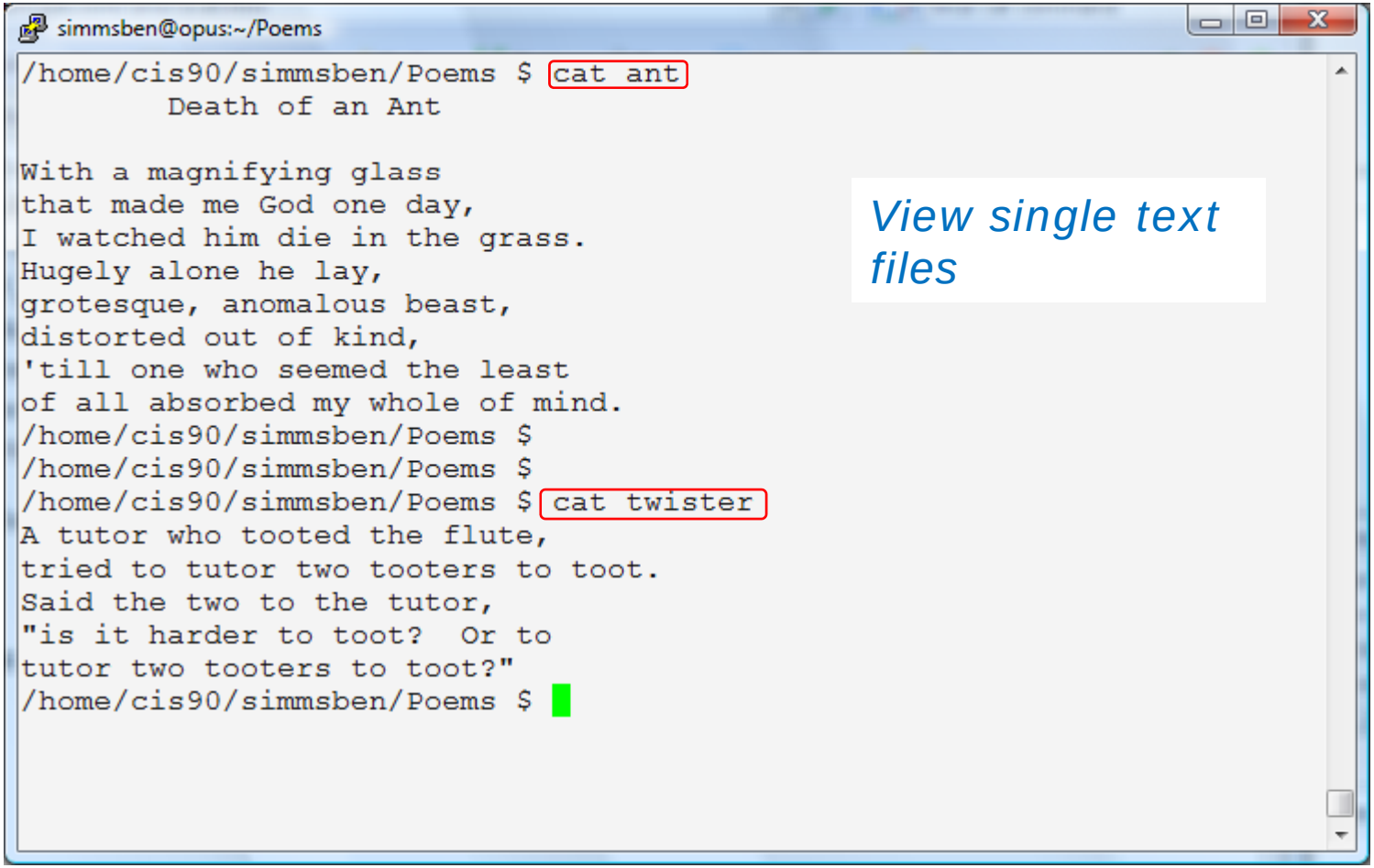
  *How many lines in the sonnet3 file? Hint: Use **wc –l sonnet3***

  How many words in the sonnet3 file?

