Lesson Module Checklist
- Slides
- WB converted

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

- Dog script examples ready

- Materials uploaded
- Backup slides, CCC info, handouts on flash drive
- Spare 9v battery for mic

Student Learner Outcomes

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

Introductions and Credits

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

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

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

Student checklist
(How to attend from home or in the classroom)

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

2) Download the presentation slides for today's lesson for easier viewing

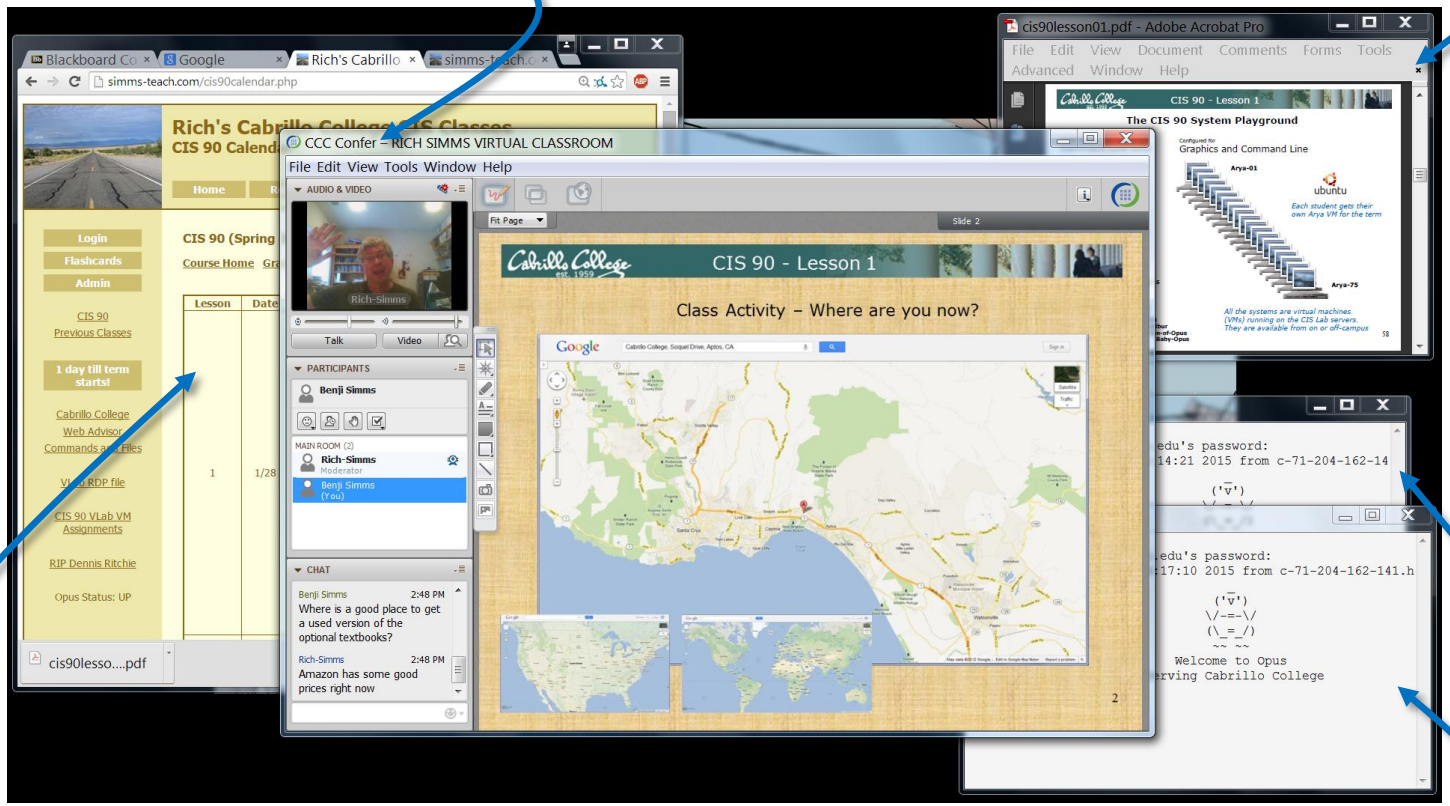
3) Click Enter virtual classroom to join CCC Confer session

4) Connect to Opus using Putty or ssh command

Student checklist
(How to layout your screen when attending class)

