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

- Practice test 1 -

- CCC Confer wall paper / quiz emailed -

- Pick up Polycom phone/extension mics -
- Check that headset is charged –
- Wireless lapel mic backup battery -
- Backup slides, CCC info, handouts on flash drive -

CIS 90 - Lesson 5

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


Dieskau

Jonathan

Ana

David


Obie

Dave

Cole

Corey

Nancy

Ryan

Elia


Tasha

Darren

Scott


Devin

Everett

Juan

Raven

Rogan


Mike


Mook


Melissa

Cameron

Jose

Jeff

Matt

Kenneth

Ousmane

Ian


Solomon

Henry

Matthew

Mason

Chan

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

[ ] Has the phone bridge been added?

[ ] Is recording on?

[ ] Does the phone bridge have the mike?

[ ] Share Slides, putty (rsimms x 3), Chrome, VLab

[ ] Disable spelling on PowerPoint

First Minute Quiz

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

email answers to: risimms@cabrillo.edu

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

Review

Objectives	Agenda
• Review Lessons 1-4 • Practice skills • Learn about filename expansion characters	• Quiz • Questions from last week • Test tips • Everything is a file • More filename expansion characters • Lots of review • Wrap up

Questions

Previous material and assignment

- Questions on previous material?
- Questions on any of the labs?
- Note: Lab 4 due today, email it to me at risimms@cabrillo.edu
  - Be sure and read the forum before turning in Lab 4 (or any lab for that matter).
  - Remember, you can re-submit labs as many times as you wish up till the deadline. The most recent submittal gets graded.

Flashcard Teams

debian	redhat	SUSE	CentOS	ubuntu	
ahrmat90	lowmic90	cofcol90	milmic90	plajua90	gueous90
blerav90	macrya90	colabd90	olscam90	porjon90	helrog90
bodian90	maxsco90	deltas90	pacnan90	pummas90	hovdav90
bunsol90	mcidar90	doucor90	phacha90	rafdav90	huljef90
cheken90	milhen90	flamat90	plajos90	reedie90	jimmel90
shidev90			veleli90		varana90

Everyone needs to be on CCC Confer today, please use your Opus username. Test that you can chat to the chat window

The link to the Virtual Classroom is on the CIS 90 Calendar page

Housekeeping

1. Only 3 passwords cracked so far!
2. Ignore Q14-practice email ... oops!
3. No lab assignment this week so you can prepare for the test next week
4. Practice test is available.
5. The first half of next week's Lesson 6 will be new lesson material. The second half will be the test covering material in Lessons 1-5.

Test next week

- 30 points, plus some extra credit
- 5 flashcard questions
  - Taken directly from the flashcards on the web site
- 25 operational questions
  - You can verify your answers using Sun, VLab VMs and Opus
- Open book, open notes, open computer
- To be taken during the last half of class
- Should take about 60-90 minutes, however if you need extra time, you can turn it in no later than 11:59PM.
- PDF form format. Fill out the form, save it and email it as an attachment to the instructor when finished cc'ing yourself.

Tips for Test

How to prepare for the test:

- Review slides for Lessons 1-5 (download and make sure you know how to electronically search PDFs)
- DO THE PRACTICE TEST
- Compare your practice test answers and methods used with others on the forum
- MAKE NOTES ON THE METHOD USED to answer each question so you can use them again on the real test
- Go through the Lesson 1-5 flashcards till you feel comfortable with the material
- Practice, practice, practice ... repeating Labs 1-4 never hurts!

What command will do "blah, blah, blah" questions

Tips on how to answer questions on lab assignments and tests

What command will do "blah, blah, blah" questions:

Examples:

- What **ls** command allows you to see the permissions of your home directory while you are in your home directory?
- What command will give you a prompt showing your current working directory path and a $?
- What command allows you to see hidden files in your current directory?

Tip: Always use Opus to test your answers for these kinds of questions. ***I will!*** *If your command doesn't work on Opus it won't be the right answer!*

Example

What **ls** command allows you to see the permissions of your home directory while you are in your home directory?

```
/home/cis90/simben $ ls -l
total 392
-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    606 Feb 29 22:17 dead.letter
-rw-r--r-- 1 simben90 cis90      0 Jul 20  2001 empty
d--------- 2 simben90 cis90   4096 Feb  1  2002 Hidden
< snipped >
-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  25390 Feb 29 22:18 uhistory

-rw-r--r-- 1 simben90 cis90    352 Mar  5 08:24 what_am_i
/home/cis90/simben $
```

Nope, that didn't work. *We got permissions of all the files in the directory but we didn't get the permissions of the directory itself!*

Example (continued)

What **ls** command allows you to see the permissions of your home directory while you are in your home directory?

```
/home/cis90/simben $ ls -dl /home/cis90/simben
drwxr-xr-x 10 simben90 cis90 4096 Mar  1 10:15

/home/cis90/simben $ ls -dl ~
drwxr-xr-x 10 simben90 cis90 4096 Mar  1 10:15

/home/cis90/simben $ ls -dl .
drwxr-xr-x 10 simben90 cis90 4096 Mar  1 10:15 .

/home/cis90/simben $ ls -dl $HOME
drwxr-xr-x 10 simben90 cis90 4096 Mar  1 10:15

/home/cis90/simben $ ls -dl
drwxr-xr-x 10 simben90 cis90 4096 Mar  1 10:15 .
```

The -d option instructs the ls command not to descend into the directory. Any of the commands above would be correct.

Example

What command will give you a prompt showing your current working directory path and a $?

```
/home/cis90/simben $ PS1=blah
blah
blahPS1="/home/cis90/simben $ "
/home/cis90/simben $
/home/cis90/simben $ cd ..
/home/cis90/simben $ cd
/home/cis90/simben $
/home/cis90/simben $ echo $PS1
/home/cis90/simben $
```

***Nope, that didn't work**. The prompt doesn't change after changing to another directory*

Example (continued)

What command will give you a prompt showing your current working directory path and a $?

```
/home/cis90/simben $ PS1=blah
blah
blahPS1="PWD $ "
PWD $
PWD $ echo $PS1
PWD $
```

***Nope, that didn't work**. A $ in front of the variable name is required to use its value.*

Example (continued)

What command will give you a prompt showing your current working directory path and a $?

```
PWD $ PS1=blah
blah
blahPS1="$PWD $ "
/home/cis90/simben $ cd ..
/home/cis90/simben $ cd
/home/cis90/simben $
/home/cis90/simben $ echo $PS1
/home/cis90/simben $
```

***Better, but still didn't work**. The prompt is still not changing after cd'ing to another directory.*

We need to block bash from expanding the $PWD variable when it's being set.

Example (continued)

What command will give you a prompt showing your current working directory path and a $?

```
/home/cis90/simben $ PS1=blah
blah
blahPS1='$PWD $ '
/home/cis90/simben $ cd ..
/home/cis90 $ cd
/home/cis90/simben $
/home/cis90/simben $ echo $PS1
$PWD $
```

Touchdown! That worked!

The single quotes prevent bash from expanding $PWD when setting the PS1 variable.

It is not expanded till the prompt is actually generated for the next command.

Example

What command allows you to see hidden files in your current directory?

```
/home/cis90/simben $ ls
bigfile       lab01.graded     Lab2.1          mission    small_town  uhistory
bin           lab01-submitted  letter          Poems      spellk      what_am_i
dead.letter   lab02.graded     log             proposal1  text.err
empty         lab03.graded     mbox            proposal2  text.fxd
Hidden        Lab2.0           Miscellaneous   proposal3  timecal
```

Nope, that didn't work!

```
/home/cis90/simben $ ls -a
.              dead.letter      Lab2.0          .mozilla    .ssh
..             .emacs           Lab2.1          .plan       text.err
.bash_history  empty            .lesshst        Poems       text.fxd
.bash_logout   Hidden           letter          proposal1   timecal
.bash_profile  lab01.graded     log             proposal2   uhistory
.bashrc        lab01-submitted  mbox            proposal3   .viminfo
bigfile        lab02.graded     Miscellaneous   small_town  what_am_i
bin            lab03.graded     mission         spellk
```

Yes, that worked!

How many arguments or “parse this command” questions

Tips on how to answer questions on lab assignments and tests

How many arguments or "parse this command" questions

Example: The shell performs file name expansion during the Parse step. When a user types the command: **file /v*/l??/*o*.[14]** on Opus, how many arguments get passed to the **file** command? What specifically are those arguments?

Tip: Use the echo command to preview how the shell will expand arguments containing metacharacters.

Example

The shell performs file name expansion during the Parse step. When a user types the command: **file /v*/l??/*o*.[14]** on Opus, how many arguments get passed to the **file** command? What specifically are those arguments?

Tip: Use the echo command to preview how the shell will expand arguments containing metacharacters.

```
/home/cis90ol/simmsben $ echo /v*/l??/*o*.[14]
/var/log/boot.log.1 /var/log/boot.log.4 /var/log/cron.1 /var/log/cron.4
  /var/log/maillog.1 /var/log/maillog.4 /var/log/spooler.1
  /var/log/spooler.4 /var/log/yum.log.1
```

*The shell will expand **/v*/l??/*o*.[14]** into the 9 arguments shown above*

Example

```
/home/cis90ol/simmsben $ file /v*/l??/*o*.[14]
/var/log/boot.log.1: empty
/var/log/boot.log.4: empty
/var/log/cron.1:     writable, regular file, no read permission
/var/log/cron.4:     writable, regular file, no read permission
/var/log/maillog.1:  writable, regular file, no read permission
/var/log/maillog.4:  writable, regular file, no read permission
/var/log/spooler.1:  empty
/var/log/spooler.4:  empty
/var/log/yum.log.1:  ASCII text
/home/cis90ol/simmsben $
```

*The shell expands **/v*/l??/*o*.[14]** into 9 arguments, each a specific file pathname, to be processed by the file command.*

The file command never sees the metacharacters typed by the user, it just sees the 9 arguments with are specific file pathnames.

Absolute/relative pathname questions:

Tips on how to answer questions on lab assignments and tests

Absolute/relative pathname questions:

Examples:

- What is the relative pathname from your home directory to the **date** command?

- What is the absolute path to the sonnet1 file in your Shakespeare directory?

*Tip: Use the **ls** command with tab completion to verify your absolute or relative pathnames*

Example

What is the relative pathname from your home directory to the **date** command?

```
/home/cis90/simmsben $ type date
date is /bin/date

/home/cis90/simben $ ls ../
ahrmat/         colabd/         huljef/         olscam/         rodduk/
answers/        deltas/         jimmel/         pacnan/         shidev/
.bash_profile   depot/          lowmic/         phacha/         simben/
bin/            doucor/         macrya/         plajos/         varana/
blerav/         flamat/         maxsco/         plajua/         veleli/
bodian/         gueous/         mcidar/         porjon/
bunsol/         guest/          milhen/         pummas/
cheken/         helrog/         milhom/         rafdav/
cofcol/         hovdav/         milmic/         reedie/
/home/cis90/simben $ ls ../../
backup/       cis191/       cis90/        guest/        rick/         turnin/
cis164/       cis192/       cis98/        jimg/         rsimms/       .Xauthority
cis172/       cis193/       gerlinde/     mikki/        ryan/
/home/cis90/simben $ ls ../../../
.autofsck   etc/         media/      opt/        selinux/    tmp/
bin/        home/        misc/       proc/       srv/        u/
boot/       lib/         mnt/        root/       sys/        usr/
dev/        lost+found/  net/        sbin/       tftpboot/   var/
/home/cis90/simben $ ls ../../../bin/date
../../../bin/date
/home/cis90/simben $
```

Tap tab key twice to see what is in that directory

No errors so this relative pathname is GOOD!

Example

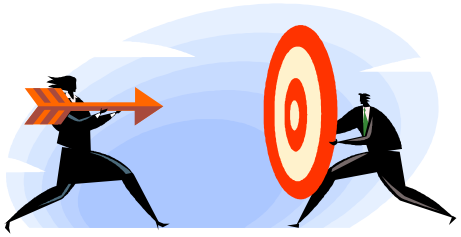
What is the absolute path to the sonnet1 file in your Shakespeare directory?

```
/home/cis90/simben $ ls /
.autofsck   etc/        media/      opt/        selinux/    tmp/
bin/        home/       misc/       proc/       srv/        u/
< snipped >
/home/cis90/simben $ ls /home/
backup/      cis191/      cis90/       guest/       rick/        turnin/
< snipped >
/home/cis90/simben $ ls /home/cis90/
ahrmat/        colabd/        huljef/        olscam/        rodduk/
answers/       deltas/        jimmel/        pacnan/        shidev/
.bash_profile  depot/         lowmic/        phacha/        simben/
< snipped >
cofcol/        hovdav/        milmic/        reedie/
/home/cis90/simben $ ls /home/cis90/simben/
.bash_history    lab01.graded     Miscellaneous/   .ssh/
< snipped >
.bashrc          lab03.graded     .plan            timecal
bigfile          Lab2.0/          Poems/           uhistory
< smipped >
Hidden/          mbox             spellk
/home/cis90/simben $ ls /home/cis90/simben/Poems/
ant          Blake/       nursery      Shakespeare/ twister      Yeats/
/home/cis90/simben $ ls /home/cis90/simben/Poems/Shakespeare/sonnet
sonnet1   sonnet11  sonnet17  sonnet26  sonnet35  sonnet5   sonnet9
sonnet10  sonnet15  sonnet2   sonnet3   sonnet4   sonnet7
/home/cis90/simben $ ls /home/cis90/simben/Poems/Shakespeare/sonnet1
/home/cis90/simben/Poems/Shakespeare/sonnet1
```

Tap tab key twice to see what is in that directory

No errors so this absolute pathname is GOOD!

Relative/Absolute Pathname Target Practice



```
[rsimms@opus bin]$ ./randomFile
file 94542 of 181093 is:

Random absolute pathname: /usr/share/hplip/base/exif.pyc

Directory: /usr/share/hplip/base
Filename: exif.pyc

Continue with tree of parent directory? (Press Enter to continue))
```

*Now try and hit the target using **ls –li**, **file** and **head** (if text file) commands using absolute and relative pathnames*

Everything is a file (new)

Everything is a file in UNIX
(even a terminal)

- A terminal
- A file
- A hard drive
- A hard drive partition
- A CD
- A partition on a USB flash drive
- Kernel run-time information

Implemented as files in UNIX

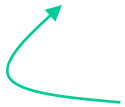
Everything is a file in UNIX
(even a terminal)

• A terminal	*e.g. /dev/pts/2*
• A file	*e.g. /home/cis90/simmsben/letter*
• A directory	*e.g /home/cis90/*
• A hard drive	*e.g. /dev/sda*
• A hard drive partition	*e.g. /dev/sda1*
• A CD	*e.g. /dev/cdrom*
• A partition on a USB flash drive	*e.g. /dev/sdb2*
• Kernel run-time information	*e.g. /proc/sys/kernel/hostname*

Everything is a file
(even a terminal)

```
/home/cis90/simmsben $ tty
/dev/pts/1
```

*Use the **tty** command to identify the specific terminal device being used*



Note this device is identified using a pathname

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

Use the TERM variable to identify the specific type of terminal being used

Everything is a file
(even a terminal)

```
/home/cis90/simmsben $ tty
/dev/pts/1
```

Show which terminal you are using

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

Show what kind of terminal you are using

```
/home/cis90/simmsben $ who
simmsben pts/1        2010-09-29 07:38 (dsl-49-64-10-90.dhcp.cruzio.com)
srecklau pts/2        2010-09-29 06:06 (62.143.60.194)
rsimms   pts/4        2010-09-29 06:47 (dsl-49-64-10-90.dhcp.cruzio.com)
```

Use who to see who is logged in

```
/home/cis90/simmsben $ ls -l /dev/pts/*
crw--w---- 1 simmsben tty 136, 1 Sep 29 07:45 /dev/pts/1
crw--w---- 1 srecklau tty 136, 2 Sep 29 07:44 /dev/pts/2
crw--w---- 1 rsimms   tty 136, 4 Sep 29 06:48 /dev/pts/4
```

Do a long listing to see all the terminal devices in use

Notice the owner is someone who has logged in

Notice the file type is "c" which is a character device file

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 device files	mknod
b	block device files	mknod

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

Everything is a file in UNIX
(even a terminal)

Nice things about files

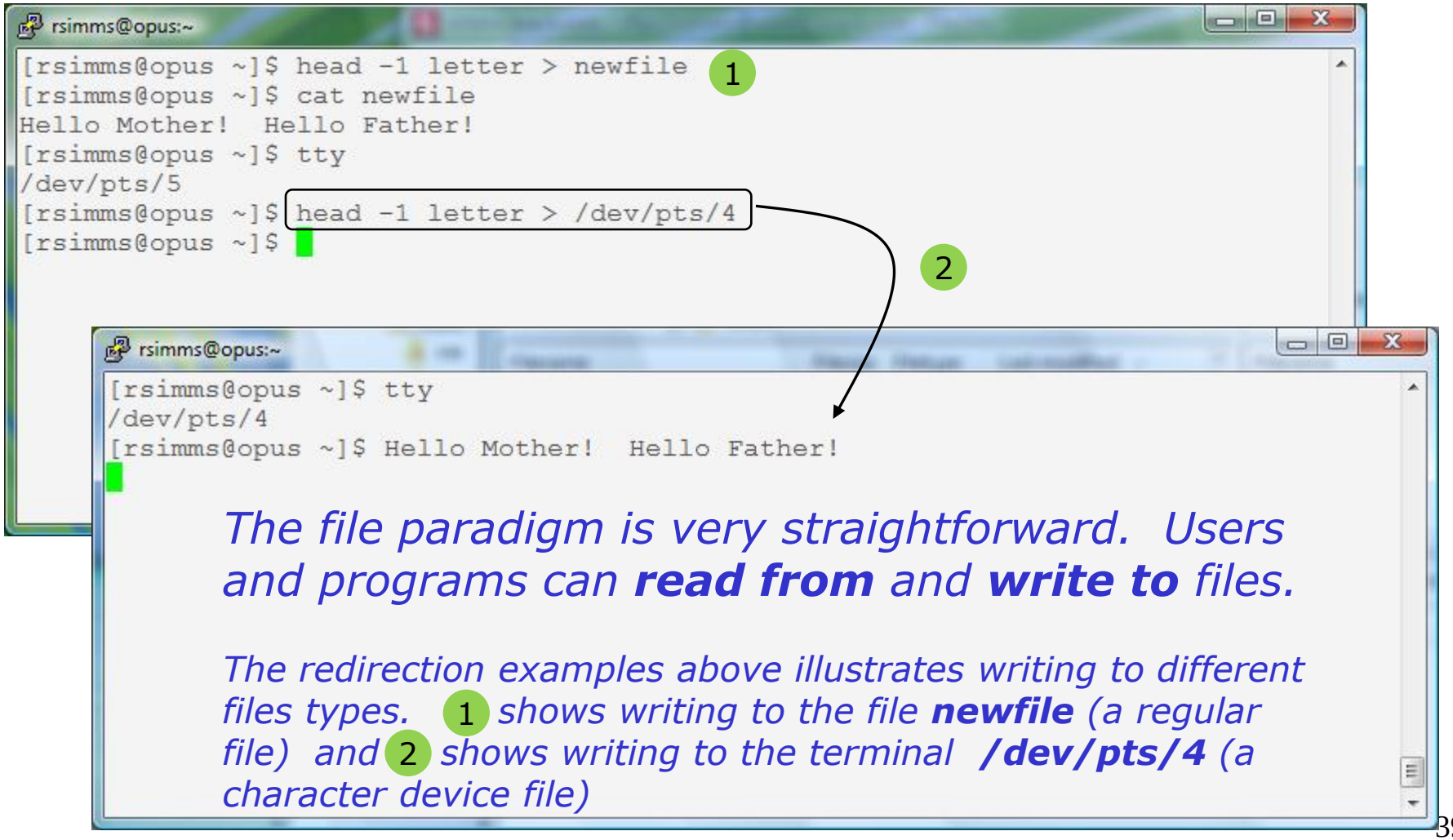
- you can write to them

```
[rsimms@opus ~]$ echo "Rich was here" > myfile
```