Viewing Files

cat command

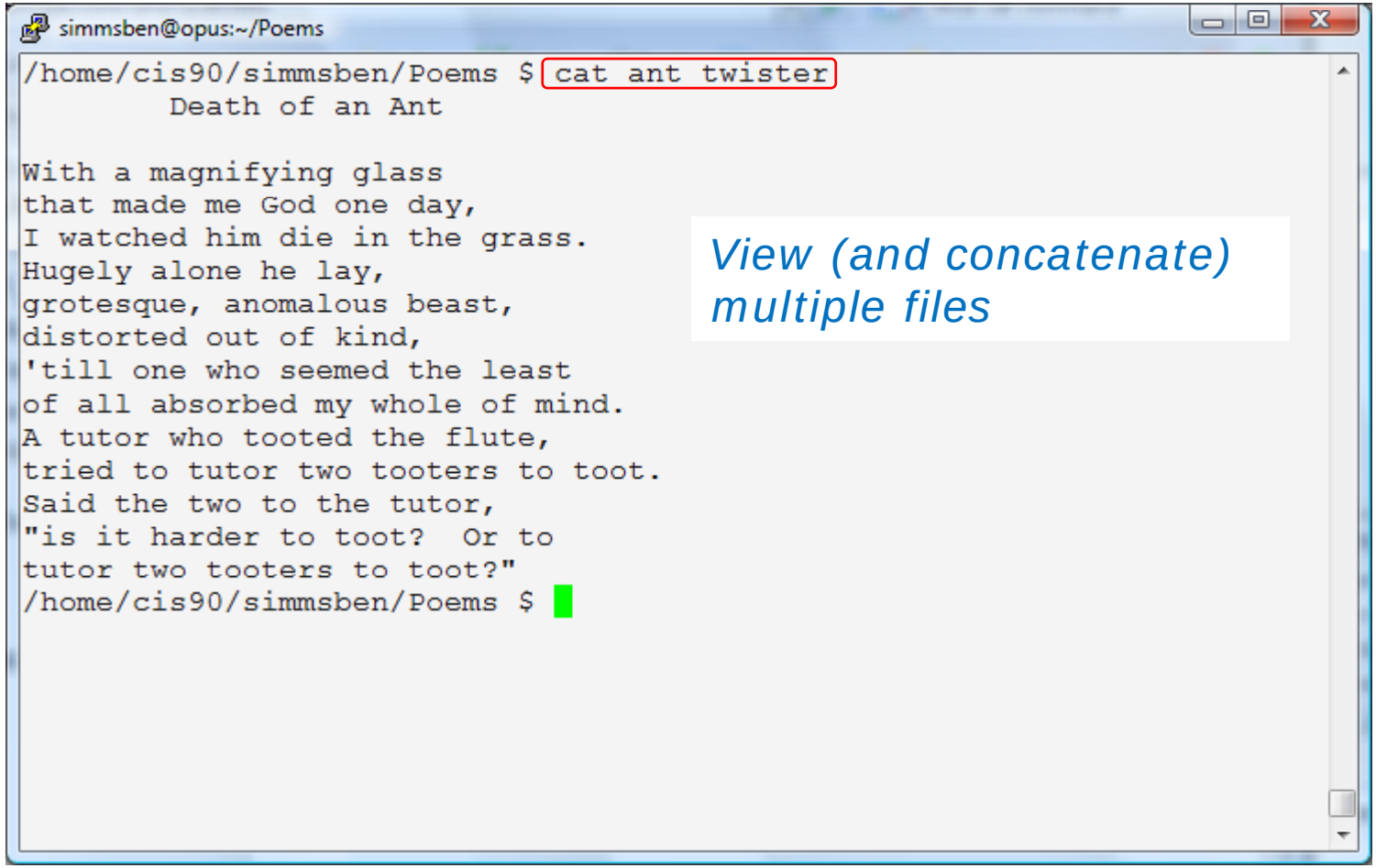
concatenate or view text files



View single text files

cat command

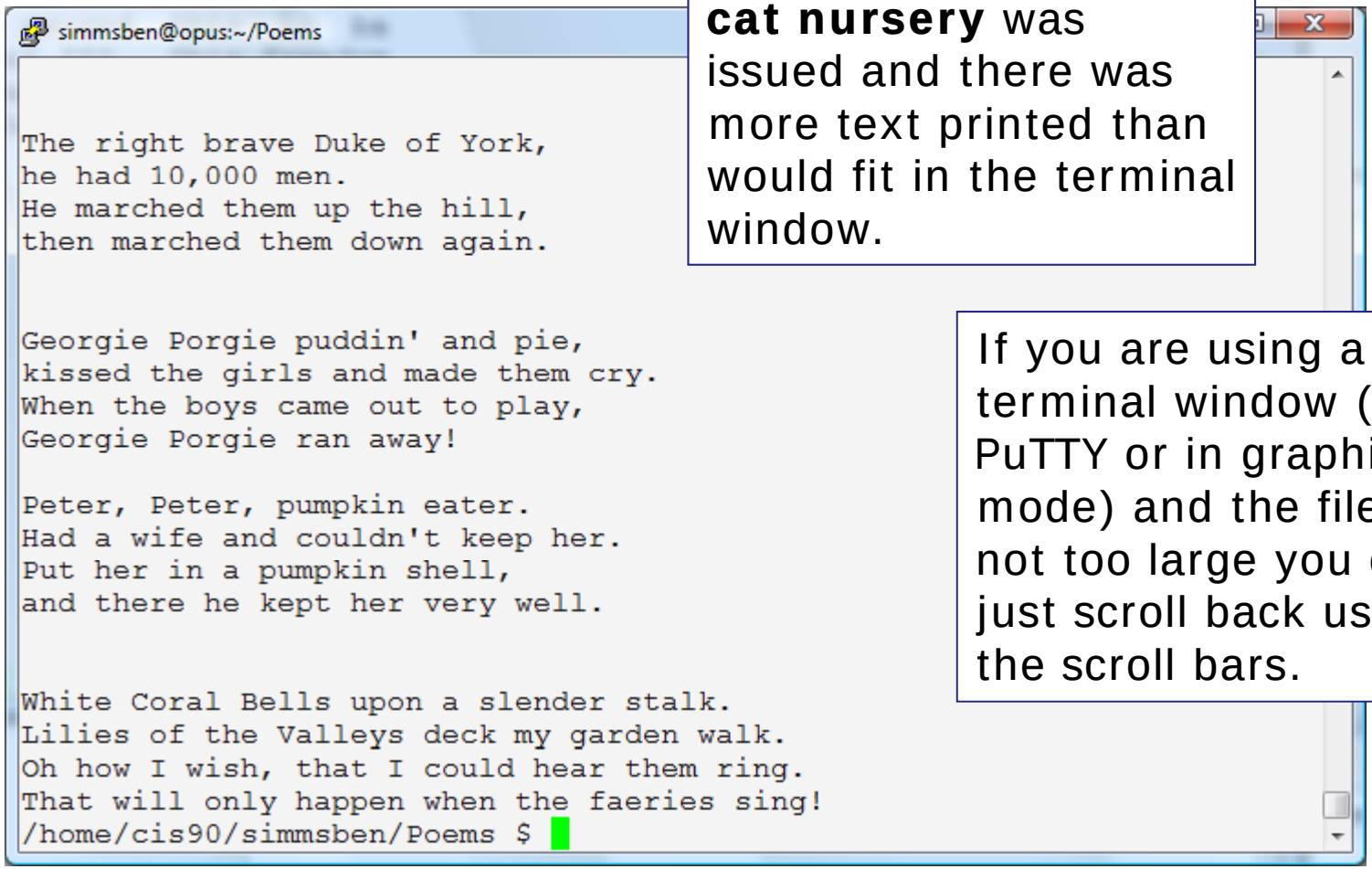
concatenate or view text files



View (and concatenate) multiple files

cat command

concatenate or view text files



cat nursery was issued and there was more text printed than would fit in the terminal window.