CCC Confer

Downloaded PDF of Lesson Slides

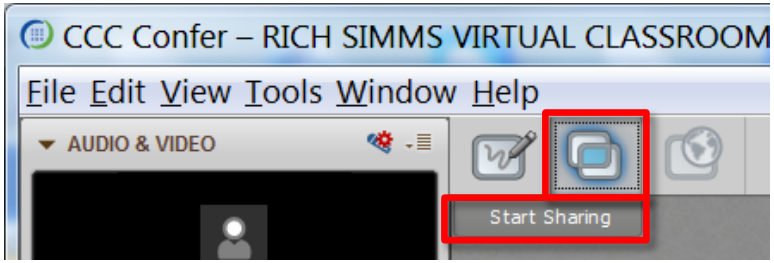


CIS 90 website Calendar page

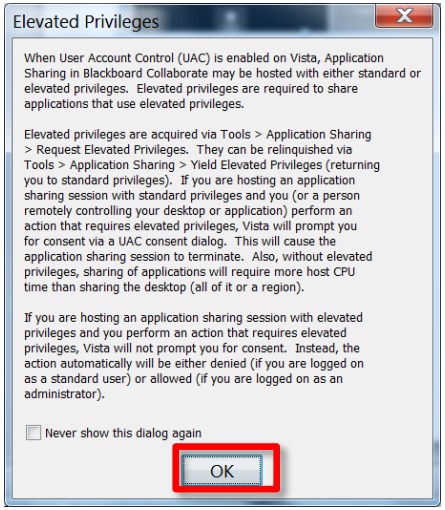
One or more login sessions to Opus

Student checklist
(To share your desktop with the class)

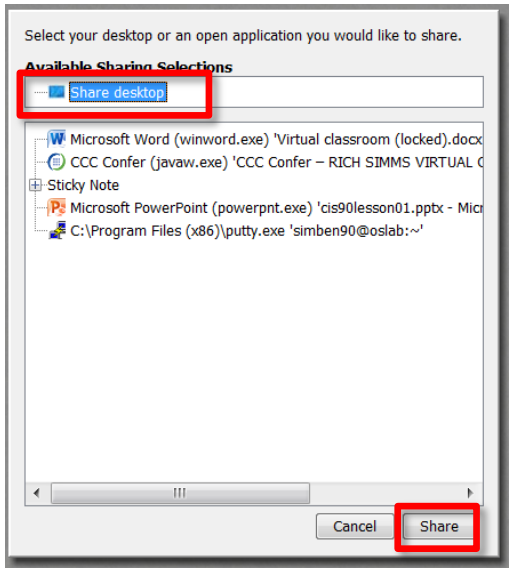
1) Instructor gives you sharing privileges



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



3) Click OK button.



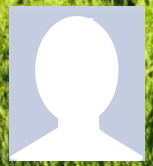
4) Select "Share desktop" and click Share button.

Instructor: **Rich Simms**
Dial-in: **888-886-3951**
Passcode: **136690**


Al

Tim

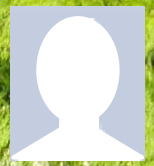

Emilio


Chris


Eddie


Clara


Ryan

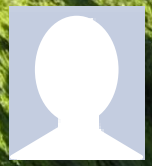

Ethan


Monte


Cameron

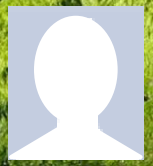

Django

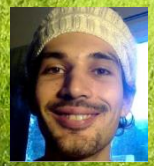

Tess


Ahmad

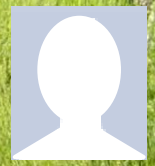

Mike


Mario

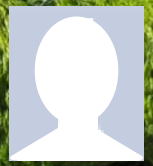

Roberto

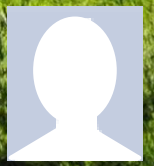

Benji

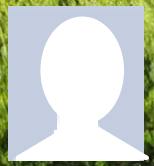

John


Efrain


Mateo


Ian

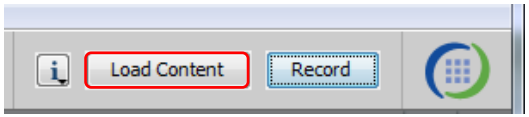

Abraham C.


Abraham N.