- and read from them

```
[rsimms@opus ~]$ cat myfile
Rich was here
```

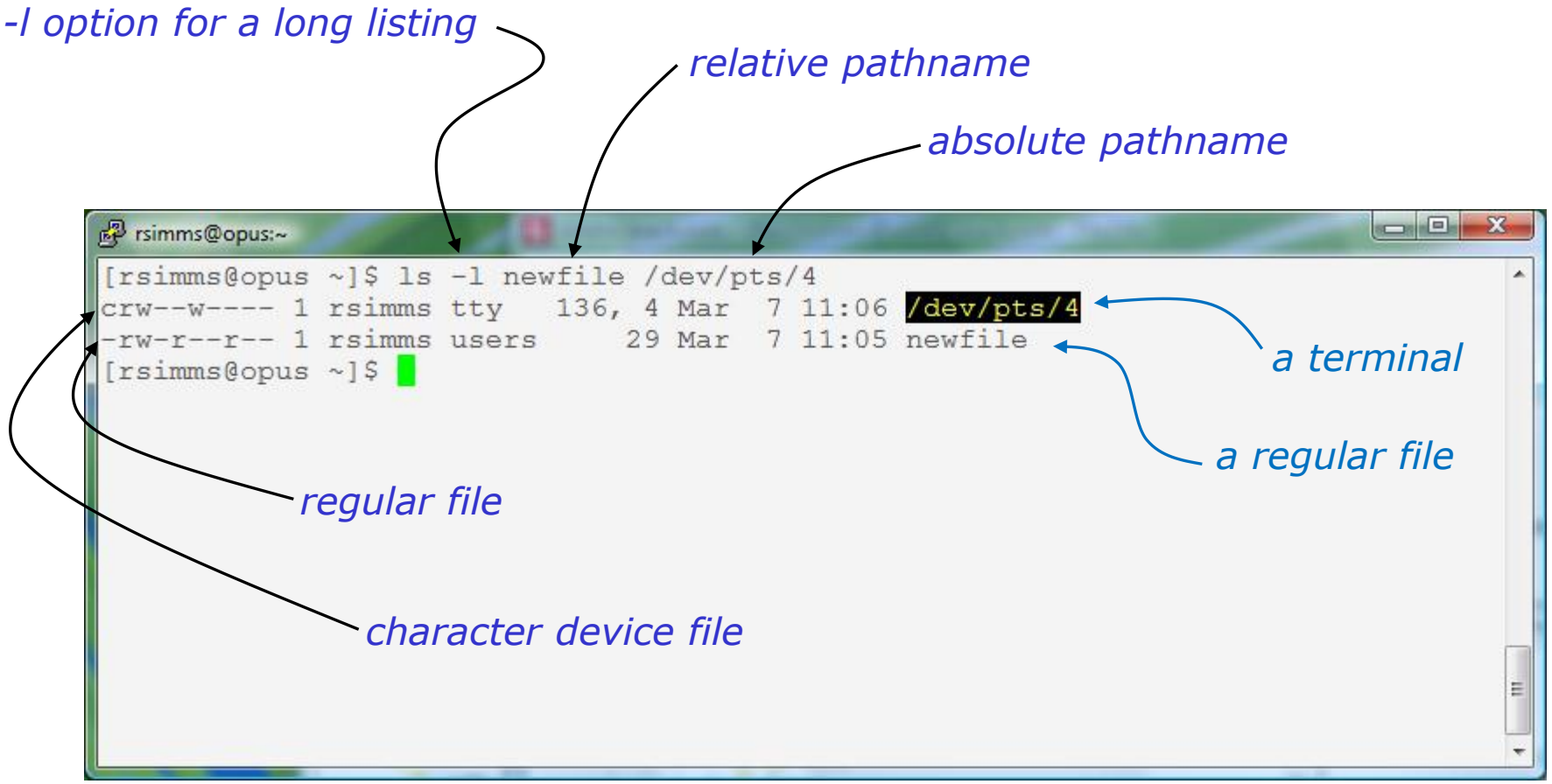
Everything is a file in UNIX
(even a terminal)



*The file paradigm is very straightforward. Users and programs can **read from** and **write to** files.*

*The redirection examples above illustrates writing to different files types. 1 shows writing to the file **newfile** (a regular file) and 2 shows writing to the terminal **/dev/pts/4** (a character device file)*

Everything is a file
(even a terminal)



Class Exercise

- Login into Opus using Putty
- Use **echo "Hello Hugo" > myfile**
- Print your new file with **cat myfile**
- Open a second Putty session and login into Opus
- You should have two terminals now (two Putty windows)
- Use **tty** to identify your terminals
- In one terminal use **echo "Hello Hugo" > /dev/pts/xx** where xx is your other terminal

File Name Expansion (new)

Filename Expansion Metacharacters

More metacharacters for making file name wildcards

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

? matches any single character in any of your current directory's filenames.

[] matches any single character contained within the brackets.

Filename Expansion Metacharacters

* ? []

Use ls to show non-hidden filenames in the current directory

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

*The shell will replace * with the non-hidden filenames in the current directory*

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

*The **echo** command above never sees the *, instead it gets all the matched filenames as arguments .*

Metacharacters
*

*Note the * metacharacter by itself does not match any hidden files in your current working directory*

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


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

Filename Expansion Metacharacters

* ? []

When the shell parses the following head command:

```
head *
```

*It expands the * with the names of all non-hidden files in the current directory to become:*

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

(all on one line)

Filename expansion happens during the shell parsing step, before the command is even located or executed.

1) Prompt
2) **Parse**
3) Search for program (along the path)
4) Execute program
5) Nap (wait till process is done)
6) Repeat

Filename Expansion Metacharacters
* ? []

Note, DOS uses *.* to match all files.

BUT, this is NOT true in UNIX

```
/home/cis90/simmsben $ echo *.*
Lab2.0 Lab2.1 text.err text.fxd
```

*Instead, *.* is expanded to match all files in the current directory containing a "."*

Filename Expansion Metacharacters
* ? []

What command would classify all files in the parent directory that start with p?

```
/home/cis90/simben $ file ../p*
../pacnan: directory
../phacha: directory
../plajos: directory
../plajua: directory
../porjon: directory
../pummas: directory
/home/cis90/simben $
```

Filename Expansion Metacharacters
* ? []

What command would list all three letter filenames in /bin

```
/home/cis90/simben $ ls /bin/???
/bin/awk  /bin/cut  /bin/how  /bin/raw  /bin/rvi
/bin/cat  /bin/env  /bin/ksh  /bin/red  /bin/sed
/bin/csh  /bin/gsh  /bin/pwd  /bin/rpm  /bin/tar
/home/cis90/simben $
```

Filename Expansion Metacharacters
* ? []

From your home directory, what command would print the first lines of all Shakespeare sonnets ending in a 2 or 5?

```
/home/cis90/simben $ head -n 1 Poems/Shakespeare/*[25]
==> Poems/Shakespeare/sonnet15 <==
When I consider every thing that grows

==> Poems/Shakespeare/sonnet2 <==
When forty winters shall besiege thy brow,

==> Poems/Shakespeare/sonnet35 <==
Whoever hath her wish, thou hast thy Will,

==> Poems/Shakespeare/sonnet5 <==
Those hours that with gentle work did frame
/home/cis90/simben $
```

Filename Expansion Metacharacters
* ? []

What commands are there in /usr/bin that start with a “n” or “m”, are 5 letters long and end with a “p”?

```
/home/cis90/simben $ ls /usr/bin/[nm]???p
/usr/bin/mcomp  /usr/bin/nohup
/home/cis90/simben $
```

Filename Expansion Metacharacters
* ? []

For the command:

`file /usr/share/man/*/[ap]??.8.gz`

What arguments are actually getting passed to the **file** command to process?

```
/home/cis90/simben $ echo /usr/share/man/*/[ap]??.8.gz
/usr/share/man/man8/arp.8.gz /usr/share/man/man8/atd.8.gz
/usr/share/man/man8/pam.8.gz /usr/share/man/man8/pup.8.gz
/usr/share/man/man8/pvs.8.gz
/home/cis90/simben $
```

Tip: Use echo to expand complicated filenames containing multiple filename expansion characters

Class Exercise

- Change to your home directory
- Use the **file** command on all files starting with prop

```
file prop*
```

- Print the headings of all files starting with l or t

```
head [lt]*
```

- Use **ls** command to list directories two levels up that start with cis and are followed by three more characters

```
ls -dl ../../cis???
```

- Make up your own wildcard using *, [], and ? in one command

Command Review

*Use the **man** command or google for the details*

New commands:

cal	- show calendars
clear	- clear the terminal screen
exit	- terminate your shell and log off
history	- show previous commands
hostname	- show the name of the computer being accessed
id	- show user and group id information
ps	- show processes (loaded programs) being run
ssh	- secure login to a remote system
uname	- show OS name
tty	- show terminal information
who	- show who else is logged on
who am i	- Identifies which login session you are using
Ctrl-Alt-F1 to Ctrl-Alt-F7	- Change between terminals and X windows (graphics)

New Files and Directories:

VMware:

Ctrl-Alt	- to move mouse cursor out of VM

*Use the **man** command or google for the details*

New commands:

apropos	- search for string in whatis database
bc	- binary calculator
cat	- print file(s)
cd	- change directory
echo	- print text
env	- show shell environment variables
info	- online documentation with hot links
file	- show file information
ls	- show directory contents
passwd	- change password
set	- show (or set) shell variables
type	- show command location in path
man	- manual page for a command
whatis	- command summary

New Files and Directories:

/etc/passwd	- user accounts
/etc/shadow	- encrypted passwords
/bin	- directory of commands
/sbin	- directory of superuser commands
/usr/bin	- directory of commands, tools and utilities
/usr/sbin	- directory of superuser commands, tools and utilities

New commands:

mail	- UNIX mail
?	print these commands
p <message list>	print messages
n	goto and print next message
e <message list>	edit messages
d <message list>	delete messages
s <message list> file	save (append) messages to file
u <message list>	undelete messages
R <message list>	reply to sender(s)
r <message list>	reply to all
m <user list>	mail to specific users
q	quit, saving read messages to local mbox file
x	quit, mark all mail as unread and undeleted.
h	print out active message headers
mesg	- Enable or disable writes to your terminal
write	- Write message to another user

New Files and Directories:

/var/mail	- Message store for mail
/var/mail/*username*	- Incoming mailbox for *username*
mbox	- File in users home directory where read messages are archived to

*Use the **man** command or google for the details*

*Use the **man** command or google for the details*

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	View (hex dump) binary/data files

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*

Round 1

ahrmat90
blerav90
bodian90
bunsol90
cheken90
shidev90

1-2 2

lowmic90
macrya90
maxsco90
mcidar90
milhen90

3-4 2

cofcol90
colabd90
deltas90
doucor90
flamat90

5-6 2

milmic90
olscam90
pacnan90
phacha90
plajos90
veleli90

7-8 2

plajua90
porjon90
pummas90
rafdav90
reedie90

9-10 2

gueous90
helrog90
hovdav90
huljef90
jimmel90
varana90

11-12 2

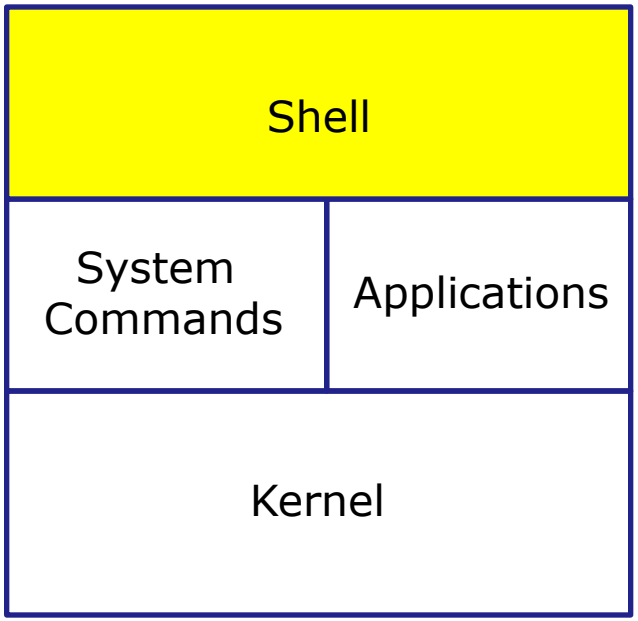
Flashcards
L1=18
L2=22
L3=5
L4=26
L5=4
Total=75

Rules

- Chat window belongs to team that is up (no one else can use)
- "Final Answer" must be from someone on team that hasn't answered yet
- All team members can help each other and suggest answers

Command line Prompt Parse (review)

Life of the Shell



1) **Prompt** for a command
2) **Parse** (interpret metacharacters, expand file names and dissect command line into options and arguments)
3) **Search** for program (along the path)
4) **Execute** program by loading into memory (becomes a process), hookup input and outputs, and pass along command line options and arguments.
5) **Nap** (wait till process is done)
6) **Repeat**

Command Syntax

Command	Options	Arguments	Redirection

Command – is the name of an executable program file.
Options – various options which control how the program will operate.
Arguments – the objects the command is directed to work upon.
Redirection – The default input stream (stdin) is from the console keyboard, the default output (stdout) and error (stderr) streams go to the console screen. Redirection can modify these streams to other files or devices.

Command Syntax

Shell prints this to prompt user to enter a command

Shell parses this command line

***Options** modify the behavior of the command*

***Arguments** are what the command works upon*

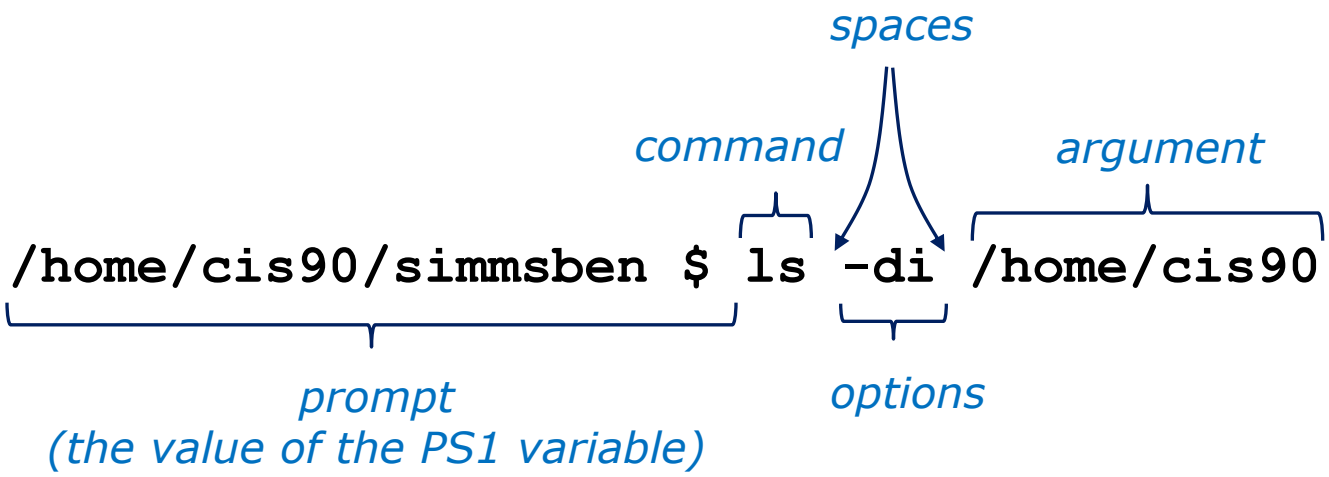
***Redirection** is covered later in the course*

Examples

```
/home/cis90/simmsben $
/home/cis90/simmsben $ ls
/home/cis90/simmsben $ ls -l
/home/cis90/simmsben $ ls -lt
/home/cis90/simmsben $ ls -lt Poems/
/home/cis90/simmsben $ ls -lt Poems/ bin/
/home/cis90/simmsben $ ls -lt Poems/ bin/ > mylist
```

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

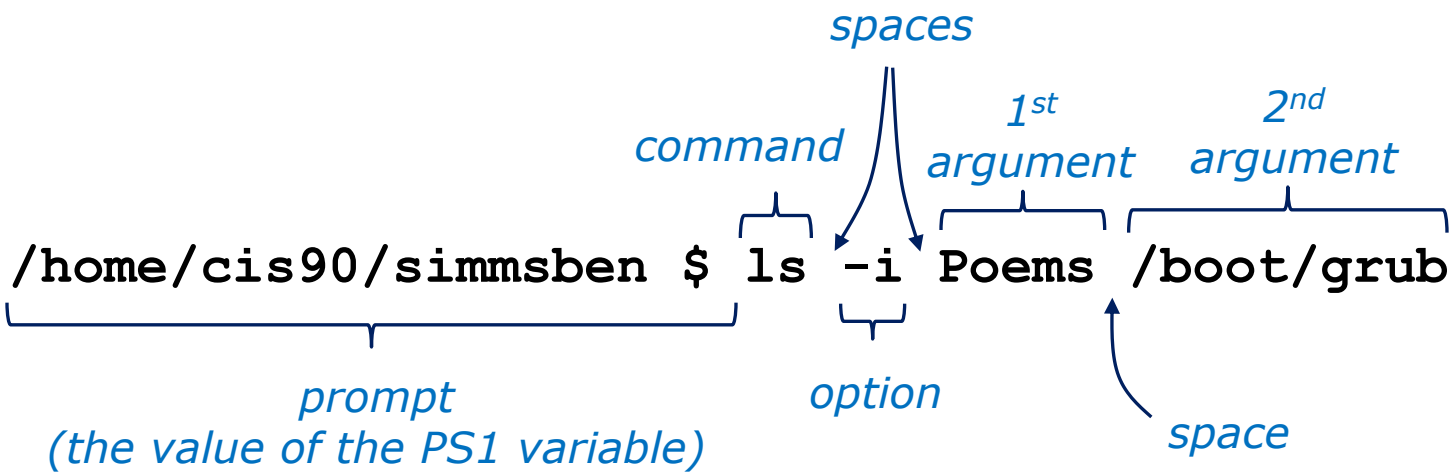
Command Line Syntax Review



Parsing the command line above yields:

One command: **ls**
Two options: **d** and **i**
One argument: **/home/cis90** (an absolute pathname to a directory)

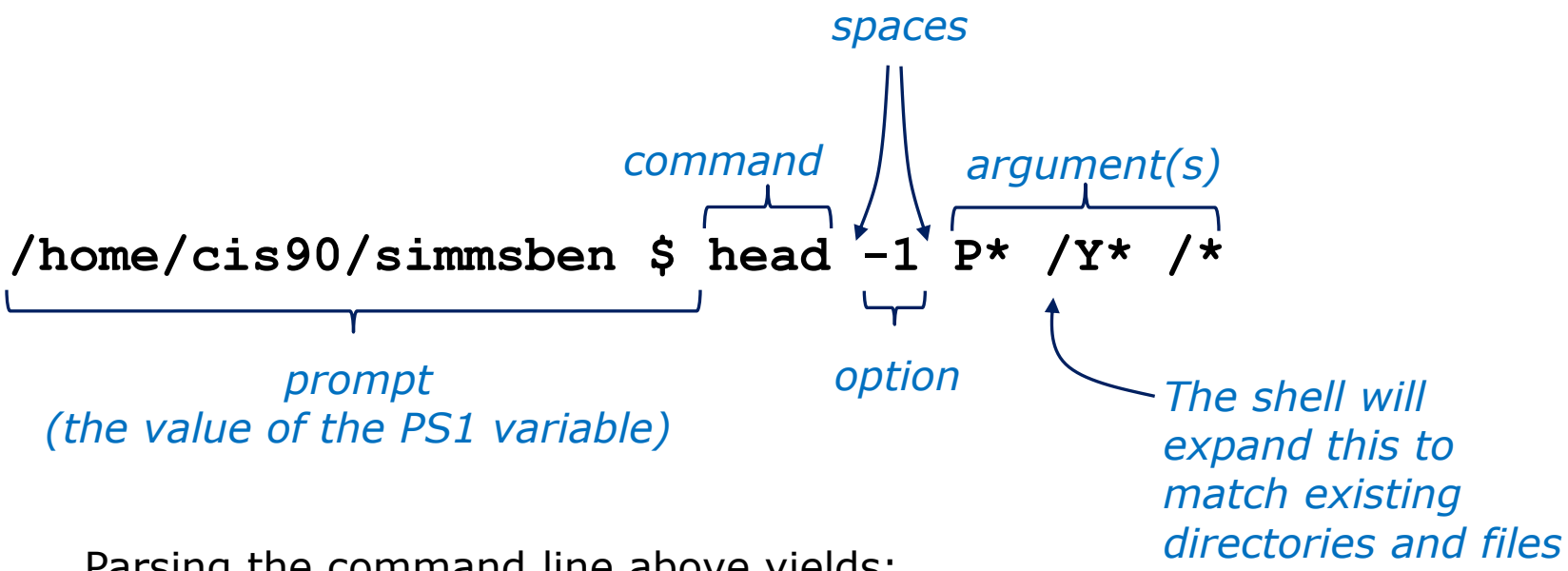
Command Line Syntax Review



Parsing the command line above yields:

One command: **ls**
One options: **i**
Two arguments:
Poems (a relative pathname to a directory)
/boot/group (an absolute pathname to a directory)

Command Line Syntax Review



Parsing the command line above yields:

One command: **head**
One option: **1**
Three arguments:

Poems/Yeats/mooncat (a relative pathname to a file)
Poems/Yeats/old (a relative pathname to a file)
Poems/Yeats/whitebirds (a relative pathname to a file)

Your turn now!