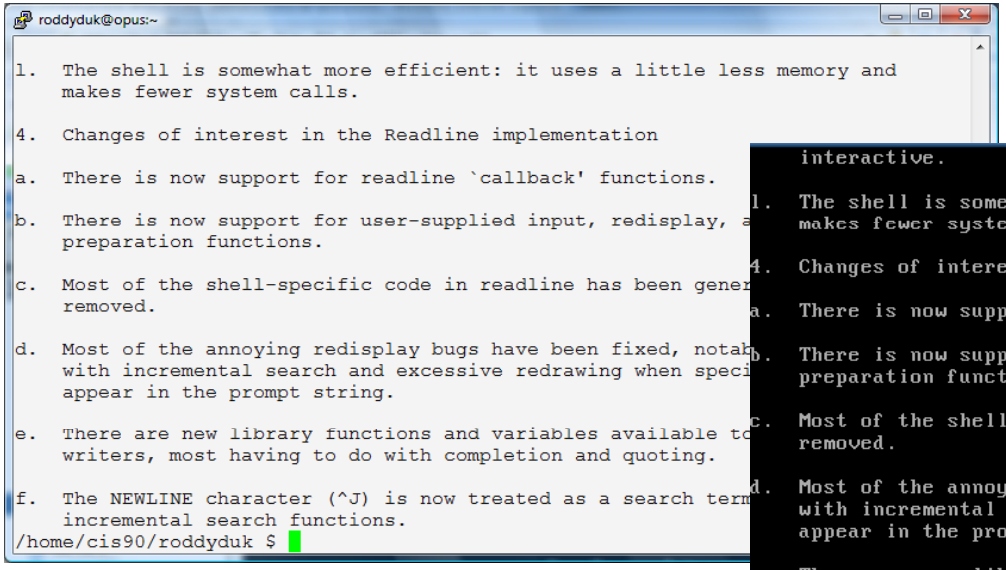
If you are using a terminal window (like PuTTY or in graphical mode) and the file is not too large you can just scroll back using the scroll bars.

cat command

concatenate or view text files

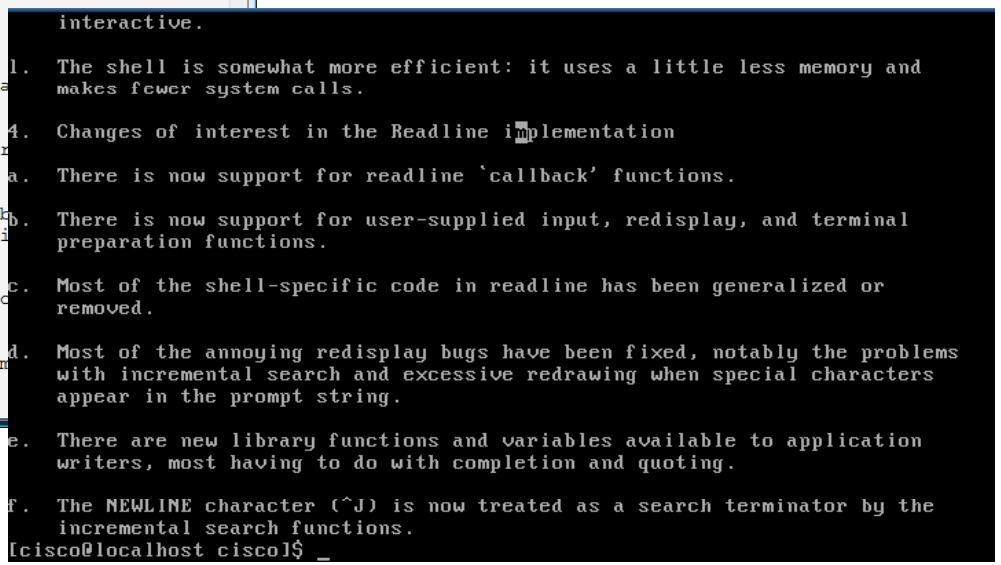
- Problem - If you **cat** really long files the text at the beginning is scrolled off and cannot be read.