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

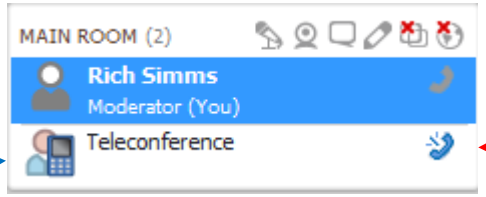
Rich's CCC Confer checklist - setup

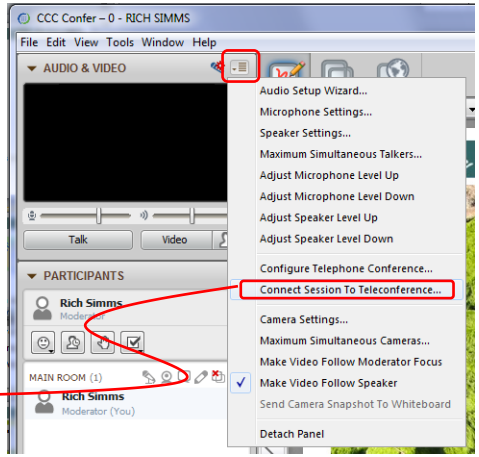
[ ] Preload White Board



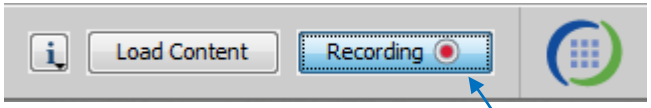
[ ] Connect session to Teleconference

Session now connected to teleconference





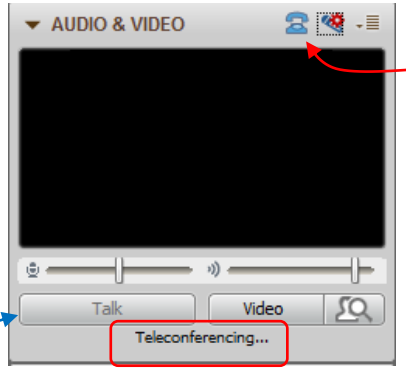
[ ] Is recording on?