/home/cis90ol/simmsben $ ls -ls /usr/bin/ls*

1) What portion of the line above is the shell prompt?

/home/cis90ol/simmsben $

2) Parse the command the user typed and identify:

The name of the program/script to run: **ls**

options: There are 2 options: **l** and **s** (long and size in blocks)

arguments: there are 6 arguments:
/usr/bin/lsattr
/usr/bin/lsb_release
/usr/bin/lsdiff
/usr/bin/lshal
/usr/bin/lspgpot
/usr/bin/lss16toppm

Meta Characters (review)

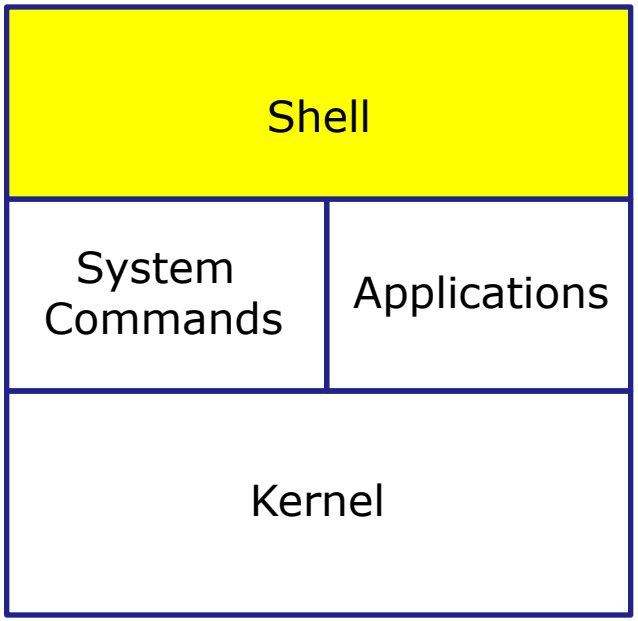
Metacharacters

Have special interpretation by the shell

Char	Description
\	Treat the following metacharacter as a plain character. Also called "escaping" the next character.
$	The following text is a shell (environment) variable and the value should be used.
<cr>	Carriage return marks the end of the command
;	Separates multiple commands on one line
'	used to enclose a string that the shell will not do further interpretation
"	Used to enclose a string that the shell will do further interpretation.
>	Redirects stdout (more in Lesson 8)
2>	Redirects stderr (more in Lesson 8)
*	Matches all non-hidden file names when used alone or zero or more characters when used as prefix, infix or postfix
?	Matches any single character of a file name
[]	Matches any single character contained within the brackets
#	Not an official metacharacter, but any text following the # is ignored by the shell

Life of the Shell

*The shell processes metacharacters during the **Parse** step*



1) **Prompt** for a command
2) **Parse** (interpret metacharacters, expand file names and dissect command line into options and arguments)
3) **Search** for program (along the path)
4) **Execute** program by loading into memory (becomes a process), hookup input and outputs, and pass along command line options and arguments.
5) **Nap** (wait till process is done)
6) **Repeat**

Metacharacters
#

 # has the ability to make everything that follows the # be ignored by the shell. Good for adding comments in scripts

```
/home/cis90/simmsben $ #OK lets escape the carriage return in next example
/home/cis90/simmsben $
```

Note there is no error message because everything after the # is ignored

Metacharacters
$

$ metacharacter has the ability to "show the value of"

```
/home/cis90/simmsben $ EYES=brown
/home/cis90/simmsben $ echo EYES
EYES
/home/cis90/simmsben $ echo $EYES
brown

/home/cis90/simmsben $ echo $LOGNAME
simmsben
/home/cis90/simmsben $
```

echo the string EYES

echo the value of the variable EYES

echo the value of the predefined environment variable LOGNAME

Metacharacters
" and '

Weak "double" quotes allow the shell to process $ metacharacters inside the quoted string

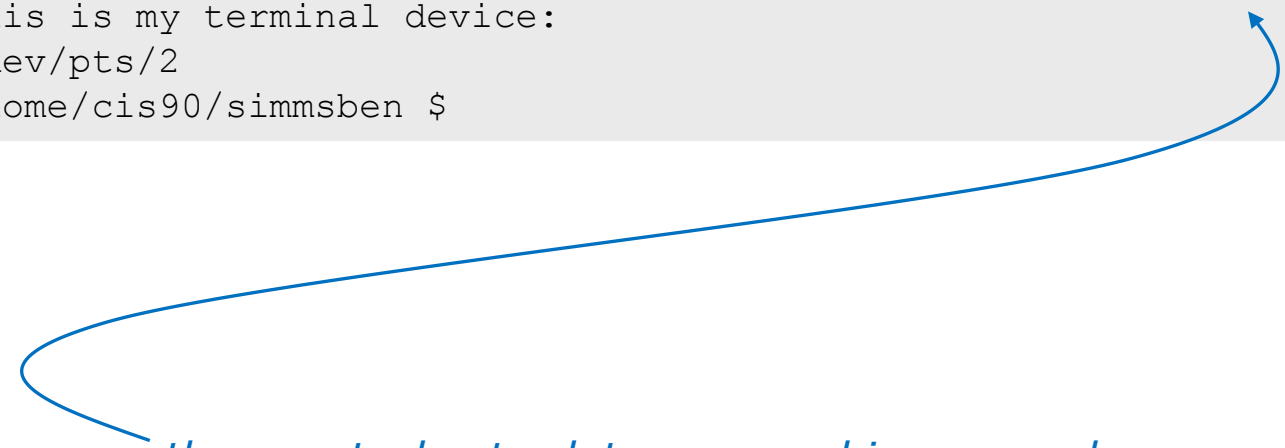
```
/home/cis90/simmsben $ echo "I am in $PWD"
I am in /home/cis90/simmsben


/home/cis90/simmsben $ echo 'I am in $PWD'
I am in $PWD
/home/cis90/simmsben $
```

Strong "single" quotes block the shell from processing $ metacharacters inside the quoted string

Metacharacters
;

```
/home/cis90/simmsben $ #Lets put two commands on one line
/home/cis90/simmsben $ echo "This is my terminal device:"; tty
This is my terminal device:
/dev/pts/2
/home/cis90/simmsben $
```



the ; metachacter lets you combine several commands on one line

Metacharacters

\

```
/home/cis90/simmsben $ #OK lets escape the carriage return in next example
/home/cis90/simmsben $ echo Lets start line 1 here \
> and finish it here
Lets start line 1 here and finish it here
/home/cis90/simmsben $
```

The \ is used to escape the next character typed. Use an escape to disable the special abilities of a metacharacter.

Escaping a carriage return (the Enter key) tells the shell to keeping inputting more characters from the next line for the current command being entered.

Metacharacters

\

Escaping the # means it is no longer treated as comment

```
/home/cis90/simmsben $ \#OK lets put a comment here
-bash: #OK: command not found
/home/cis90/simmsben $
/home/cis90/simmsben $
/home/cis90/simmsben $ echo $PS1
$PWD $
/home/cis90/simmsben $ echo \$PS1
$PS1
/home/cis90/simmsben $
```

and you get an error when the shell processes your comment

Escaping the $ means $ is no longer treated "the value of"

Class Exercise

- Use the # metacharacter
  #this is just a comment

- Use the $ and ; metacharacter
  echo $LOGNAME; echo LOGNAME

- Use the \ metacharacter
  \#This is not a comment

- Use strong an weak quotes metacharacters
  echo "My username is $LOGNAME"
  echo 'Use $LOGNAME to show your username'

Round 2

ahrmat90
blerav90
bodian90
bunsol90
cheken90
shidev90

13-15 2

lowmic90
macrya90
maxsco90
mcidar90
milhen90

16-18 2

cofcol90
colabd90
deltas90
doucor90
flamat90

19-21 2

milmic90
olscam90
pacnan90
phacha90
plajos90
veleli90

22-14 2

plajua90
porjon90
pummas90
rafdav90
reedie90

25-27 2

gueous90
helrog90
hovdav90
huljef90
jimmel90
varana90

28-30 2

Flashcards
L1=18
L2=22
L3=5
L4=26
L5=4
Total=75

Rules

- Chat window belongs to team that is up (no one else can use)
- "Final Answer" must be from someone on team that hasn't answered yet
- All team members can help each other and suggest answers

Environment Variables (review)

Shell (Environment) Variables

common environment variables

Shell Variable	Description
HOME	Users home directory (starts here after logging in and returns with a cd command (with no arguments)
LOGNAME	User's username for logging in with.
PATH	List of directories, separated by :'s, for the Shell to search for commands (which are program files) .
PS1	The prompt string.
PWD	Current working directory
SHELL	Name of the Shell program being used.
TERM	Type of terminal device , e.g. dumb, vt100, xterm, ansi, etc.

Shell (Environment) Variables

Show variable values

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

/home/cis90/simben $ echo $LOGNAME
simben90

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

/home/cis90/simben $ echo $PWD
/home/cis90/simben

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

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

Use echo to show the values of variables

Shell (Environment) Variables
PATH

```
/home/cis90/simben $ echo $PATH
/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/cis90/si
mben/../bin:/home/cis90/simben/bin:.
```

These are the directories in Benji's PATH in the order they will be searched:

1st: /usr/kerberos/bin
2nd: /usr/local/bin
3rd: /bin
4th: /usr/bin
5th: /home/cis90/simben/../bin
6th: /home/cis90/simben/bin
7th: .

The PATH variable is used by the shell to locate commands

Shell (Environment) Variables

Set variable values

Use an "=" with no spaces to set values of variables

```
/home/cis90/simben $  # Change the prompt variable
/home/cis90/simben $  PS1='[\u@\h \W]\$ '
[simben90@opus ~]$ echo $PS1
[\u@\h \W]\$
[simben90@opus ~]$


[simben90@opus ~]$ # Change it back again
[simben90@opus ~]$ PS1='$PWD $ '
/home/cis90/simben $ echo $PS1
$PWD $
/home/cis90/simben $
```

Shell Variables

Set variable values

If the variable has never been use before then it is created

```
/home/cis90/simben $ myfavoritedog="Benji Simms"
/home/cis90/simben $ echo $myfavoritedog
Benji Simms
/home/cis90/simben $
```

Shell (Environment) Variables

env command – show all environment variables

```
/home/cis90/simmsben/Poems $ env
HOSTNAME=opus.cabrillo.edu
SHELL=/bin/bash
TERM=xterm
HISTSIZE=1000
USER=simmsben
LS_COLORS=no=00:fi=00:di=00;34:ln=00;36:pi=40;33:so=00;35:bd=40;33;01:cd=40;33;01:or=01;05;37;41:mi
=01;05;37;41:ex=00;32:*.cmd=00;32:*.exe=00;32:*.com=00;32:*.btm=00;32:*.bat=00;32:*.sh=00;32:*.csh=
00;32:*.tar=00;31:*.tgz=00;31:*.arj=00;31:*.taz=00;31:*.lzh=00;31:*.zip=00;31:*.z=00;31:*.Z=00;31:*
.gz=00;31:*.bz2=00;31:*.bz=00;31:*.tz=00;31:*.rpm=00;31:*.cpio=00;31:*.jpg=00;35:*.gif=00;35:*.bmp=
00;35:*.xbm=00;35:*.xpm=00;35:*.png=00;35:*.tif=00;35:
USERNAME=
MAIL=/var/spool/mail/simmsben
PATH=/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/cis90/simmsben/../bin:/home/cis90/simmsbe
n/bin:.
INPUTRC=/etc/inputrc
PWD=/home/cis90/simmsben/Poems
LANG=en_US.UTF-8
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
SHLVL=1
HOME=/home/cis90/simmsben
BASH_ENV=/home/cis90/simmsben/.bashrc
LOGNAME=simmsben
CVS_RSH=ssh
LESSOPEN=|/usr/bin/lesspipe.sh %s
G_BROKEN_FILENAMES=1
_=/bin/env
OLDPWD=/home/cis90/simmsben
/home/cis90/simmsben/Poems $
```

*Use the **env** command to show all environment variables (a subset of the shell variables)*

Shell Variables

set command – show all shell variables

```
/home/cis90/simmsben/Poems $ set
BASH=/bin/bash
BASH_ARGC=()
BASH_ARGV=()
BASH_ENV=/home/cis90/simmsben/.bashrc
BASH_LINENO=()
BASH_SOURCE=()
BASH_VERSINFO=([0]="3" [1]="2" [2]="25" [3]="1"
[4]="release" [5]="i686-redhat-linux-gnu")
BASH_VERSION='3.2.25(1)-release'
COLORS=/etc/DIR_COLORS.xterm
COLUMNS=80
CVS_RSH=ssh
DIRSTACK=()
EUID=1160
GROUPS=()
G_BROKEN_FILENAMES=1
HISTFILE=/home/cis90/simmsben/.bash_history
HISTFILESIZE=1000
HISTSIZE=1000
HOME=/home/cis90/simmsben
HOSTNAME=opus.cabrillo.edu
HOSTTYPE=i686
IFS=$' \t\n'
IGNOREEOF=10
INPUTRC=/etc/inputrc
LANG=en_US.UTF-8
LESSOPEN='|/usr/bin/lesspipe.sh %s'
LINES=24
LOGNAME=simmsben
```

*Use the **set** command to show all shell variables (which includes the environment variables)*

```
LS_COLORS='no=00:fi=00:di=00;34:ln=00;36:pi=40;33:so=00;35
:bd=40;33;01:cd=40;33;01:or=01;05;37;41:mi=01;05;37;41:ex=
00;32:*.cmd=00;32:*.exe=00;32:*.com=00;32:*.btm=00;32:*.ba
t=00;32:*.sh=00;32:*.csh=00;32:*.tar=00;31:*.tgz=00;31:*.a
rj=00;31:*.taz=00;31:*.lzh=00;31:*.zip=00;31:*.z=00;31:*.Z
=00;31:*.gz=00;31:*.bz2=00;31:*.bz=00;31:*.tz=00;31:*.rpm=
00;31:*.cpio=00;31:*.jpg=00;35:*.gif=00;35:*.bmp=00;35:*.x
bm=00;35:*.xpm=00;35:*.png=00;35:*.tif=00;35:'
MACHTYPE=i686-redhat-linux-gnu
MAIL=/var/spool/mail/simmsben
MAILCHECK=60
OLDPWD=/home/cis90/simmsben
OPTERR=1
OPTIND=1
OSTYPE=linux-gnu
PATH=/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/
cis90/simmsben/../bin:/home/cis90/simmsben/bin:.
PIPESTATUS=([0]="0")
PPID=26514
PROMPT_COMMAND='echo -ne
"\033]0;${USER}@${HOSTNAME%%.*}:${PWD/#$HOME/~}"; echo -ne
"\007"'
PS1='$PWD $'
PS2='> '
PS4='+ '
PWD=/home/cis90/simmsben/Poems
SHELL=/bin/bash
SHELLOPTS=braceexpand:emacs:hashall:histexpand:ignoreeof:i
nteractive-comments:monitor
SHLVL=1
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
TERM=xterm
UID=1160
USER=simmsben
USERNAME=
_=env
consoletype=pty
```

Class Exercise

- Change your prompt with:
  PS1='$LOGNAME, command please: '

- Change your prompt with:
  PS1='[\u@\h \W]\$ '

- Change your prompt with:
  PS1="$PWD $ "
  Now change directories using **cd**, what happenned?

- Restore original prompt with:
  PS1='$PWD $ '

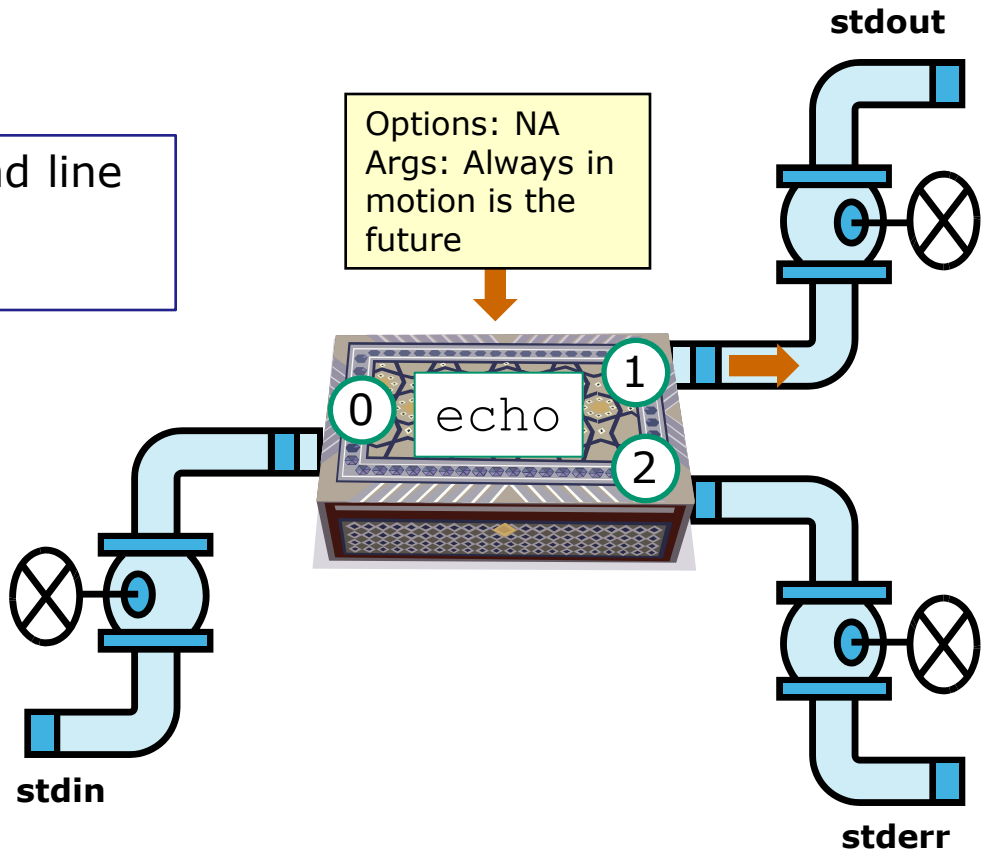
Program to Process (continuing)

Example program to process: echo command

```
[rsimms@opus ~]$ echo Always in motion is the future
Always in motion is the future
[rsimms@opus ~]$
```