- For example: **cat /usr/share/doc/bash-3.2/NEWS**



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

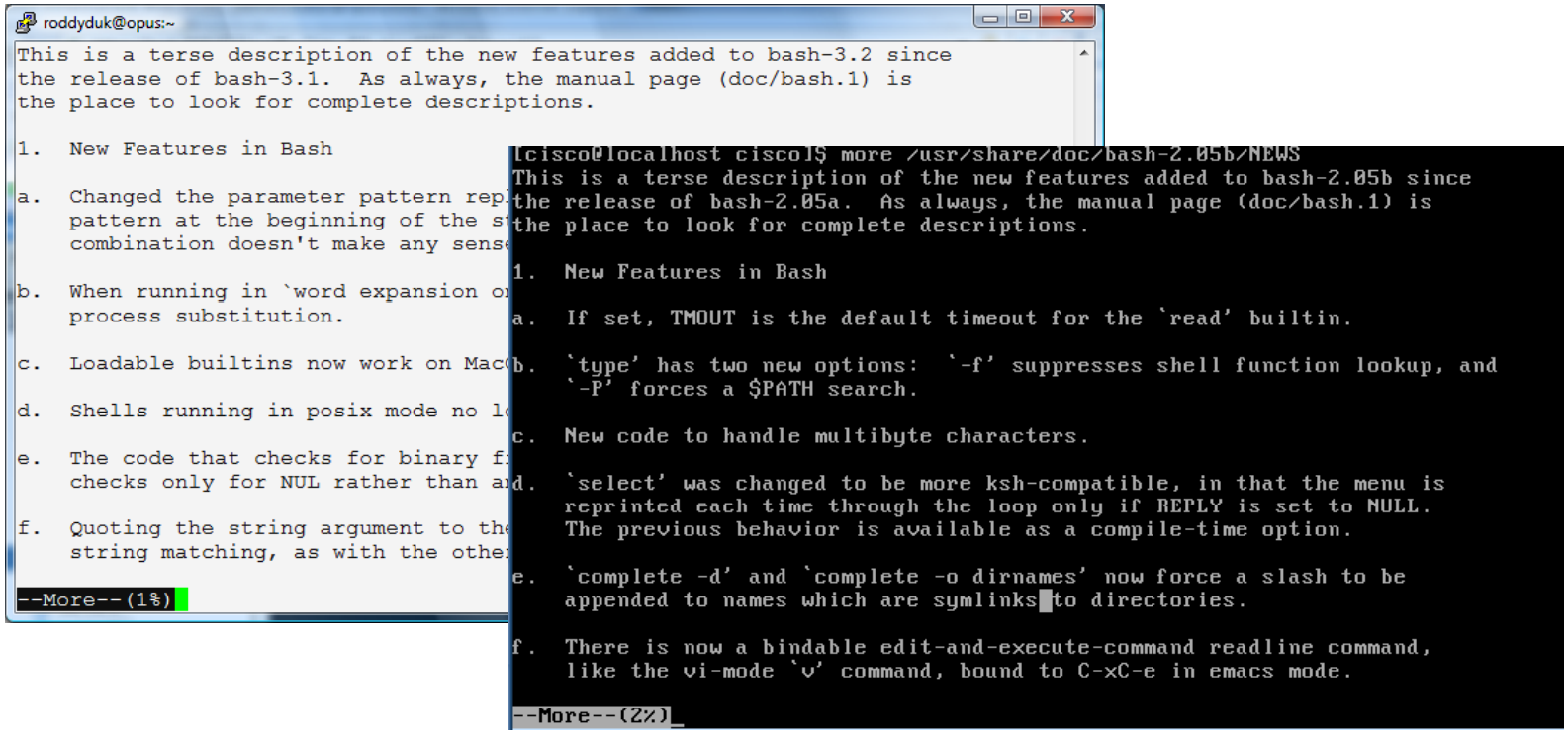
Virtual terminals have no scroll bars !



more command

filter for scrolling through long text files

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



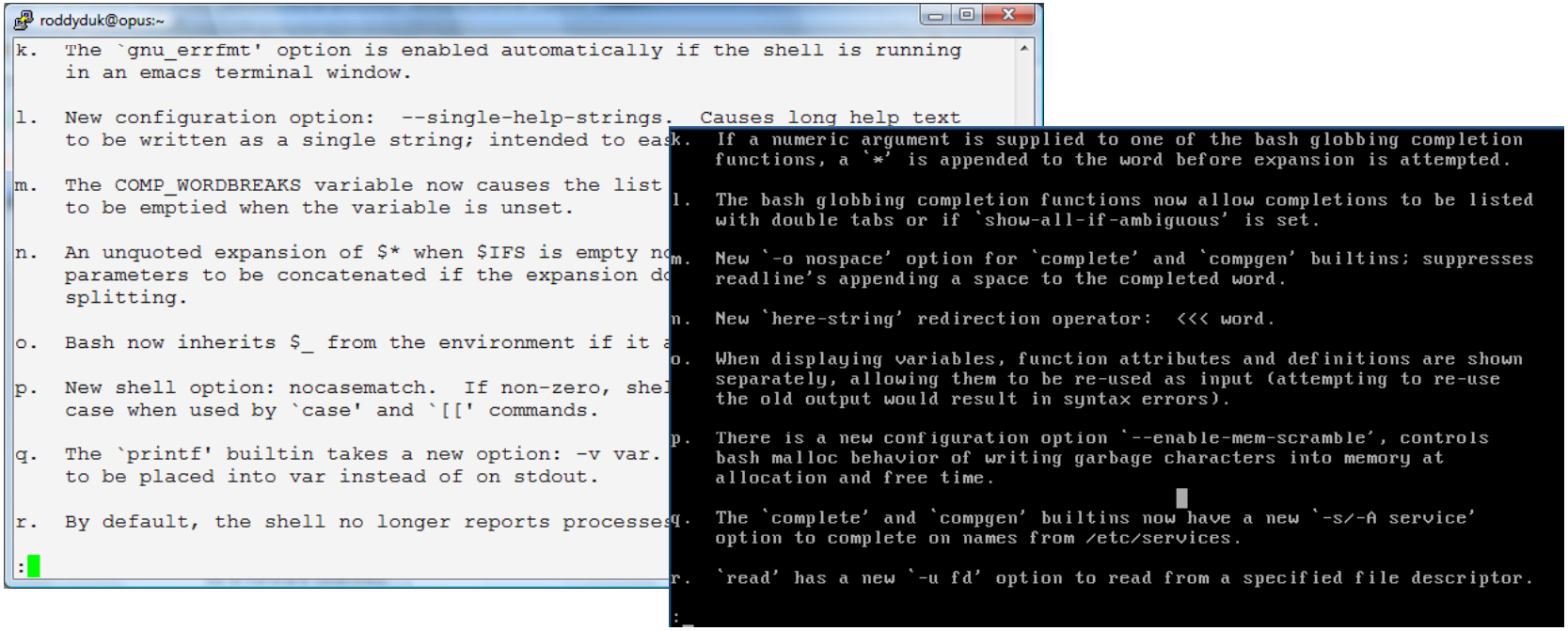
Use the space key to page forward and q to quit