Red dot means recording

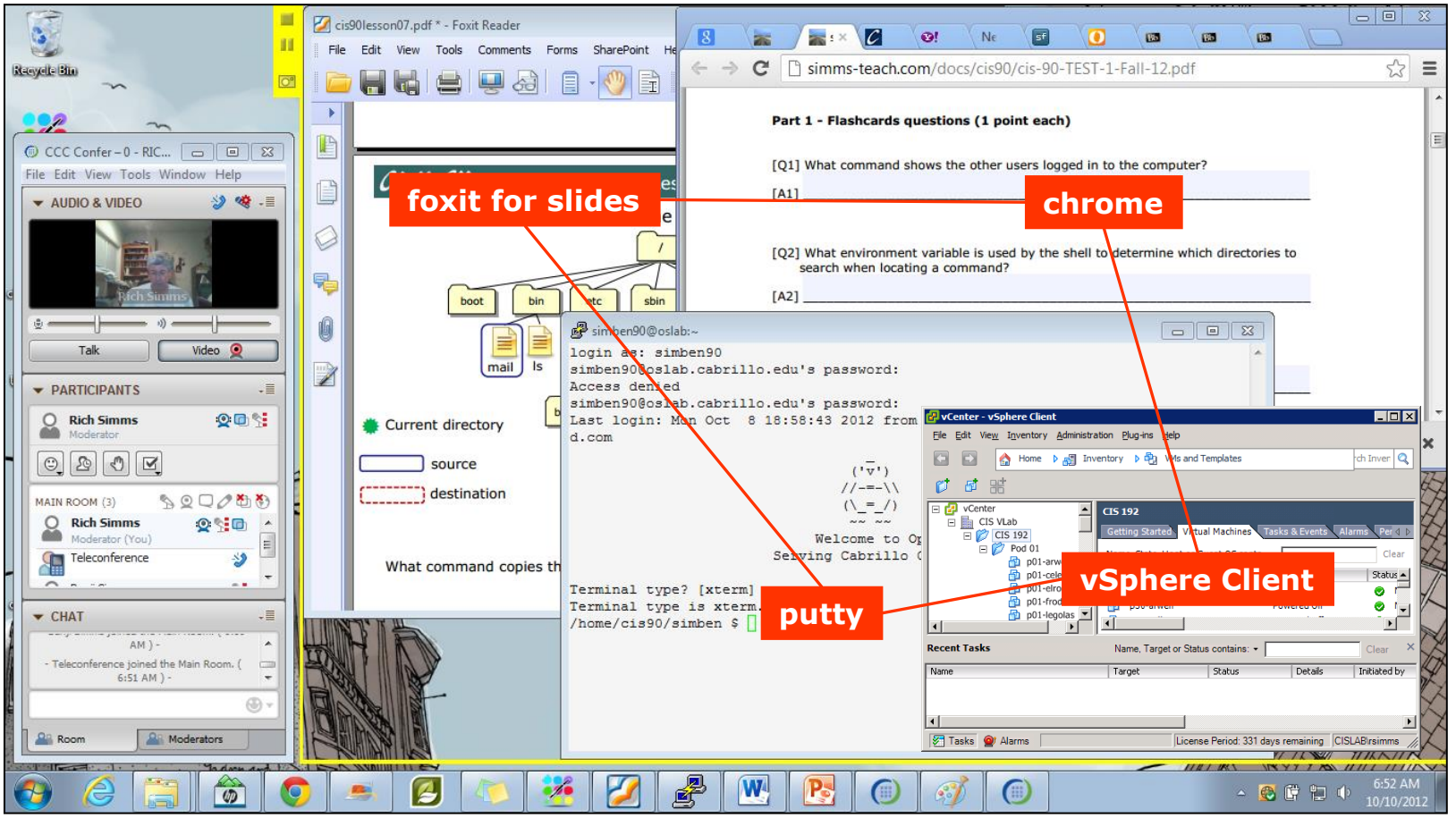
[ ] Use teleconferencing, not mic

Should be greyed out



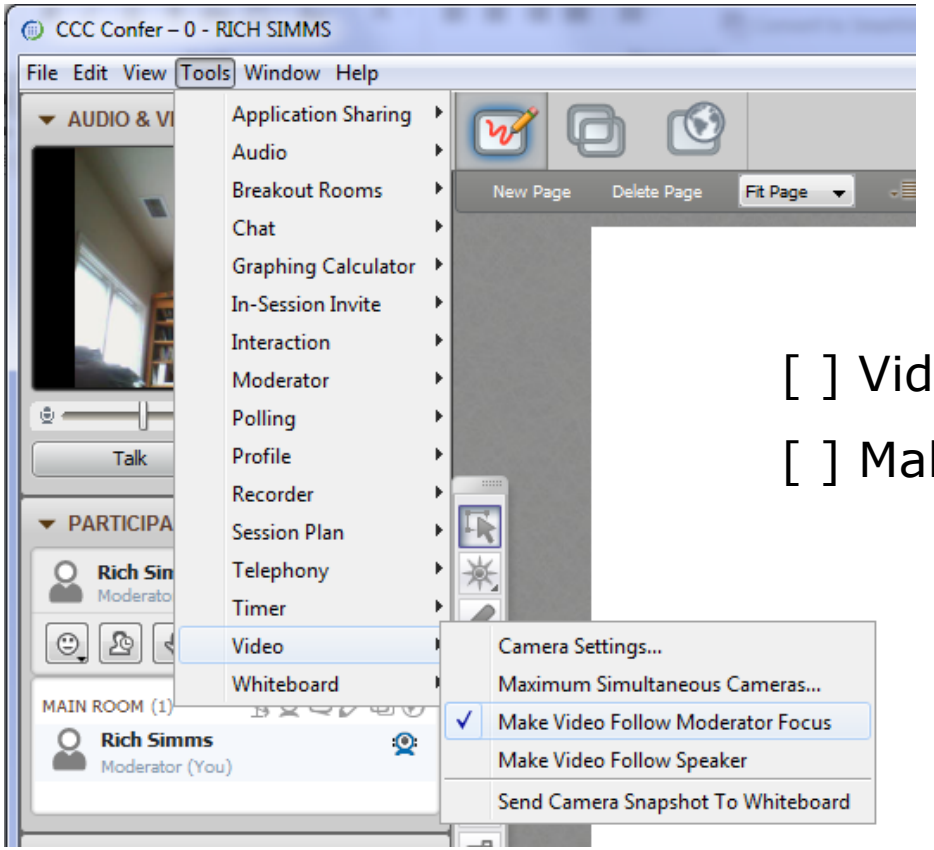
Should show as this live "off hook" telephone handset icon and the Teleconferencing … message displayed