Inputs: Command line

Outputs: stdout



/dev/pts/1

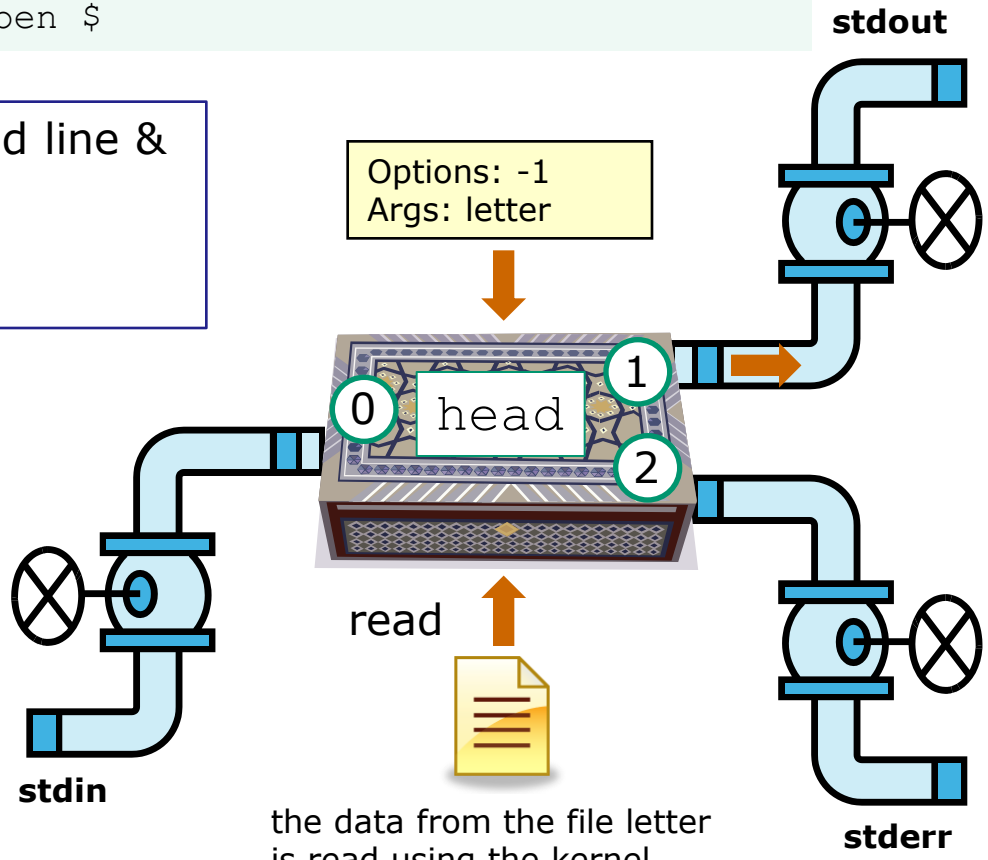
```
Always in
motion is
the future
```

Example program to process: head command

```
/home/cis90/simmsben $ head -1 letter
Hello Mother!  Hello Father!
/home/cis90/simmsben $
```

Inputs: Command line & Operating System

Outputs: stdout



/dev/pts/1

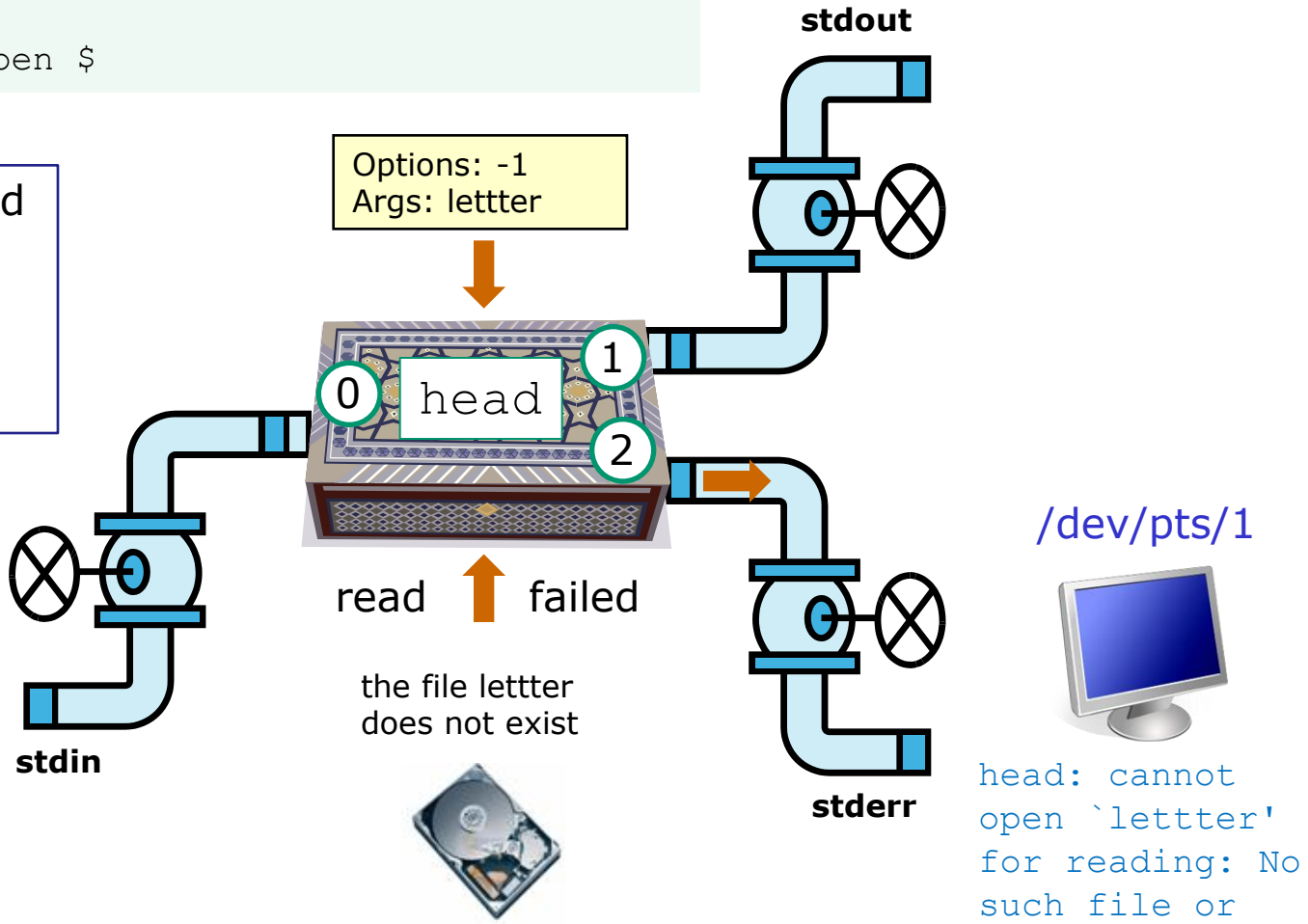
```
Hello Mother!
Hello Father!
```

Example program to process: head command

```
/home/cis90/simmsben $ head -1 lettter
head: cannot open `lettter' for reading: No such
file or directory
/home/cis90/simmsben $
```

Inputs: Command line & Operating System

Outputs: stderr



Example program to process: ls command

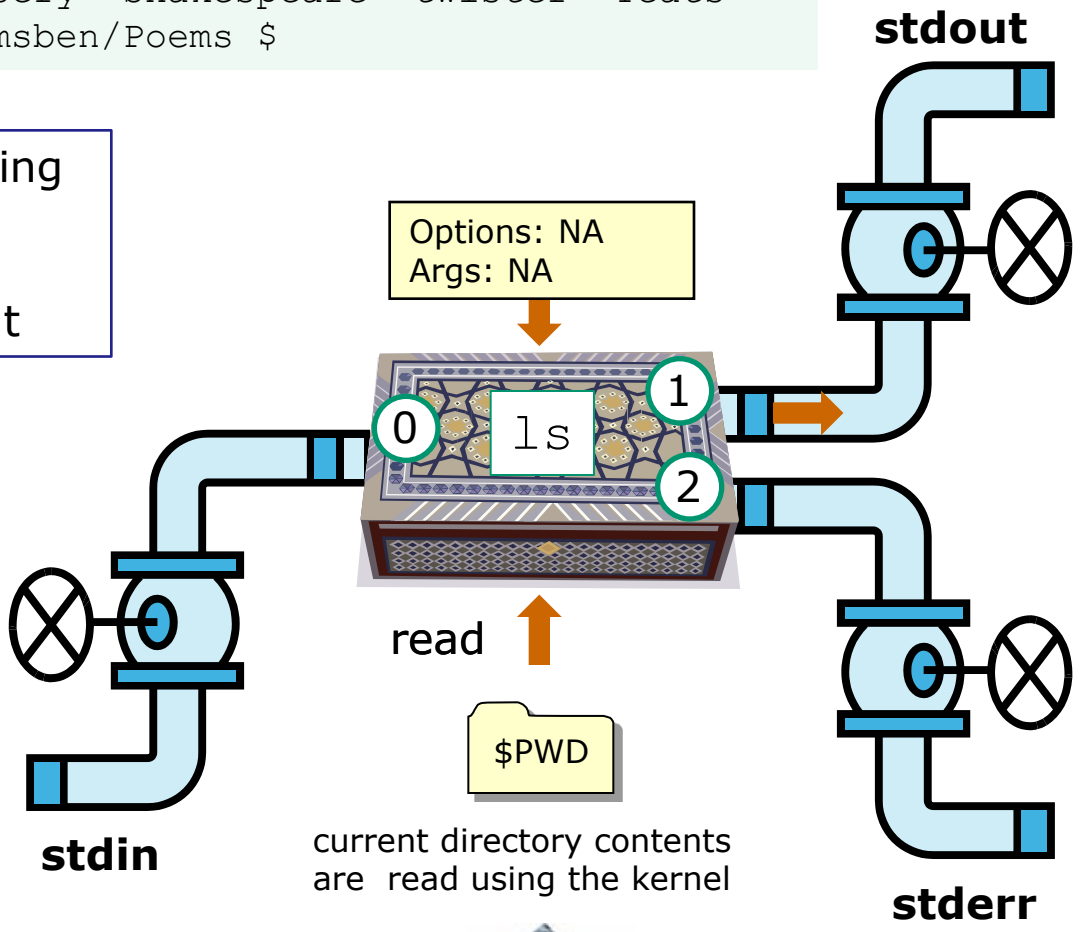
```
/home/cis90/simmsben/Poems $ ls
ant  Blake  nursery  Shakespeare  twister  Yeats
/home/cis90/simmsben/Poems $
```

/dev/pts/1

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

Inputs: Operating System

Outputs: stdout

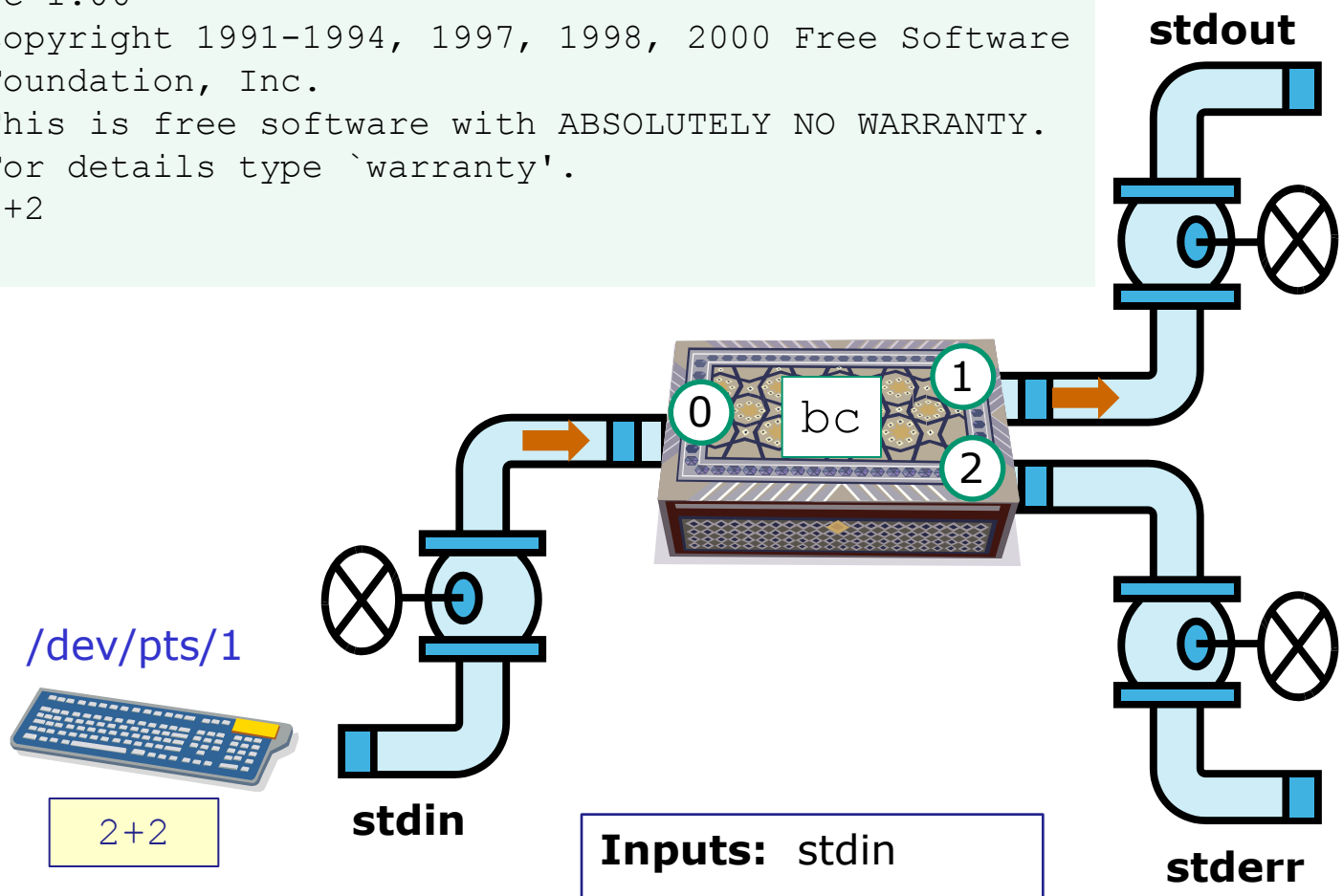


Example program to process: bc command

```
/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
```

/dev/pts/1

```
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'.
4
```

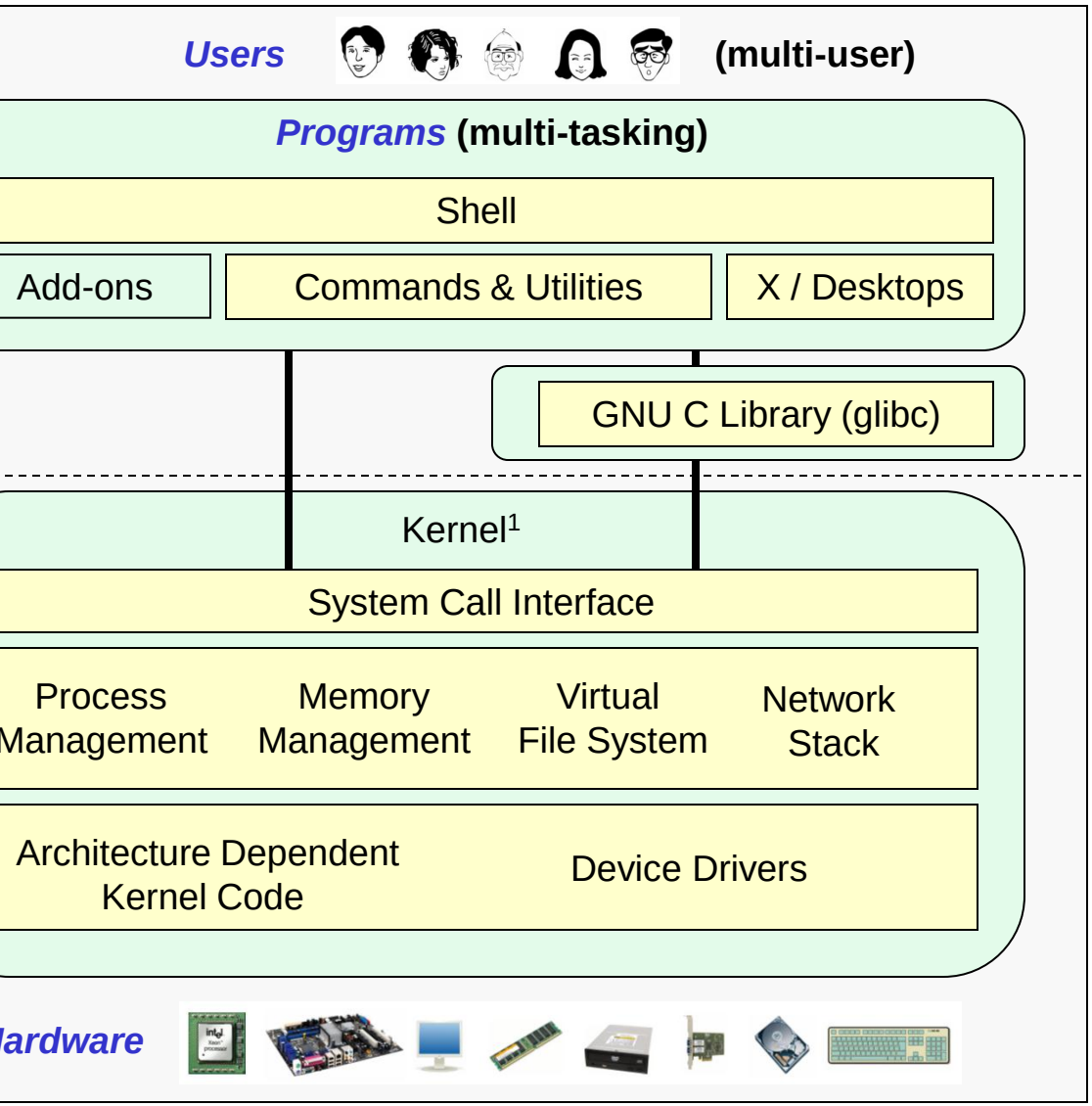


Inputs: stdin

Outputs: stdout

Architecture (review)

GNU/Linux Operating System Architecture



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

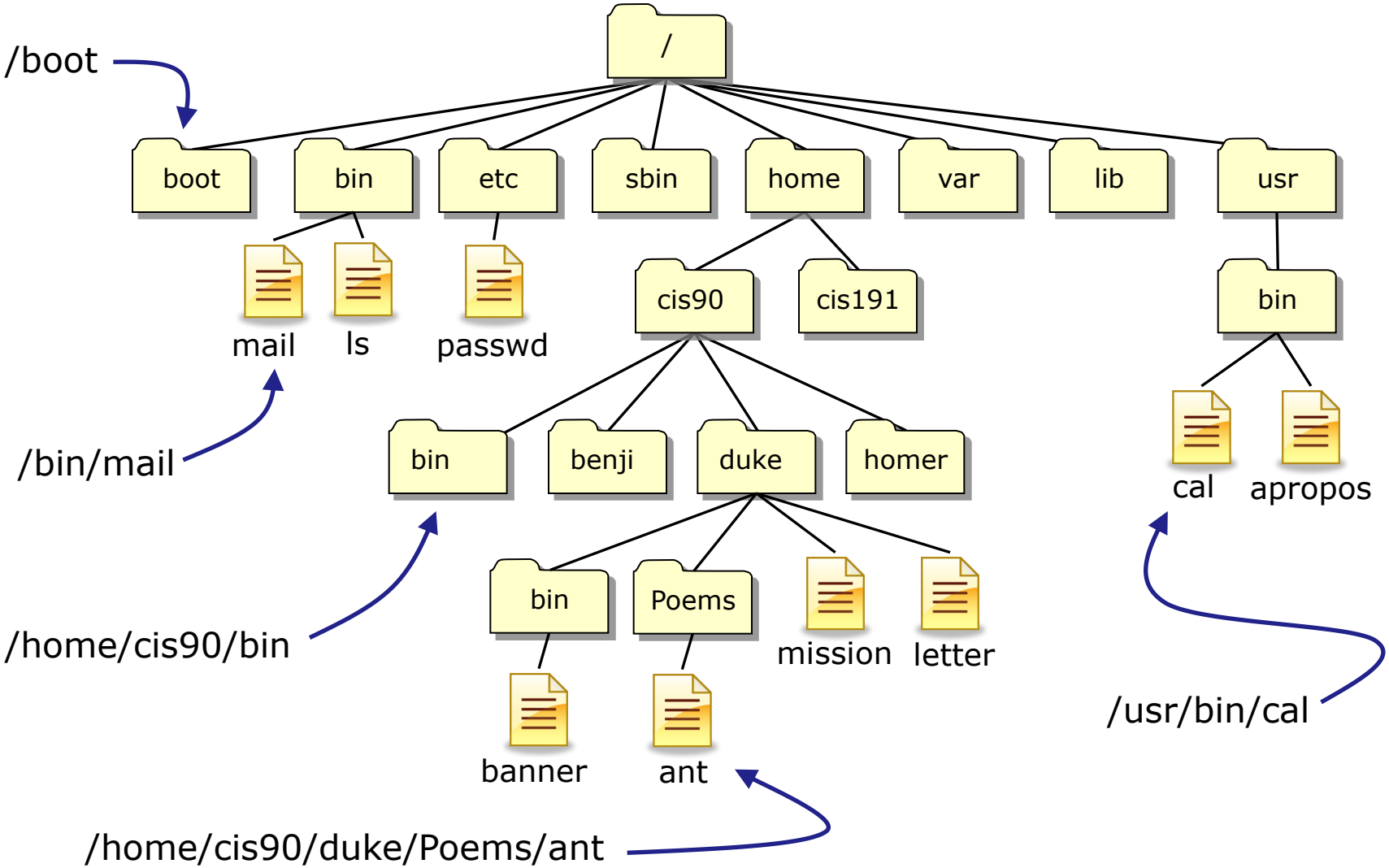
Linus Torvalds, as a student, initially conceived and assembled the Linux kernel in 1991. The kernel was later re-licensed under the GNU General Public License in 1992.