less command

filter for scrolling through long text files

- Use the **less** command to page forward and backward through really long text files. (just like the man command works)

- For example: **less /usr/share/doc/bash-3.2/NEWS**

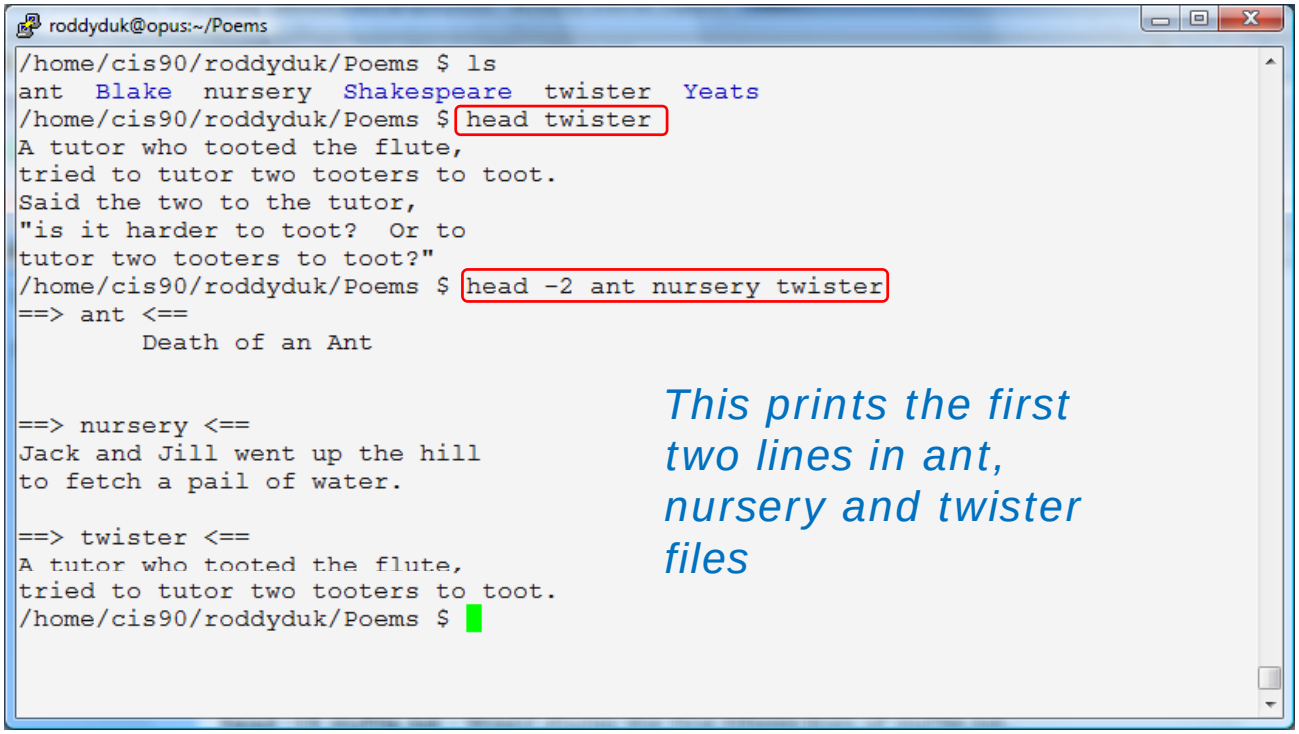


Use the pg up/dn and up/down arrows to move through text file. Use q to quit
(See the man page for many more options like searching)

head command

print just the beginning of a text file

- Use the **head** command to the first several lines of a file. Use the –number option to control the number of lines printed.
- For example:

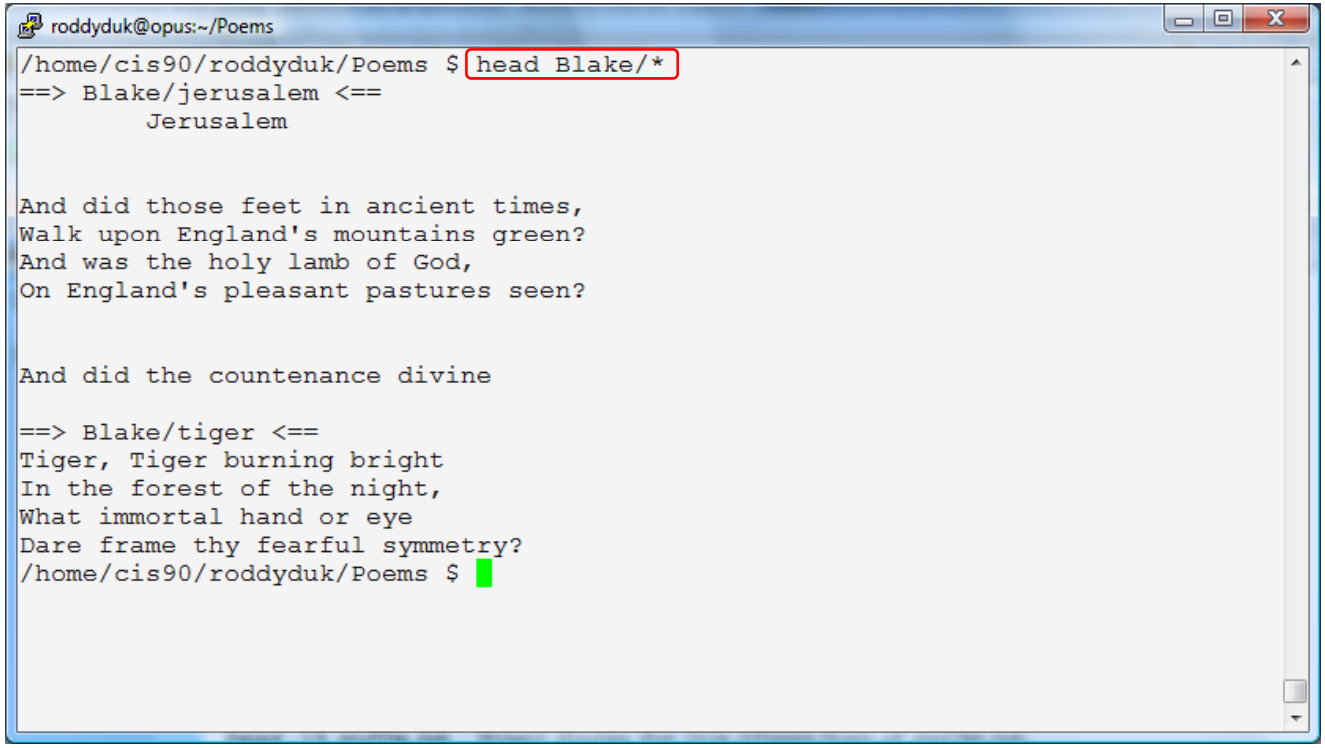


This prints the first two lines in ant, nursery and twister files

head command

print just the beginning of a text file

- Another example: **head Blake/*** to print headings of the files in the Blake directory:

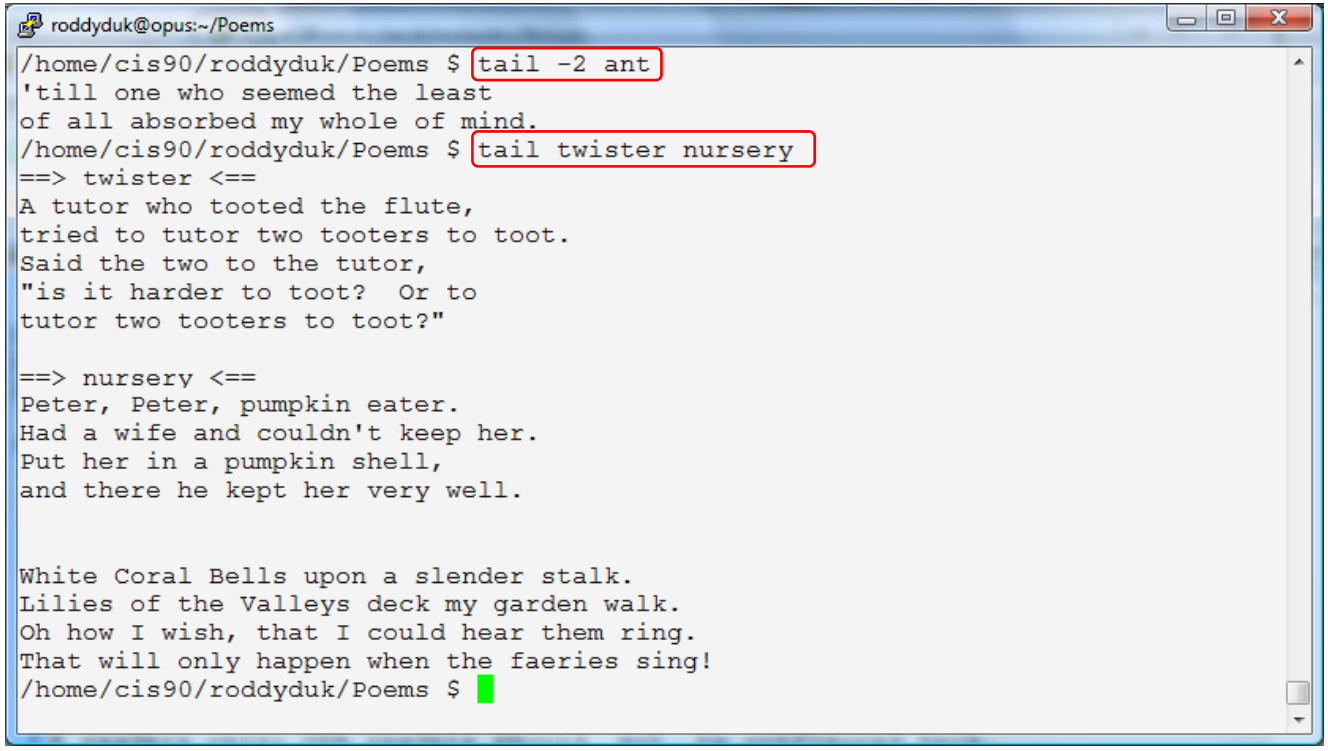


tail

print just the end of a text file

- Use the **tail** command to the print the last several lines of a file. Use the –number option to control the number of lines printed.