Rich's CCC Confer checklist - app layout



[ ] layout and share apps

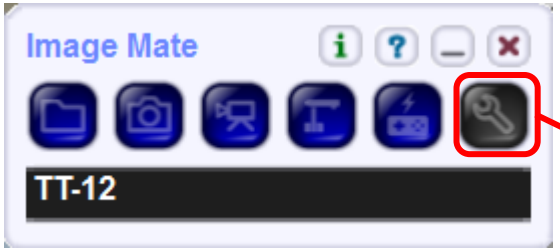
Rich's CCC Confer checklist - video

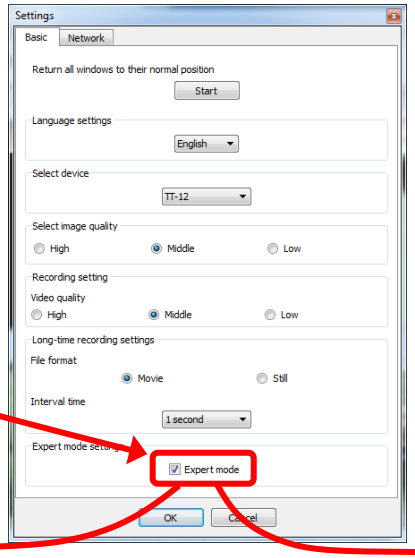


[ ] Video (webcam)

[ ] Make Video Follow Moderator Focus

Rich's CCC Confer checklist - Elmo



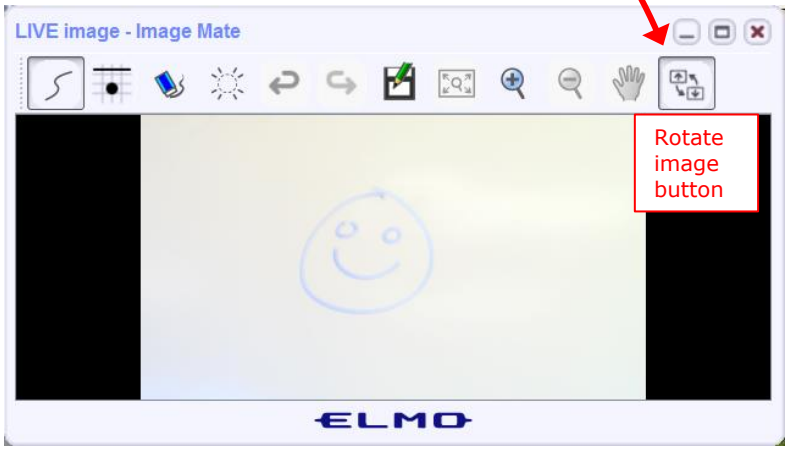


The "rotate image" button is necessary if you use both the side table and the white board.

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

Elmo rotated down to view side table

Elmo rotated up to view white board

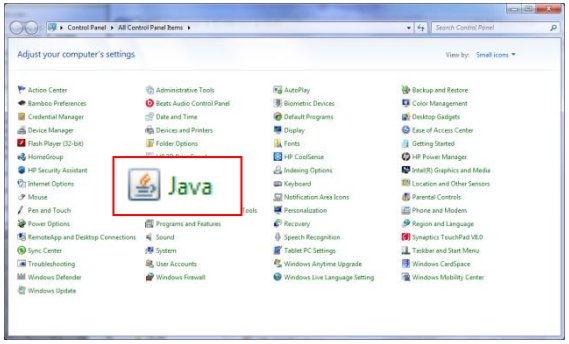


Run and share the Image Mate program just as you would any other app with CCC Confer

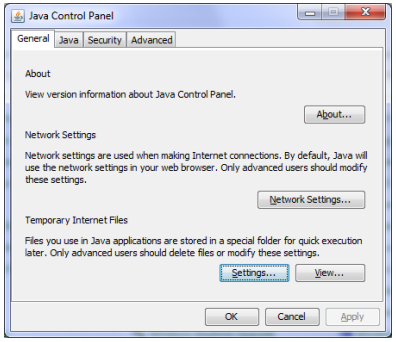
Rich's CCC Confer checklist - universal fix