[1]See "Anatomy of the Linux kernel" by M. Tim Jones at http://www-128.ibm.com/developerworks/linux/library/l-linux-kernel/

File System (review)

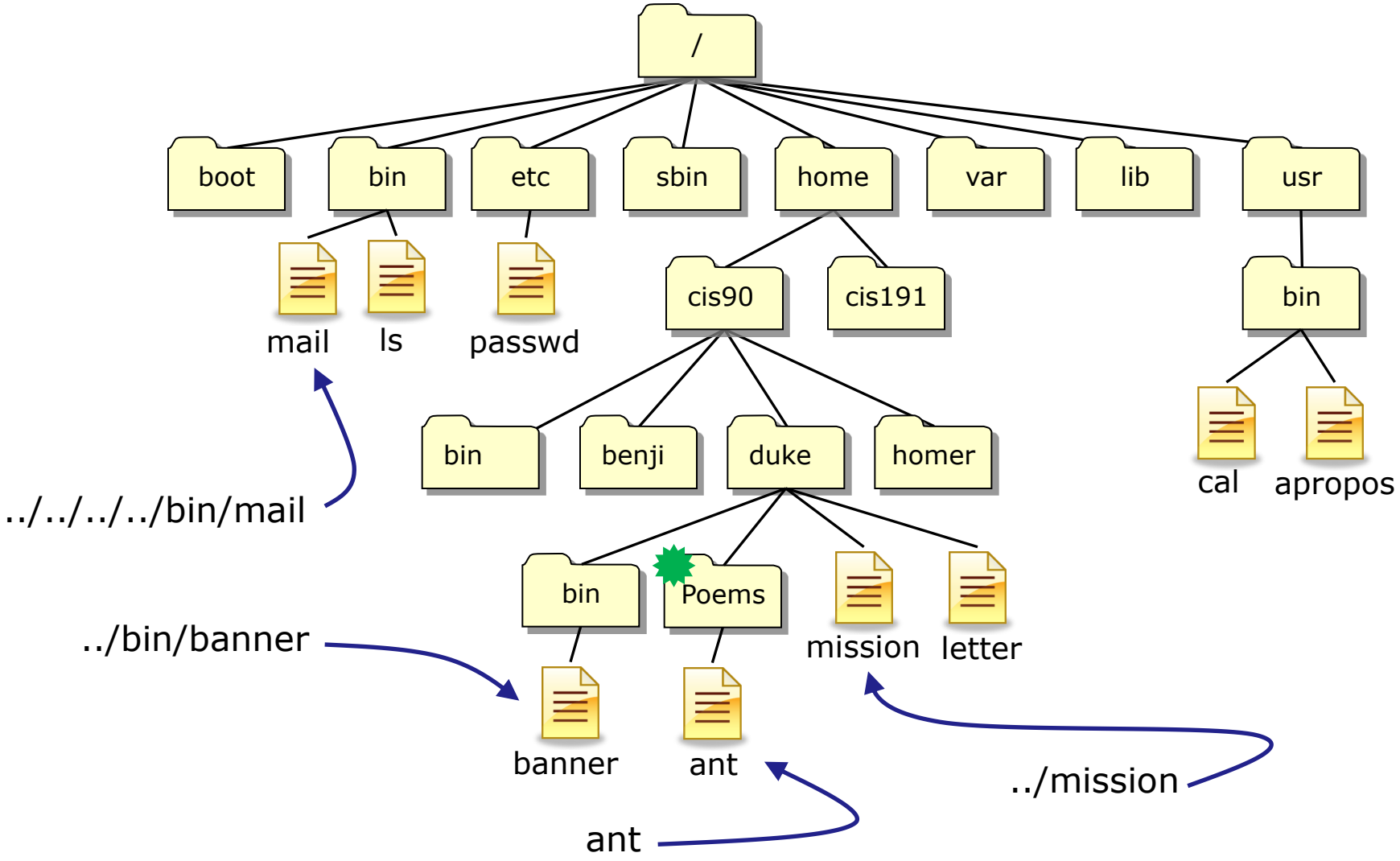
Absolute Pathnames

Fully specified names starting with /



Relative Pathnames

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

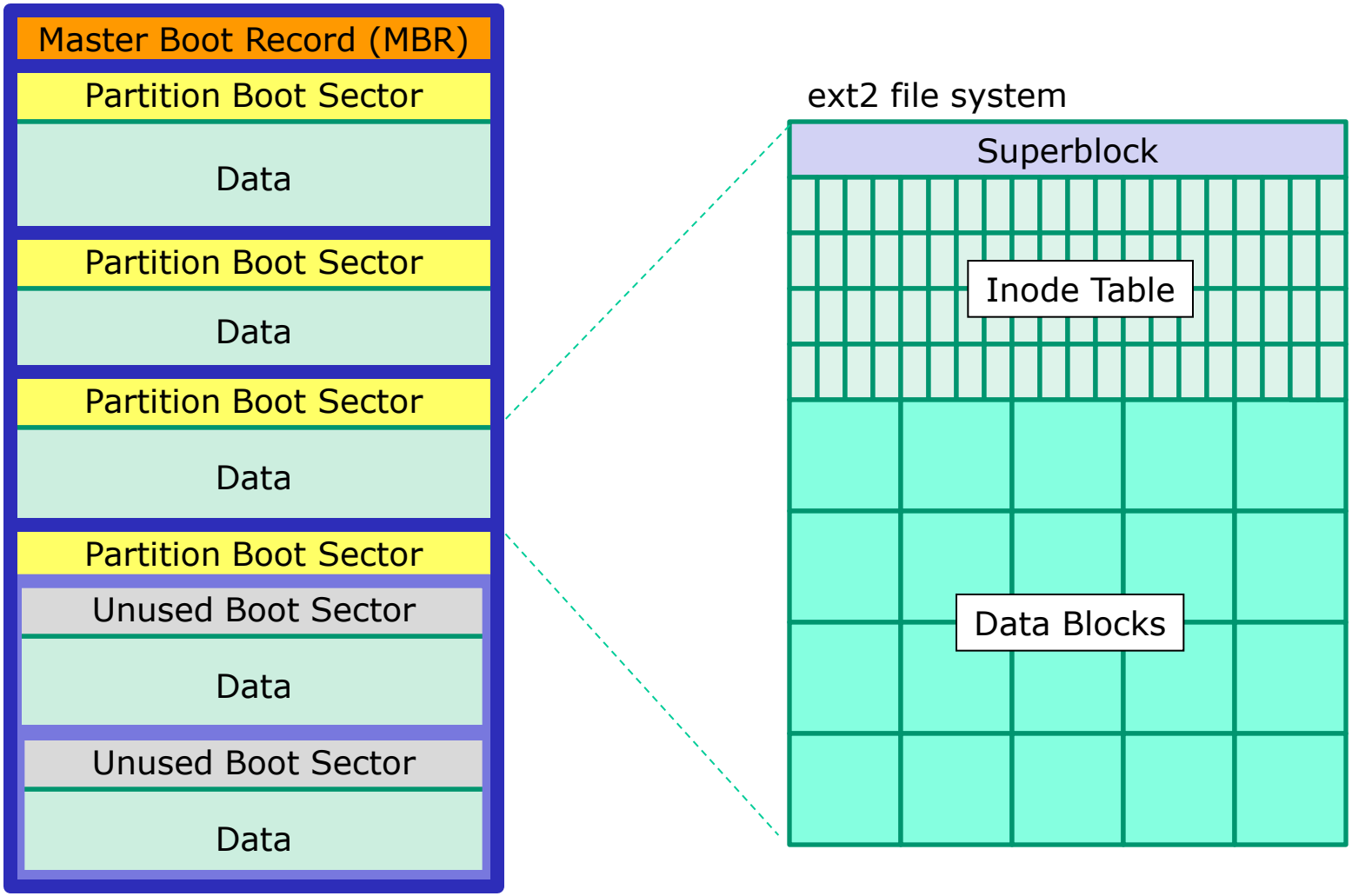


Top Level Directories

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

File Systems

Linux

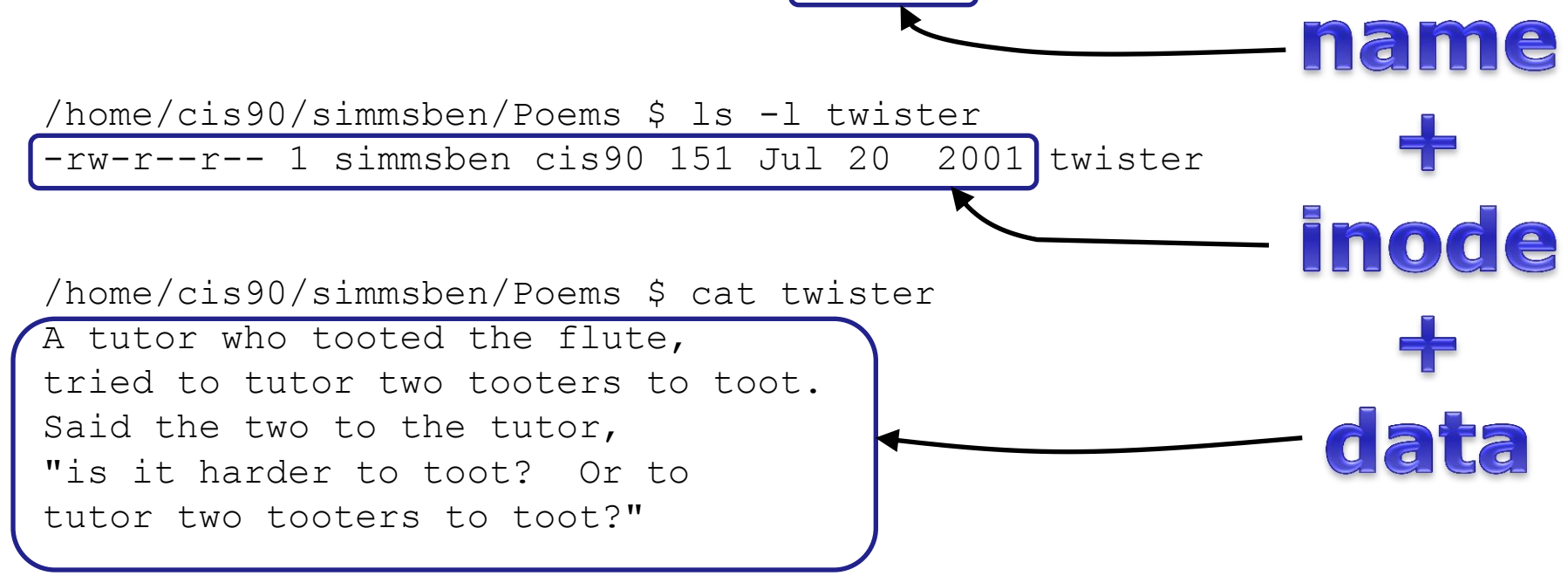


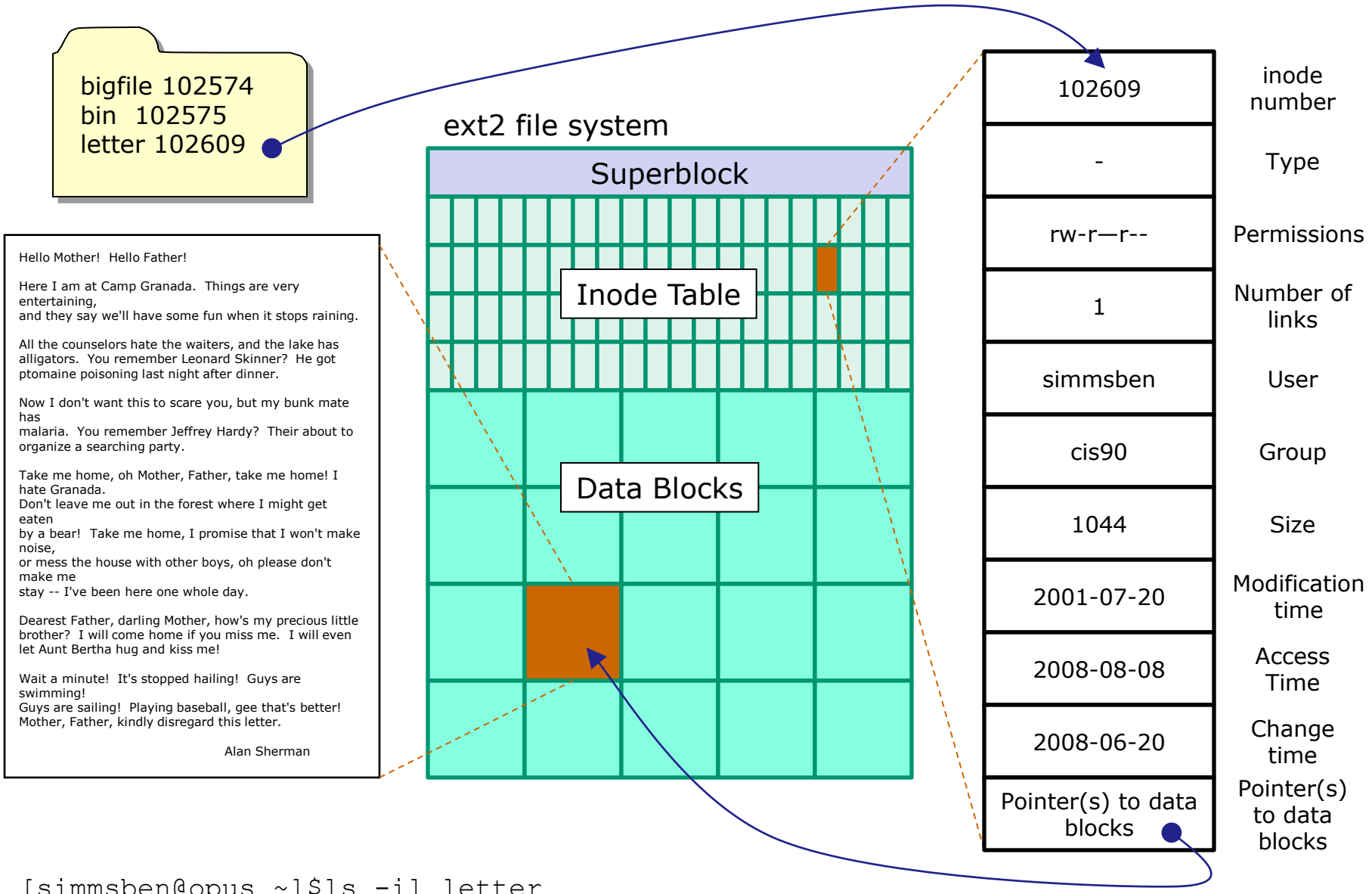
UNIX Files
The three elements of a file

```
/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?"
```






```
[simmsben@opus ~]$ls -il letter
102609 -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) *Use the **file** command to further classify files*	touch
l	symbolic link	ln -s
c	character device files	mknod
b	block device files	mknod

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

Column 1 of a long listing shows file types

```
simmsben@opus:~
/home/cis90/simmsben $ls -la
total 320
drwx------ 9 simmsben cis90  4096 Aug  8 11:51 .
drwxr-x--- 9 rsimms   cis90  4096 Jun 30 14:57 ..
-rw------- 1 simmsben cis90 11409 Aug  7 19:20 .bash_history
-rw------- 1 simmsben cis90    24 Jul 20  2001 .bash_logout
-rw------- 1 simmsben cis90   354 Sep 17  2003 .bash_profile
-rw------- 1 simmsben cis90   146 Jan 18  2004 .bashrc
-rw-rw-r-- 1 simmsben cis90    56 Jul  8 17:22 bcommands
-rw-r--r-- 2 simmsben cis90 10576 Jul 20  2001 bigfile
drwxr-xr-x 2 simmsben cis90  4096 Sep 11  2005 bin
-rw-rw-r-- 1 simmsben cis90  1044 Aug  8 11:52 deleteme
-rw-r--r-- 1 simmsben cis90   515 Jun 30 14:57 .emacs
-rw-r--r-- 1 simmsben cis90     0 Jul 20  2001 empty
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
-rw------- 1 simmsben cis90    35 Aug  8 13:58 .lesshst
-rw-r--r-- 1 simmsben cis90  1044 Jul 20  2001 letter
-rw------- 1 simmsben cis90  5799 Jul 24 21:08 mbox
drwxr-xr-x 2 simmsben cis90  4096 Sep 11  2005 Miscellaneous
-rw-r--r-- 1 simmsben cis90   759 Jun  6  2002 mission
drwxr-xr-x 4 simmsben cis90  4096 Jun 30 14:57 .mozilla
-rw-r--r-- 1 simmsben cis90    40 Jul 20  2001 .plan
drwxr-xr-x 5 simmsben cis90  4096 Jul  9 14:24 Poems
-rw-r--r-- 1 simmsben cis90  1074 Aug 26  2003 proposal1
-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  5467 Jul  6 13:41 results-e1
-rw-r--r-- 1 simmsben cis90  1286 Jul  6 12:20 results-e1a
-rw-rw-r-- 1 simmsben cis90   688 Jul 24 15:35 salsa
-rw-r--r-- 1 simmsben cis90  1580 Nov 16  2004 small_town
-rw-r--r-- 1 simmsben cis90   485 Aug 26  2003 spellk
-rw-r--r-- 1 simmsben cis90   250 Jul 20  2001 text.err
-rw-r--r-- 1 simmsben cis90   231 Jul 20  2001 text.fxd
-rwxr-xr-x 1 simmsben cis90   509 Jun  6  2002 timecal
-rw------- 1 simmsben cis90   661 Jul 24 13:59 .viminfo
-rw-r--r-- 1 simmsben cis90   352 Jul 20  2001 what_am_i
-rw------- 1 simmsben cis90   126 Aug  7 14:23 .Xauthority
-rw-r--r-- 1 simmsben cis90   658 Jun 30 14:57 .zshrc
/home/cis90/simmsben $
```

All directories in the UNIX file tree contain these two hidden . and .. directories (d in column 1)

A regular file (- in column 1) Its hidden because it starts with a .

A directory (d in column 1) Color is blue because it's a directory

Regular file (- in column 1)

regular file (- in column 1) Color is green because with execute bits are set

Symbolic links

A symbolic link file (l in column 1)

```
/home/cis90/simben $ ls -l accounts /etc/passwd
lrwxrwxrwx 1 simben90 cis90   11 Mar  7 08:52 accounts -> /etc/passwd
-rw-r--r-- 1 root     root  7183 Mar  6 08:17 /etc/passwd
/home/cis90/simben $
```

```
/home/cis90/simben $ head -5 /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin

/home/cis90/simben $ head -5 accounts
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

The accounts file in Benji's directory is a symbolic link to the /etc/passwd file.