- For example:



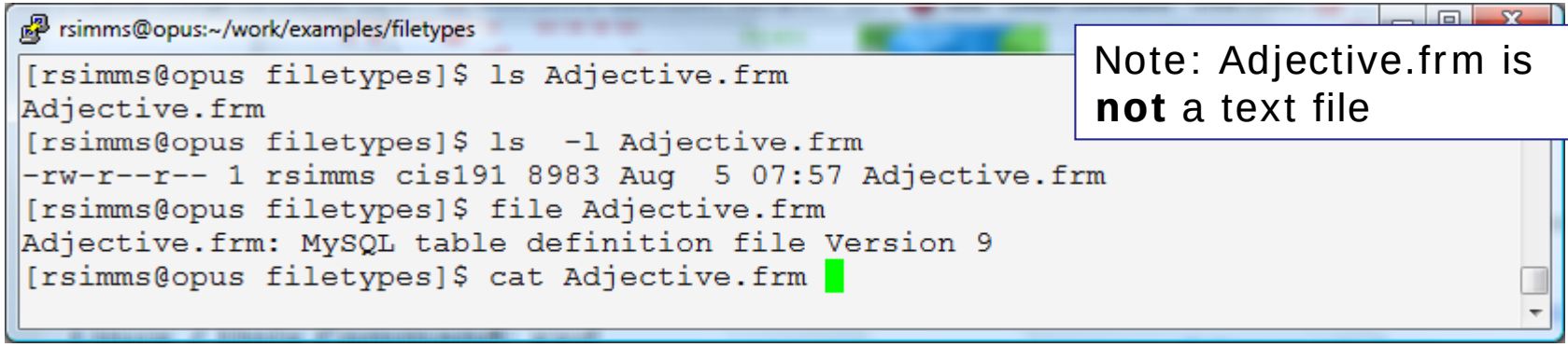
Class Exercise

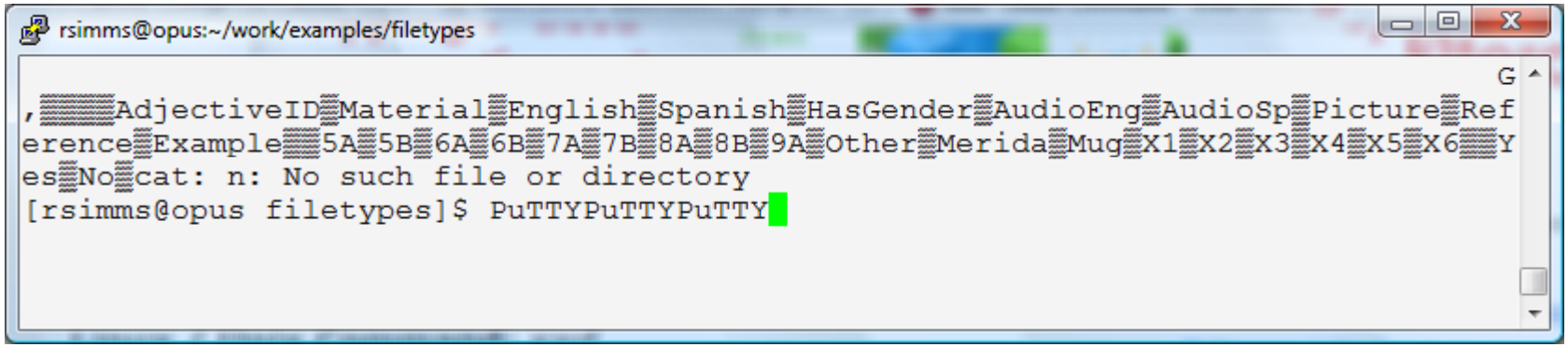
Navigate with cd, pwd, and ls

- Navigate to your Yeats directory
- Print the first line of the mooncat file with **head -1 mooncat**
- With one command print the first line of all files in the Yeats directory with **head -1 ***
- Use **tail -1 mooncat** to see the last line there
- Try **tail -1 *** to print the last lines in all Yeats poems. What happened?
- Try use **man tail**, review the n option, then try **tail –n 1 ***
- Navigate to your home directory use the **more** and **less** command to view bigfile

binary data files

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





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

xxd command

view hex dump of binary files

Example: **xxd Adjective.frm** (a MySql database schema file)