Universal Fix for CCC Confer:
1) Shrink (500 MB) and delete Java cache
2) Uninstall and reinstall latest Java runtime

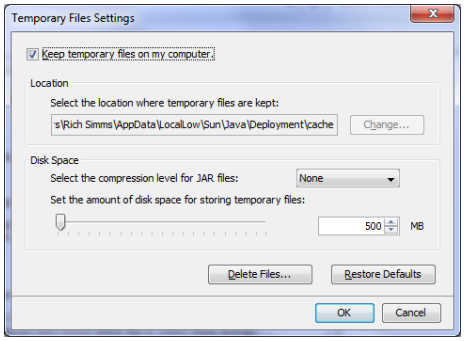
Control Panel (small icons)



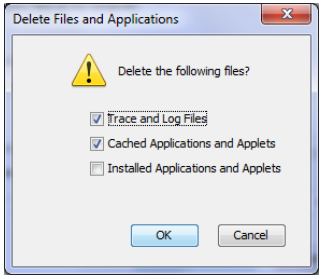
General Tab > Settings…



500MB cache size



Delete these



Google Java download

Quiz

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

See electronic white board

email answers to: risimms@cabrillo.edu

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

Quiz

No Quiz Today !

More Shell Scripting

Objectives	Agenda
• Transfer files between computers • Archive files using tar • Learn some scripting techniques	• No Quiz • Questions from last week • scp • tar • tar + scp • Housekeeping • Refresh on shell scripts • Project • Review how scripts are executed • Scripting tips - vi • Scripting tips - sleep • Scripting tips $(cmd) and `cmd` • Scripting tips - field extraction • Scripting tips - simple if • Scripting tips - or logic • Scripting tips - and logic • Scripting tips - file types • Scripting tips - if-then-else • Scripting tips - set command • Scripting tips - color • Scripting tips - user <-> home directory • Scripting tips - simple for loop • Wrap up

Sound Check

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

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

Questions

Questions?

Lesson material?

Labs? Tests?

How this course works?

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

Who questions much, shall learn much, and retain much.
- Francis Bacon

If you don't ask, you don't get.
- Mahatma Gandhi

Chinese Proverb	他問一個問題，五分鐘是個傻子，他不問一個問題仍然是一個傻瓜永遠。 *He who asks a question is a fool for five minutes; he who does not ask a question remains a fool forever.*

scp

Copying files between systems

ssh protocol

Secure Shell Protocol

- Allows secure (encrypted) connections between computers
  - **ssh** command - for login and running remote commands
  - **scp** command - for copying files between systems

Copying files on same system

cp command syntax:

cp *<source file> <target file>*

cp *<source file> <target directory>*

cp *<source file> <source file> <target directory>*

cp -r *<source directory branch> <target directory>*

Copying files between systems

Some **scp** command syntax examples:

Capital P (unlike ssh command which uses little p)

scp -P <port> <username@host>**:***<source file> <target file>*

scp -P <port> <username@host>**:***<source file> <target directory>*

scp -P <port> <username@host>**:***<multiple source files> <target directory>*

scp **-r** -P <port> <username@host>**:***<source directory branch> <target directory>*

*When copying files between systems it is necessary to use specify the **hostname** of the remote system. You may also have to specify the **username** if different and the **port** if it is not 22.*

scp practice

Log into your Arya VM

```
/home/cis90/simben $ ssh cis90@arya-02
cis90@arya-02's password:
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-44-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

226 packages can be updated.
0 updates are security updates.

   _____
  /  _  \_______ ___.__._____
 /  /_\  \_  __ <   |  |\__  \
/    |    \  | \/\___  | / __ \_
\____|__  /__|   / ____|(____  /
        \/       \/          \/
       Winter is coming

Last login: Sat Feb 21 18:23:19 2015 from opus.cis.cabrillo.edu
cis90@Arya-02:~$
```

Log into your <u>own</u> Arya VM

We've just logged into the Arya VM from Opus

FYI, alternate ssh commands that would also work from Opus:

```
ssh -p 22 cis90@arya-02
ssh -p 22 cis90@arya-02.cis.cabrillo.edu
```