These "shortcuts" can be used for convenience

```
/home/cis90/simben $ ls -i accounts /etc/passwd
  99983 accounts  1280173 /etc/passwd
/home/cis90/simben $
```

Note they have different inodes

Round 3

debian	redhat	SUSE	CentOS	ubuntu	
ahrmat90	lowmic90	cofcol90	milmic90	plajua90	gueous90
blerav90	macrya90	colabd90	olscam90	porjon90	helrog90
bodian90	maxsco90	deltas90	pacnan90	pummas90	hovdav90
bunsol90	mcidar90	doucor90	phacha90	rafdav90	huljef90
cheken90	milhen90	flamat90	plajos90	reedie90	jimmel90
shidev90			veleli90		varana90
31-35 1	*36-40 1*	*41-45 1*	*46-50 1*	*51-55 1*	*56-60 1*

Flashcards
L1=18
L2=22
L3=5
L4=26
L5=4
Total=75

Rules

- Chat window belongs to team that is up (no one else can use)
- "Final Answer" must be from someone on team that hasn't answered yet
- All team members can help each other and suggest answers

Flashcards

Flash Cards

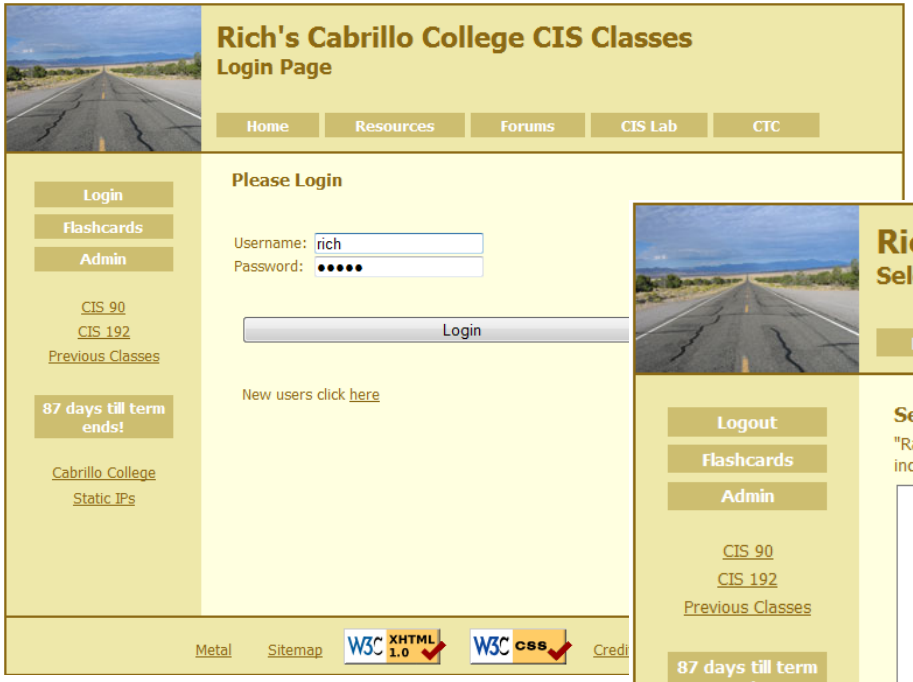
Click on Flashcards in left panel

Register if this is the first time using Flashcards

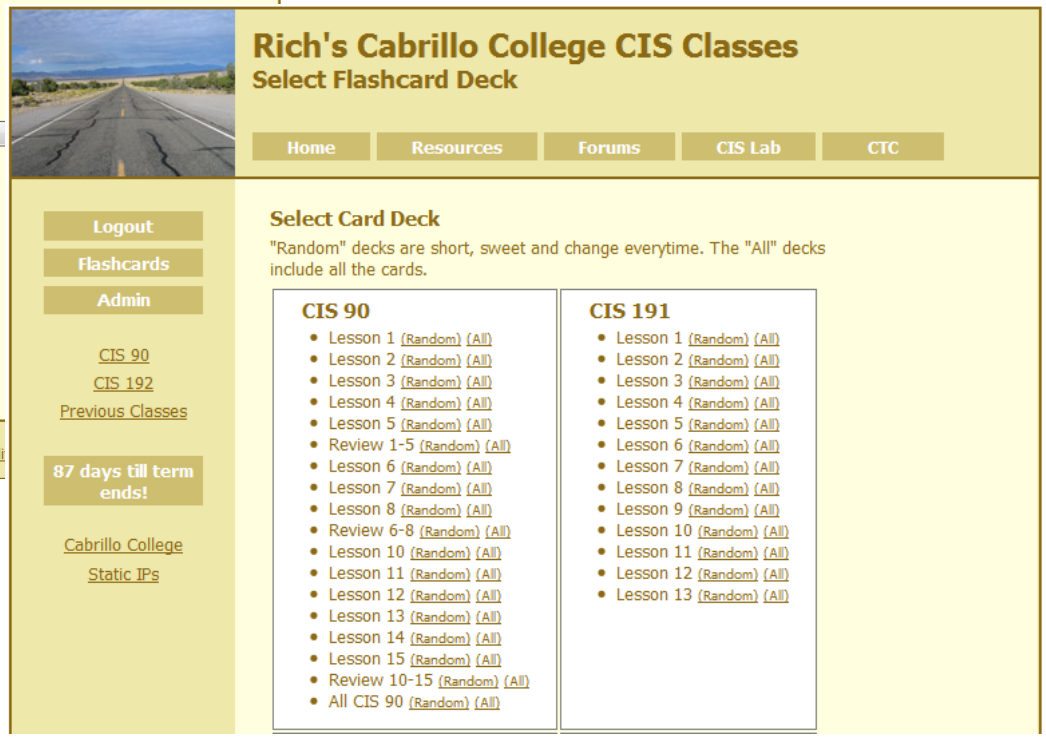
Register and choose a username and password of your choice

Logging in and using Flashcards

Login with your username and password



Select deck of cards



Class Exercise
Flashcards

- Browse to simms-teach.com
- Register with a username and password of your choice
- Verify you can login and use the flash cards.

Using VLab (review)

Lab Resources

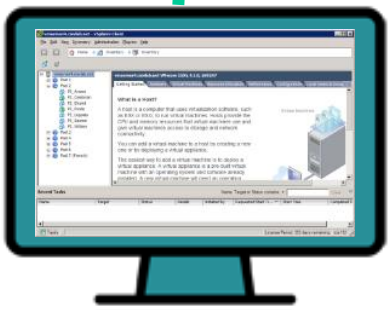
Remote Access to **CIS VLab**

Room 1403 on Aptos Campus

Internet

cislab (Win 2008)

vmserver4 (VMware ESXi)



You can access the course VMs from home or the CTC using RDP (Remote Desktop Protocol)

Home

CTC

MrEko

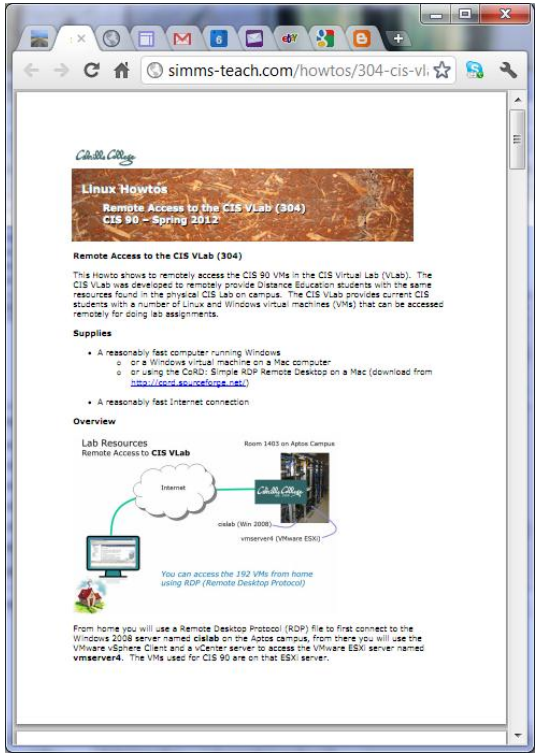
NotOpus

Fang

Kate

VLab Howto

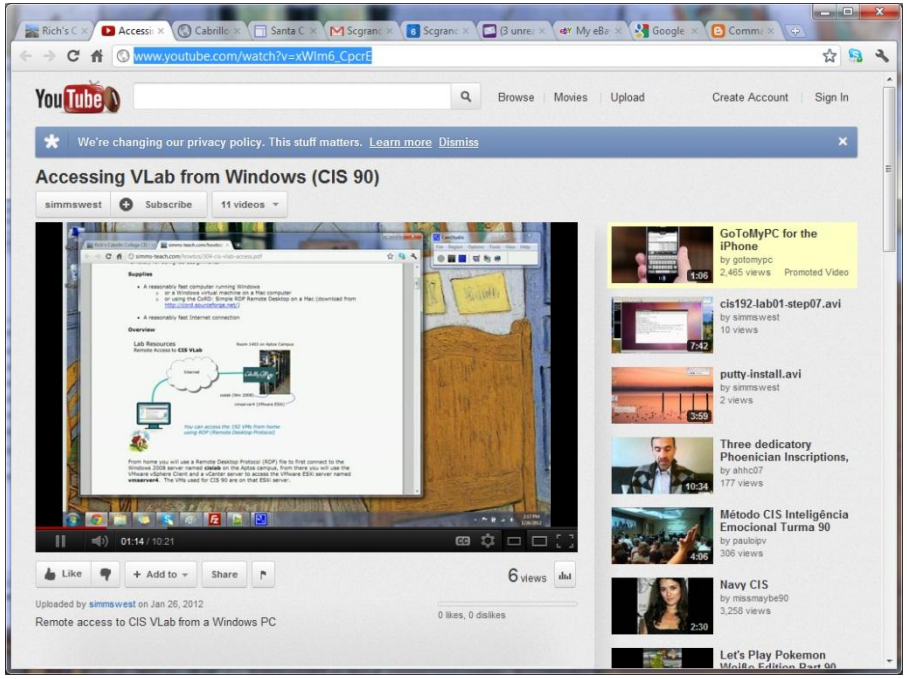
Howto #304: Accessing VLab
http://simms-teach.com/howtos/304-cis-vlab-access.pdf



Documents how to access CIS VLab

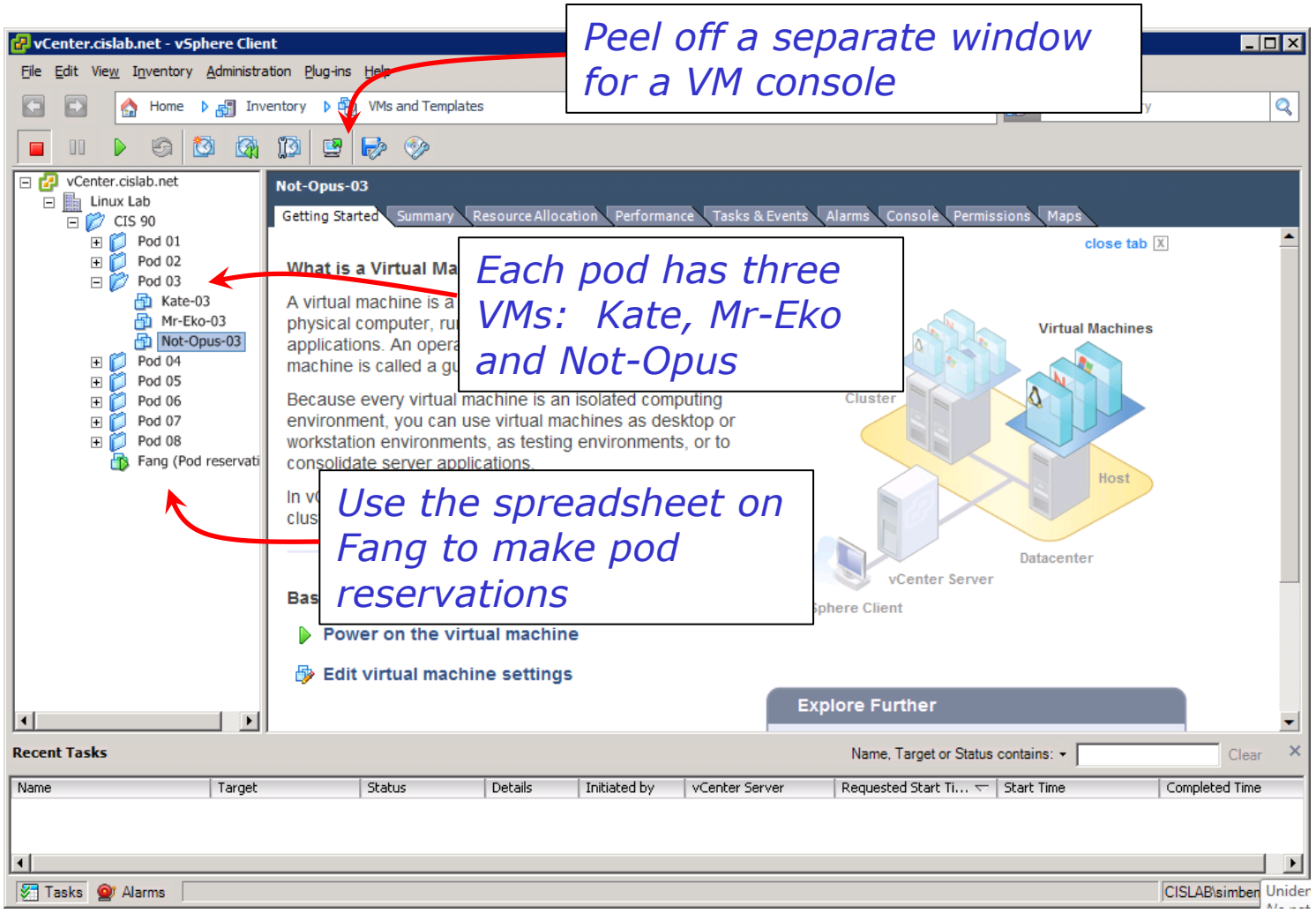
VLab Video

Accessing VLab from Windows (video)
http://www.youtube.com/watch?v=xWlm6_CpcrE



Shows how to access CIS VLab

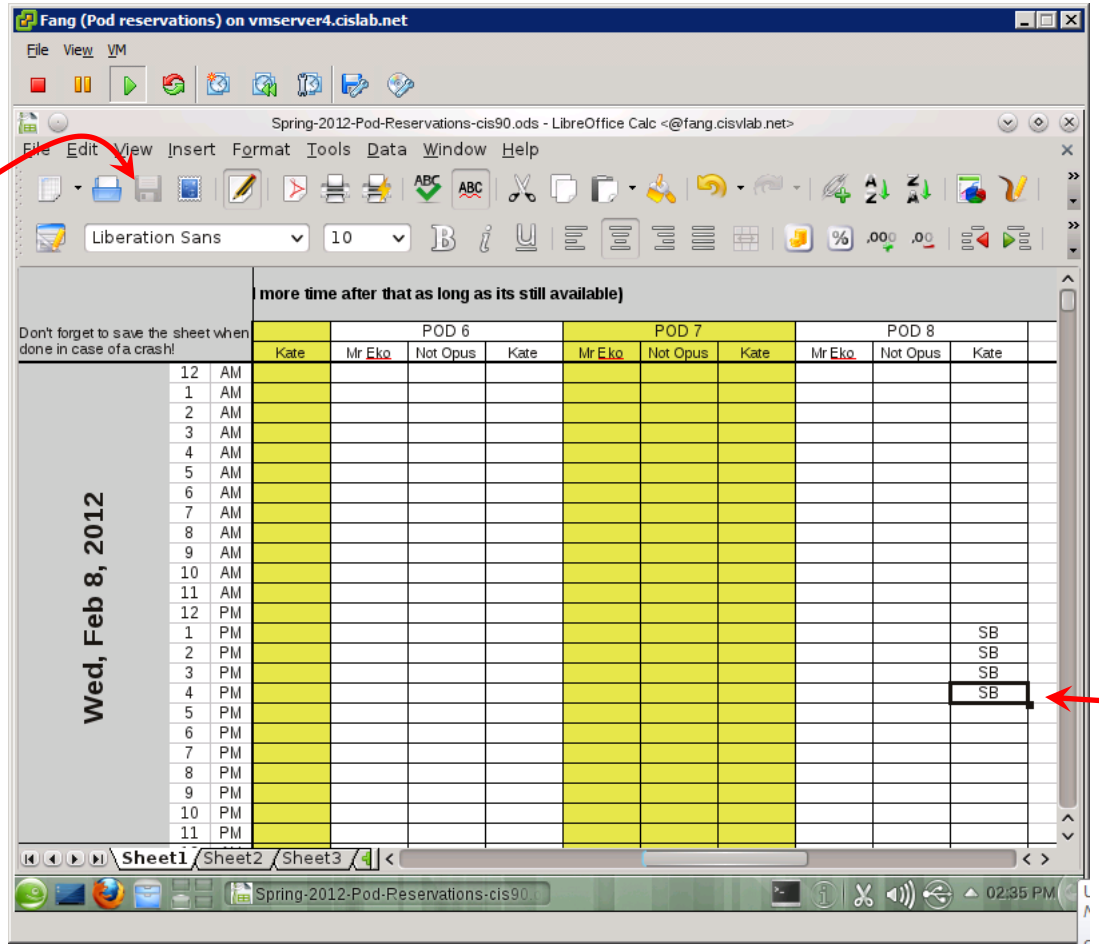
CIS VLab



VMware vSphere Client

The Fang VM (openSUSE)

It's a good idea to save the spreadsheet after you make your changes

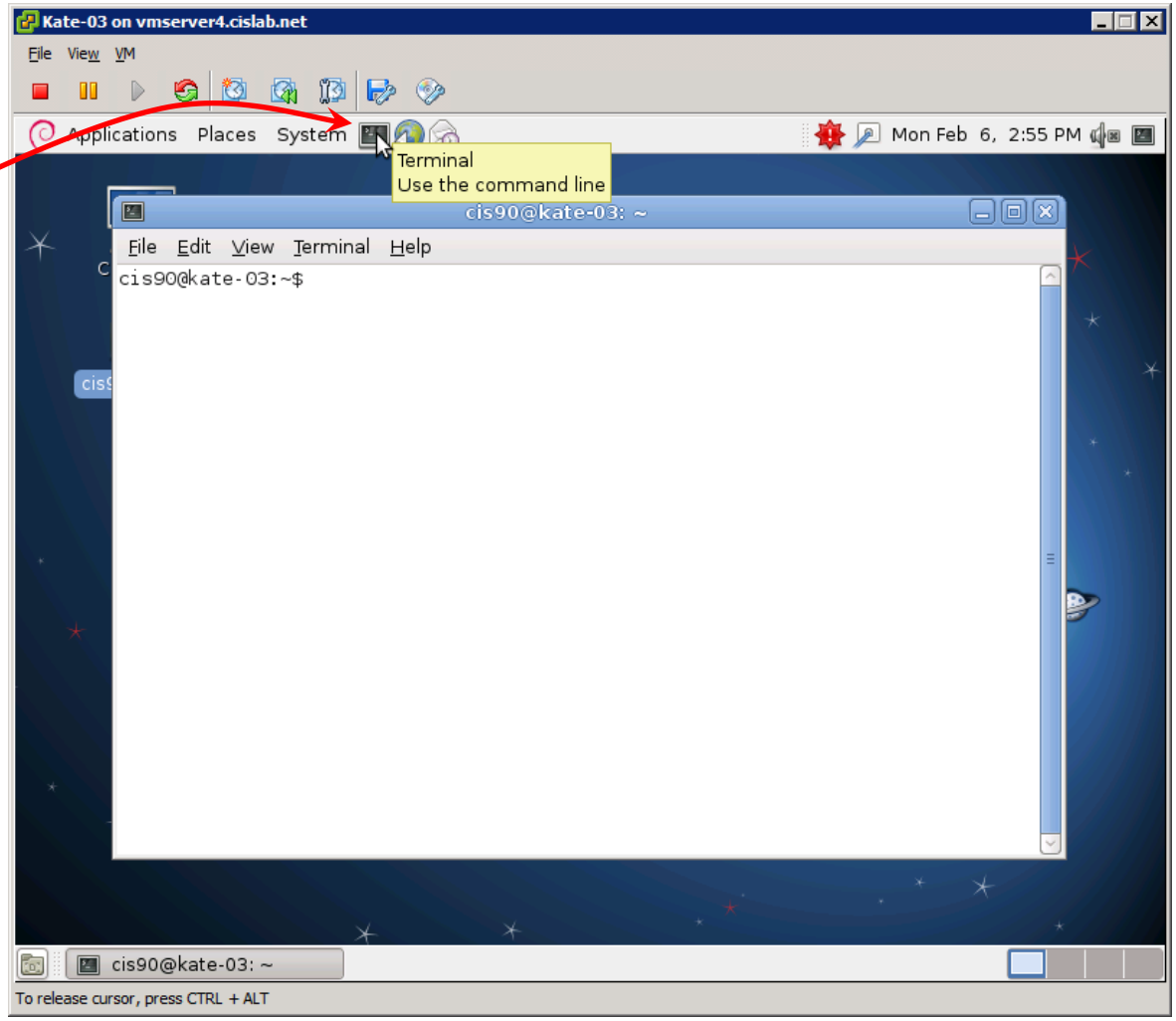


Type your initials into the spreadsheet cells to indicate the date, time and VMs you wish to reserve.