rsimms@opus:~/work/examples/filetypes

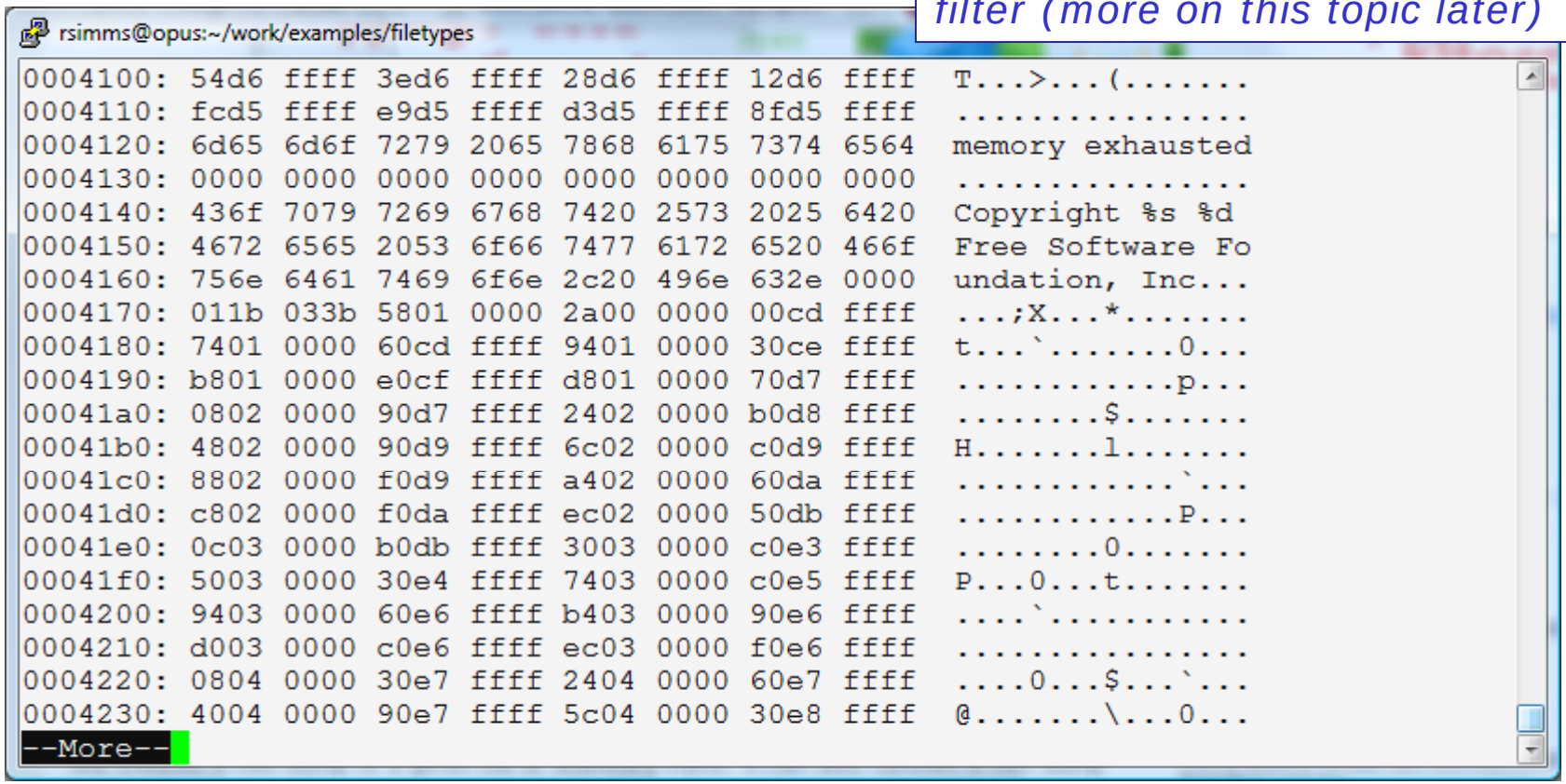
```
00021e0: 0008 8110 0001 f7c0 0000 0608 472c 0108  ............G,..
00021f0: 0000 0080 0000 00fd c000 0007 0847 2c01  .............G,.
0002200: 3401 0000 8000 0000 fdc0 0000 080a 0909  4...............
0002210: 0060 0200 0881 1000 02f7 c000 0009 0946  .`.............F
0002220: 2c01 6102 0000 8000 0000 fdc0 0000 0a08  ,.a.............
0002230: 472c 018d 0300 0080 0000 00fd c000 000b  G,..............
0002240: 0847 2c01 b904 0000 8000 0000 fdc0 0000  .G,.............
0002250: 0c0a 452c 01e5 0500 0080 0000 00fd c000  ..E,............
0002260: 000d 0847 2c01 1107 0000 8000 0000 fdc0  ...G,...........
0002270: 0000 ff41 646a 6563 7469 7665 4944 ff4d  ...AdjectiveID.M
0002280: 6174 6572 6961 6cff 456e 676c 6973 68ff  aterial.English.
0002290: 5370 616e 6973 68ff 4861 7347 656e 6465  Spanish.HasGende
00022a0: 72ff 4175 6469 6f45 6e67 ff41 7564 696f  r.AudioEng.Audio
00022b0: 5370 ff50 6963 7475 7265 ff52 6566 6572  Sp.Picture.Refer
00022c0: 656e 6365 ff45 7861 6d70 6c65 ff00 ff35  ence.Example...5
00022d0: 41ff 3542 ff36 41ff 3642 ff37 41ff 3742  A.5B.6A.6B.7A.7B
00022e0: ff38 41ff 3842 ff39 41ff 4f74 6865 72ff  .8A.8B.9A.Other.
00022f0: 4d65 7269 6461 ff4d 7567 ff58 31ff 5832  Merida.Mug.X1.X2
0002300: ff58 33ff 5834 ff58 35ff 5836 ff00 ff59  .X3.X4.X5.X6...Y
0002310: 6573 ff4e 6fff 00                        es.No..
[rsimms@opus filetypes]$
```

xxd command

view hex dump of binary files

Example: **xxd /bin/pwd | more**

For long files, the output of xxd can be "piped" into the more filter (more on this topic later)



Class Exercise

- **cd /home/cis90/depot/filetypes/**
- **xxd Adjective.frm**
- **xxd Adjective.frm | more**

Using xxd to dump contents of binary file

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*

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