Don't shut down this VM or close the spreadsheet.

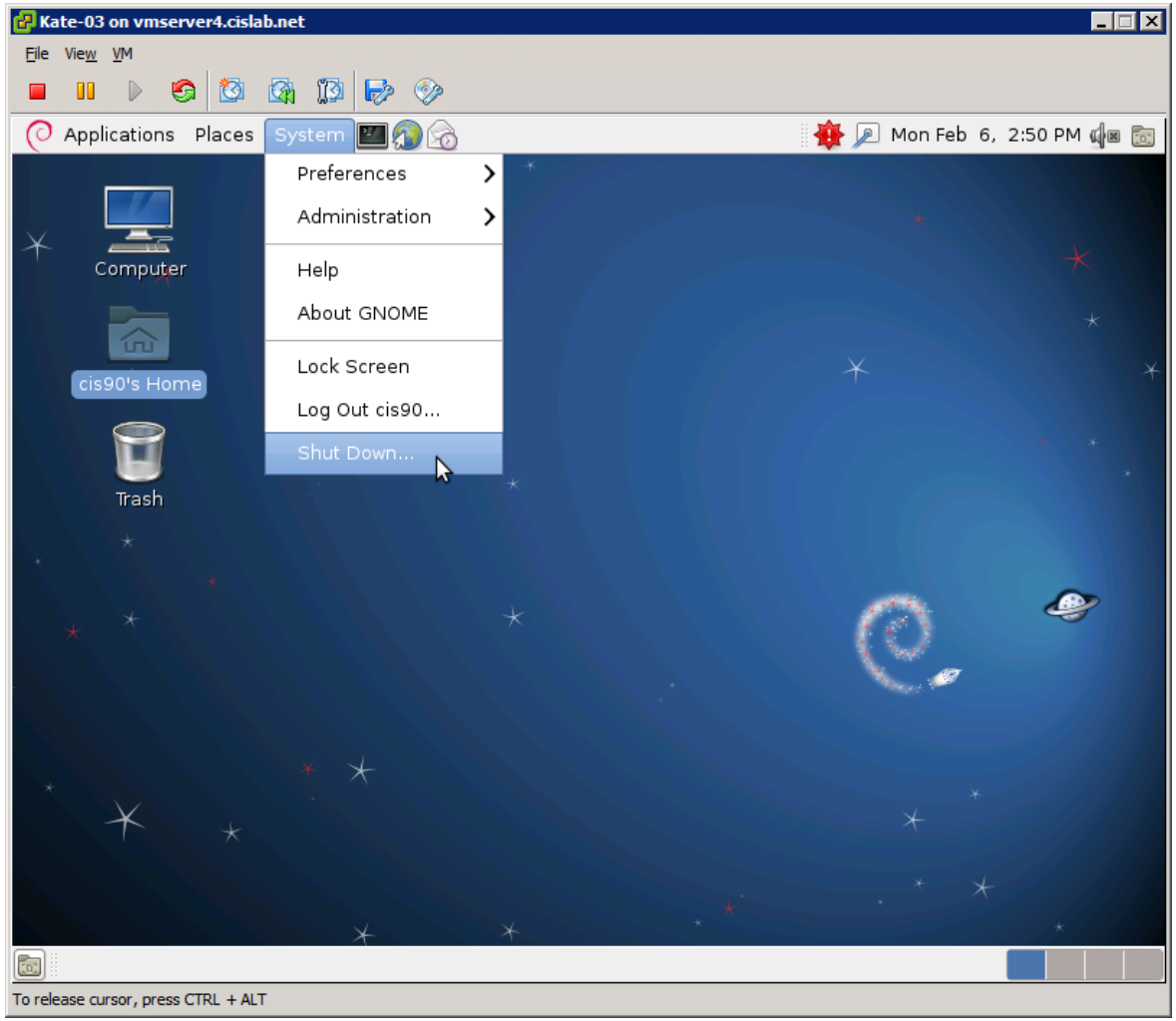
The Kate VM (Debian)

Terminal icon



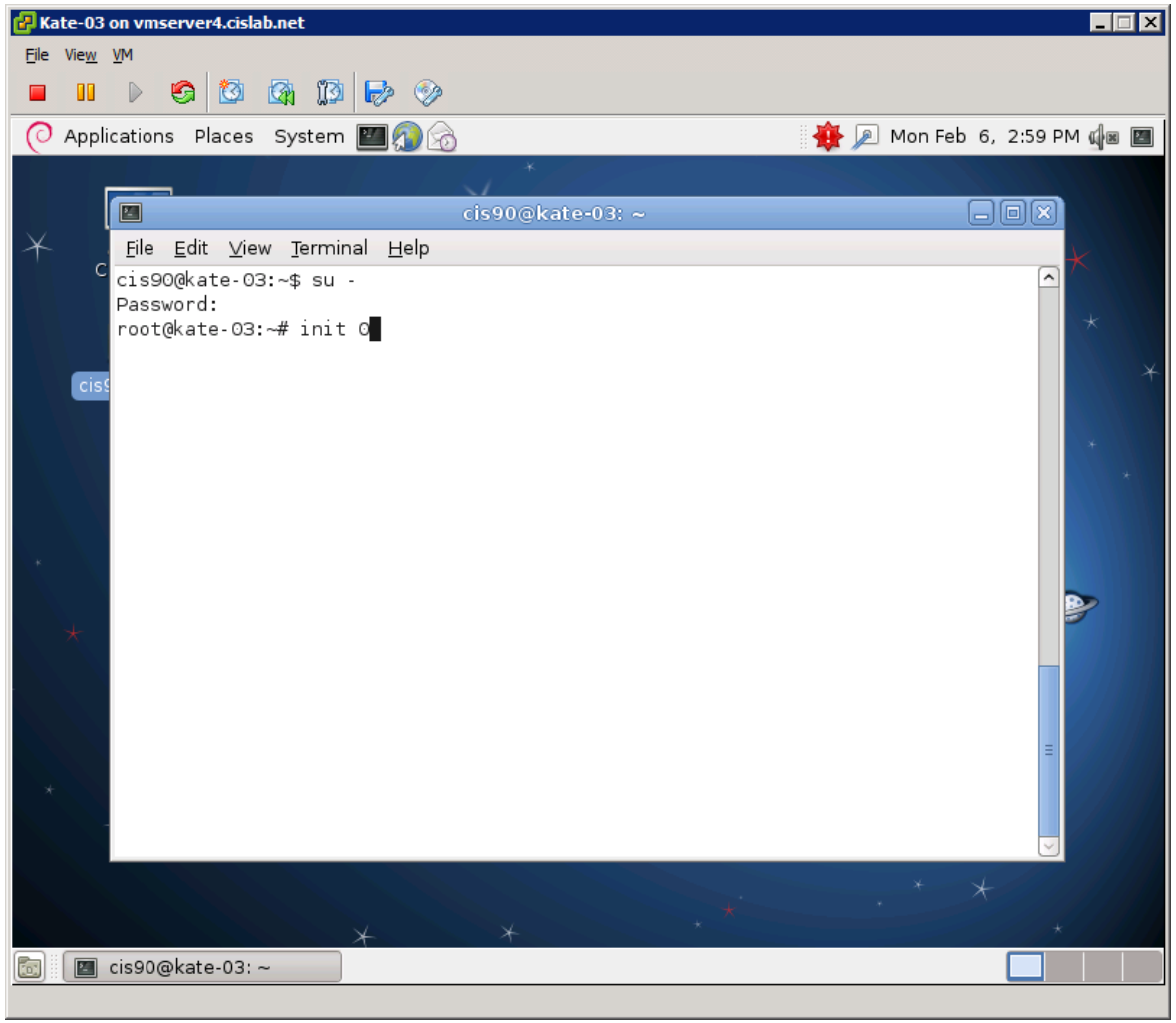
Click the Terminal icon on the panel to run a graphical terminal

The Kate VM (Debian)



To shutdown: System > Shut Down...

The Kate VM (Debian)

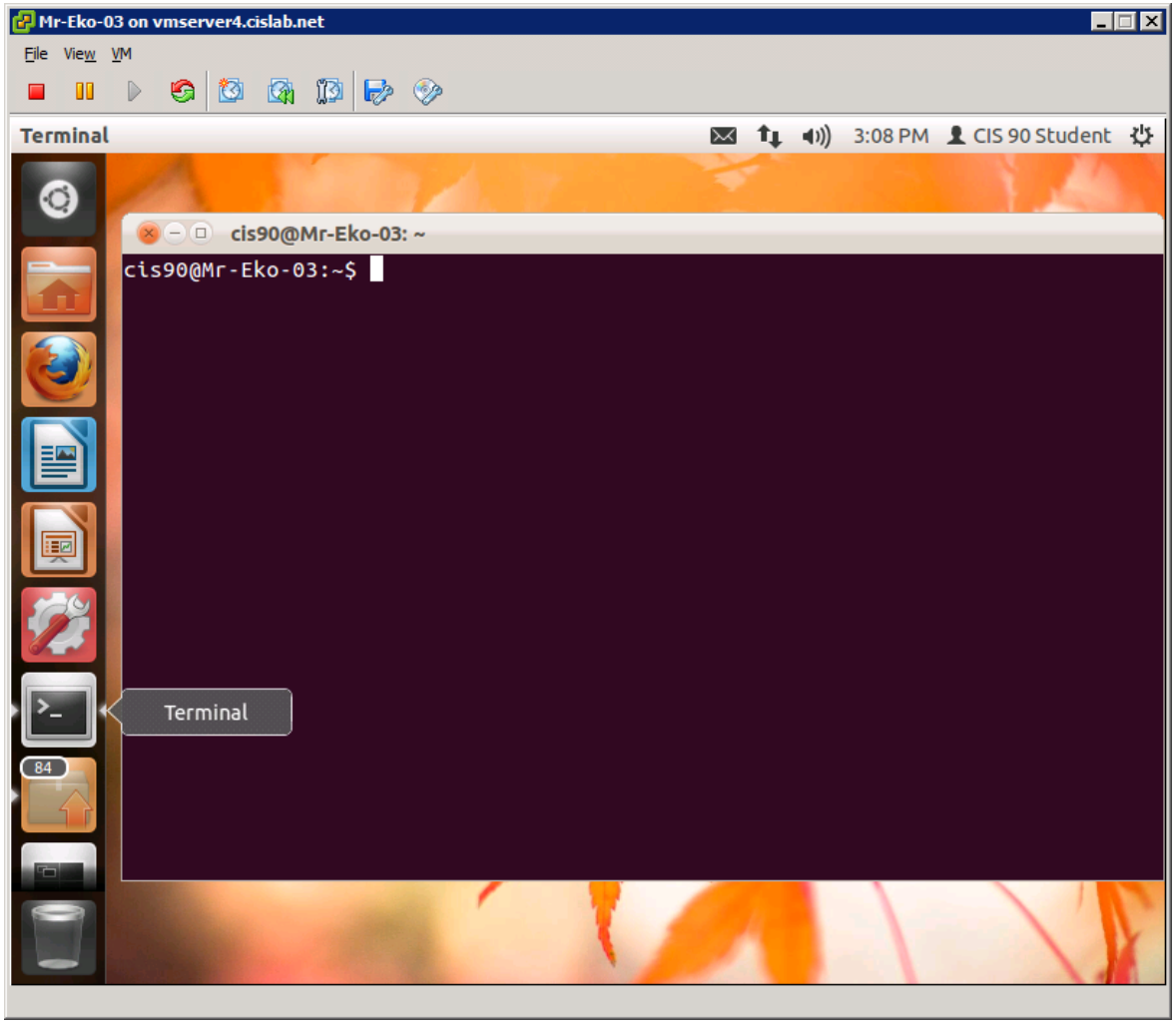


Use **su -** command to become root user

Another way to shutdown: Become root user, then issue **init 0** command

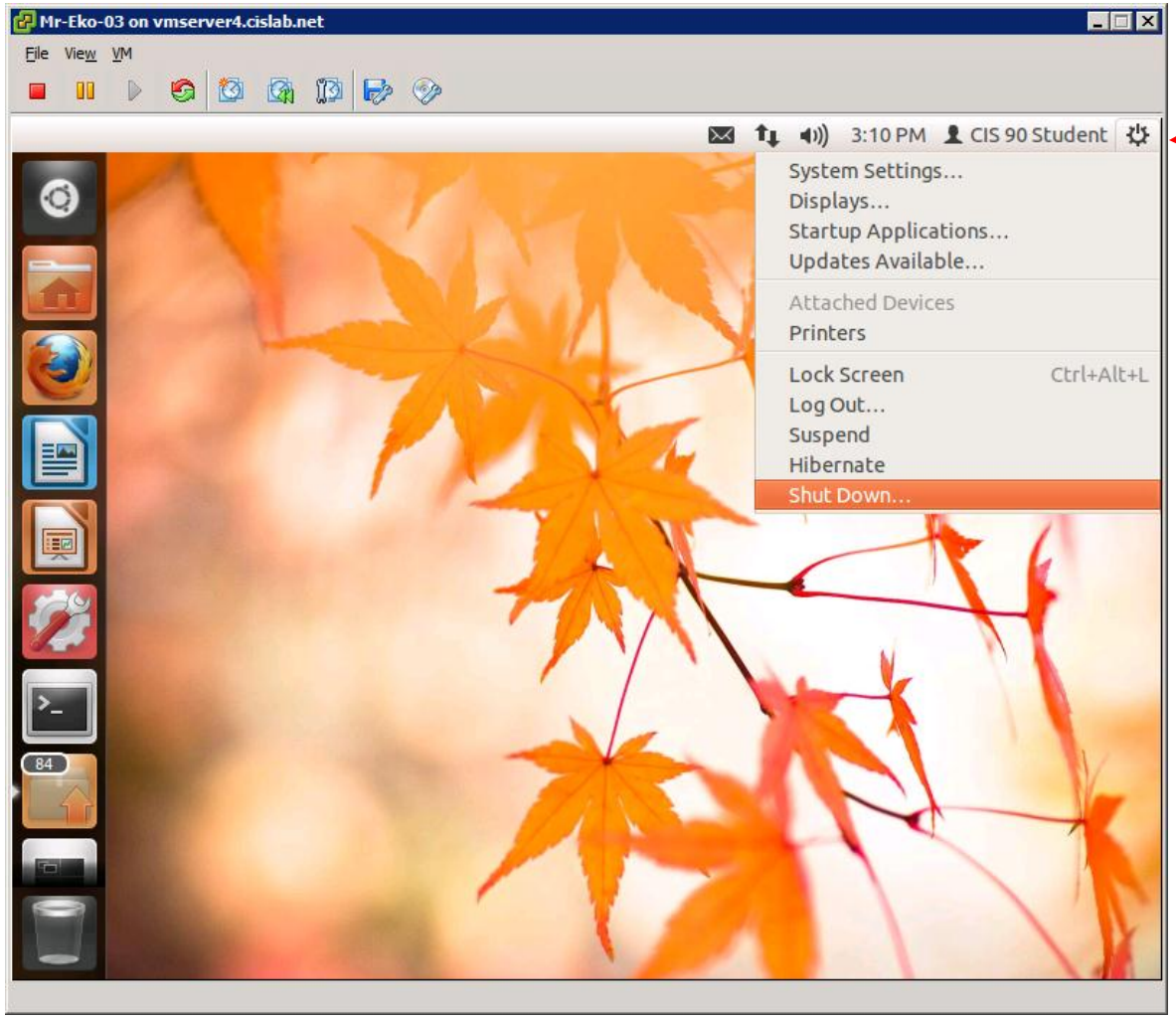
The Mr-Eko VM (Ubuntu)

Terminal icon



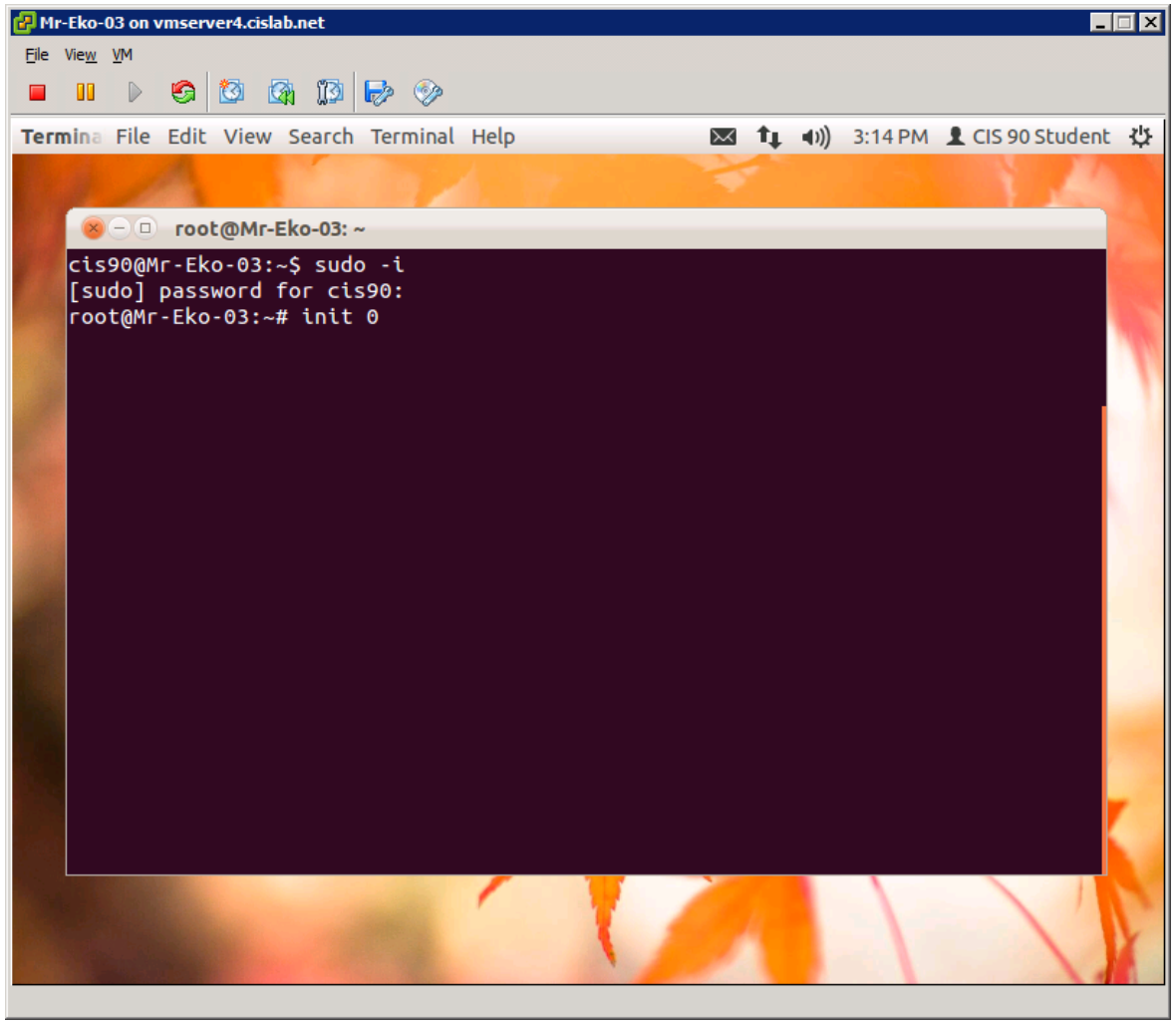
Click the Terminal icon on the panel to run a graphical terminal

The Mr-Eko VM (Ubuntu)



To shutdown: Click Power icon, then Shut Down...

The Mr-Eko VM (Ubuntu)

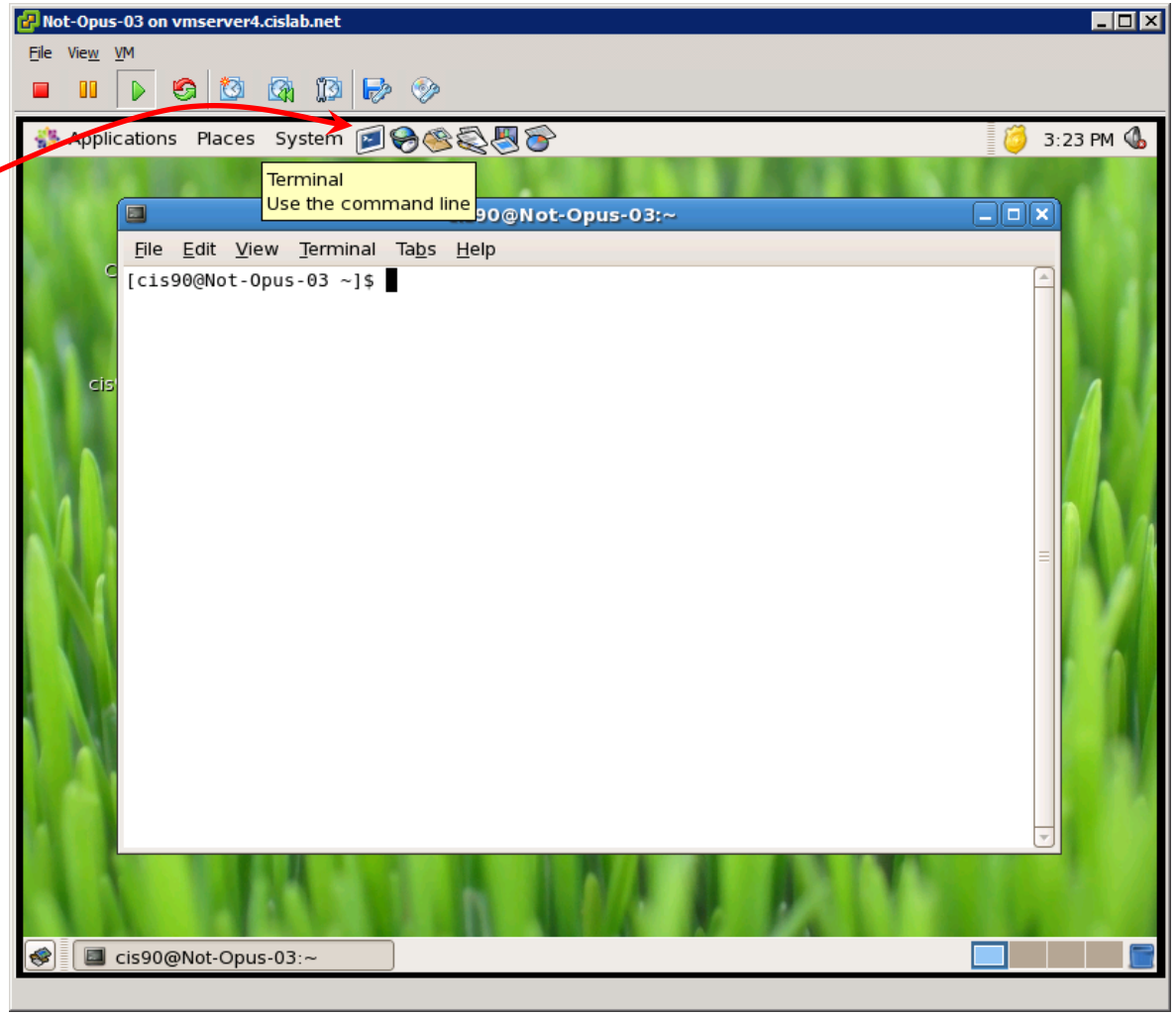


Use **sudo –i** command to become root user

Another way to shutdown: Become root user, then issue **init 0** command

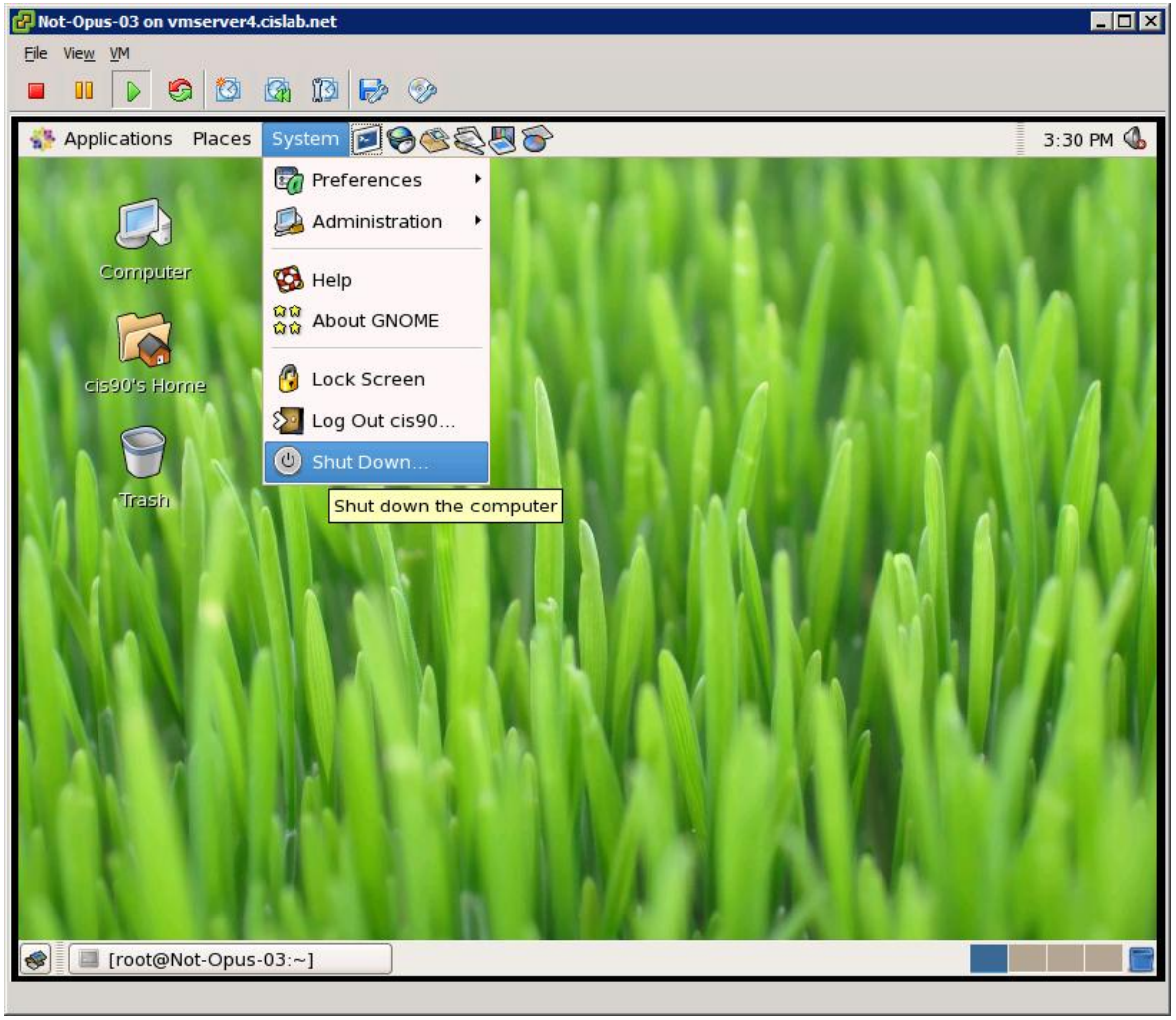
The Not-Opus VM (CentOS)

Terminal icon



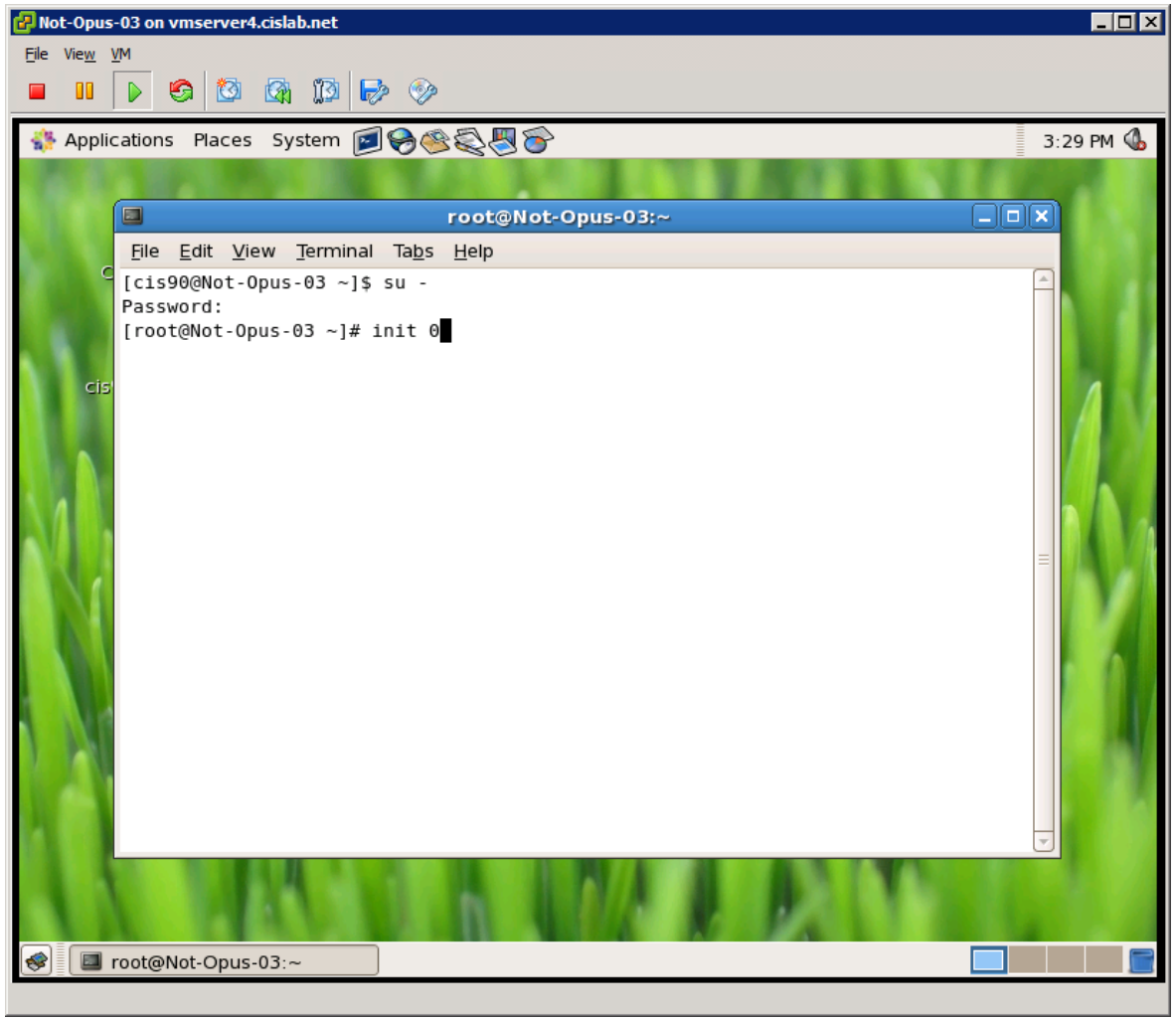
Click the Terminal icon on the panel to run a graphical terminal

The Not-Opus VM (CentOS)



To shutdown: System > Shut Down...

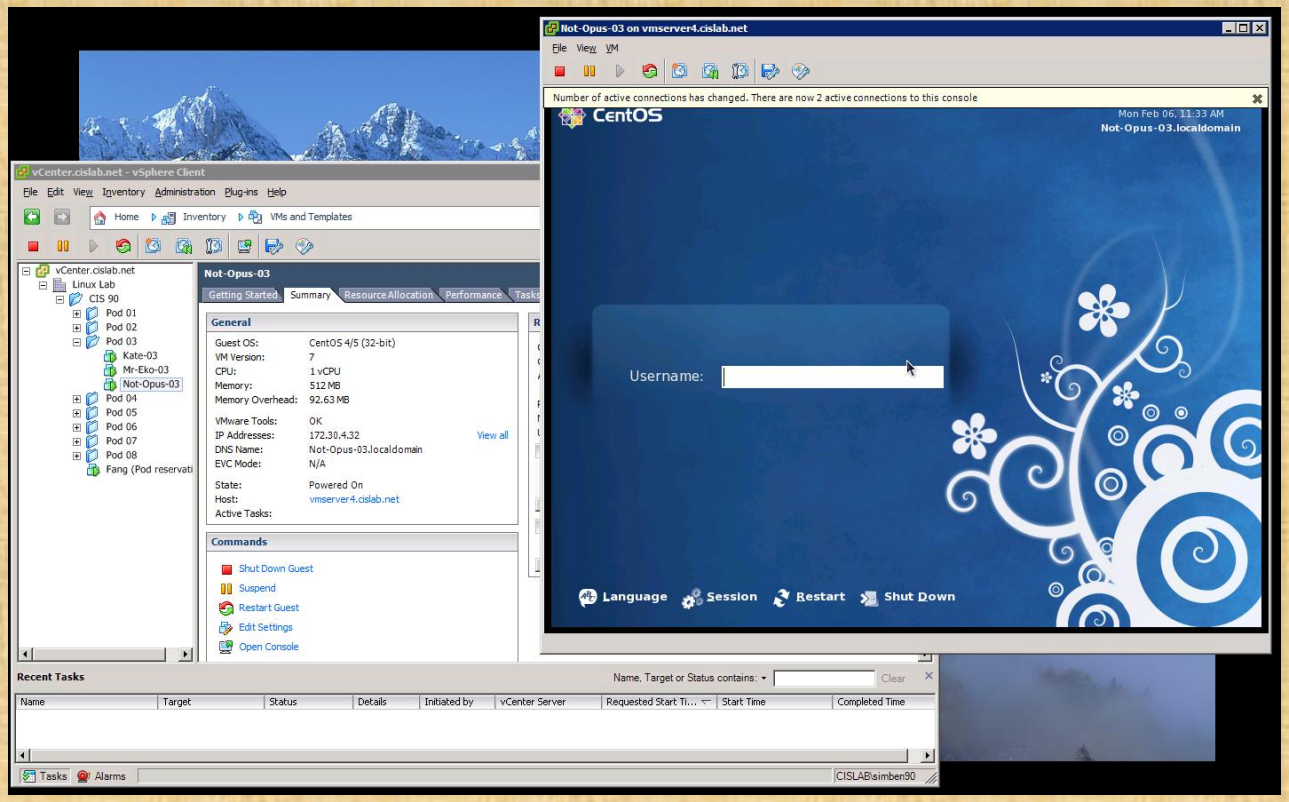
The Not-Opus VM (CentOS)



Use **su -** command to become root user

Another way to shutdown: Become root user, then use **init 0** command

Class Activity



Try logging into CIS VLab with your own credentials

- View Pod reservations on Fang
- View one or more VMs
- Close VMware vSphere when finished

Using Sun

From a classroom or CIS Lab station:

Putty to 172.30.4.18

From a VLab VM command line:

ssh *username***@sun**

or

ssh *username***@172.30.4.18**

On Sun, login using your VLab credentials (username and password)

Class Activity

1) Login into Sun using your VLab username and password:

   - Classroom students:

     Putty to 172.30.4.18

   - Remote students:

     Reserve a VLab VM
     Power on the VM, login as cis90 and use:

     ssh *username***@172.30.4.18**

Test Prep

Test next week

- 30 points, plus some extra credit
- 5 flashcard questions
  - Taken directly from the flashcards on the web site
- 25 operational questions
  - You can verify your answers using Sun, VLab VMs and Opus
- Open book, open notes, open computer
- To be taken during the last half of class
- Should take about 60-90 minutes, however if you need extra time, you can turn it in no later than 11:59PM.
- PDF form format. Fill out the form, save it and email it as an attachment to the instructor when finished cc'ing yourself.

How to prepare for the test:

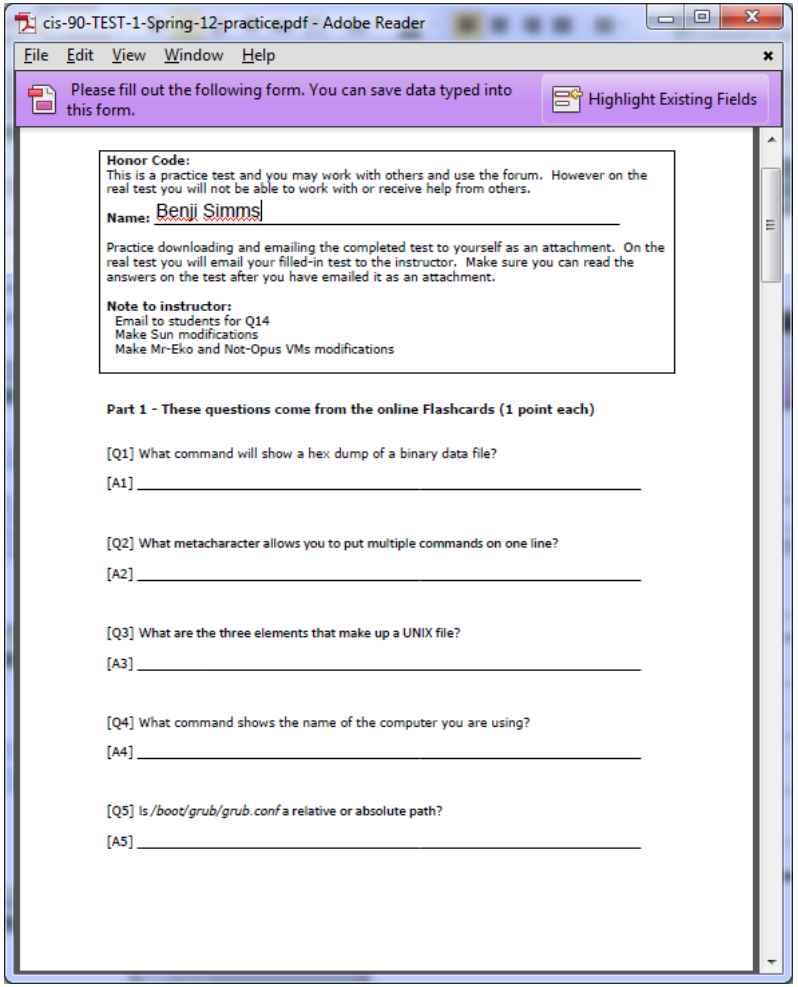
- Review slides for Lessons 1-5 (download and make sure you know how to electronically search PDFs)
- DO THE PRACTICE TEST
- Compare your practice test answers and methods used with others on the forum
- Make notes on the steps you took to answer each question so you can use them again on the real test
- Go through the Lesson 1-5 flashcards till you feel comfortable with the material
- Practice, practice, practice ... repeating Labs 1-4 never hurts!

First Test

1. Example flash card question:
   What is the program called that prompts you for a command, then locates that command and executes it?

2. Example operational question:
   From your home directory change to the Poems/Yeats/ directory. What one-liner (one ore more commands followed by Enter) would clear the screen and print the last line of all three Yeats poems without having to type the names of each individual poem file name?

Practice Test



A practice test is available on the web site Calendar page.

Download it, open with Adobe Reader, fill in with your answers, and save it.

Make sure you can email it as an attachment to yourself to verify your answers were saved.

You may need to download the latest version of Adobe Reader if you have problems filling it out.

Wrap up

New commands:

NA	NA

New Files and Directories:

NA	NA

Next Class

Assignment: Check Calendar Page on web site to see what is coming up.

No Quiz
No Lab due
Test !

Backup