Copy one file from Opus

scp -P <port> <username@host>:*<source file>* *<target directory>*
scp -P <port> <username@host>:*<source file>* *<target file>*

```
cis90@Arya-02:~$ scp simben90@opus:letter .
simben90@opus's password:
letter                                        100% 1044     1.0KB/s   00:00
cis90@Arya-02:~$

cis90@Arya-02:~$ scp simben90@opus:letter letter
simben90@opus's password:
letter                                      100% 1044     1.0KB/s   00:00
cis90@Arya-02:~$
```

FYI, from off-campus use either of these commands to copy to your home system:

```
scp -P 2220 simben90@oslab.cis.cabrillo.edu:letter .
scp -P 2220 simben90@oslab.cis.cabrillo.edu:letter letter
```

Use your own Opus username and password when trying this

Copy your Shakespeare sonnet files on Opus

scp -P <port> <username@host>**:**<*multiple source files*> <*target directory*>

```
cis90@Arya-02:~$ scp simben90@opus:poems/Shakespeare/sonnet* .
simben90@opus's password:
sonnet1                                       100%  614     0.6KB/s   00:00
sonnet10                                      100%  620     0.6KB/s   00:00
sonnet11                                      100%  689     0.7KB/s   00:00
sonnet15                                      100%  618     0.6KB/s   00:00
sonnet17                                      100%  647     0.6KB/s   00:00
sonnet2                                       100%  631     0.6KB/s   00:00
sonnet26                                      100%  601     0.6KB/s   00:00
sonnet3                                       100%  615     0.6KB/s   00:00
sonnet35                                      100%  598     0.6KB/s   00:00
sonnet4                                       100%  588     0.6KB/s   00:00
sonnet5                                       100%  622     0.6KB/s   00:00
sonnet7                                       100%  581     0.6KB/s   00:00
sonnet9                                       100%  620     0.6KB/s   00:00
cis90@Arya-02:~$
```

FYI, from off-campus use this command to copy to your home system:

```
scp -P 2220 simben90@oslab.cis.cabrillo.edu:poems/Shakespeare/sonnet* .
```

Use your own Opus username and password when trying this

Recursively copy your entire poems/ branch on Opus

scp -r -P <port> <username@host>**:**<*source directory branch*> <*target directory*>

```
cis90@Arya-02:~$ scp -r simben90@opus:poems .
simben90@opus's password:
sonnet10                      100%  620     0.6KB/s   00:00
sonnet15                      100%  618     0.6KB/s   00:00
sonnet26                      100%  601     0.6KB/s   00:00
sonnet3                       100%  615     0.6KB/s   00:00
sonnet35                      100%  598     0.6KB/s   00:00
sonnet2                       100%  631     0.6KB/s   00:00
sonnet4                       100%  588     0.6KB/s   00:00
sonnet1                       100%  614     0.6KB/s   00:00
.1979.egg                     100%  733     0.7KB/s   00:00
sonnet11                      100%  689     0.7KB/s   00:00
sonnet7                       100%  581     0.6KB/s   00:00
sonnet5                       100%  622     0.6KB/s   00:00
sonnet9                       100%  620     0.6KB/s   00:00
sonnet17                      100%  647     0.6KB/s   00:00
mooncat                       100%  856     0.8KB/s   00:00
1982.egg                      100%  134     0.1KB/s   00:00
whitebirds                    100%  863     0.8KB/s   00:00
old                           100%  520     0.5KB/s   00:00
1978.egg                      100%  734     0.7KB/s   00:00
nursery                       100%  779     0.8KB/s   00:00
ant                           100%  237     0.2KB/s   00:00
twilight                      100%  654     0.6KB/s   00:00
artichoke                     100% 1436     1.4KB/s   00:00
dog                           100% 1842     1.8KB/s   00:00
.1983.egg                     100%  734     0.7KB/s   00:00
twister                       100%  151     0.2KB/s   00:00
bird                          100%  975     1.0KB/s   00:00
woman                         100% 1273     1.2KB/s   00:00
1984.egg                      100%  404     0.4KB/s   00:00
you                           100%  236     0.2KB/s   00:00
diner                         100%  741     0.7KB/s   00:00
eden                          100%  189     0.2KB/s   00:00
hope                          100%  343     0.3KB/s   00:00
charm                         100%  203     0.2KB/s   00:00
forget                        100%  228     0.2KB/s   00:00
.1988.egg                     100%  405     0.4KB/s   00:00
tiger                         100%  115     0.1KB/s   00:00
1991.egg                      100%  725     0.7KB/s   00:00
jerusalem                     100%  582     0.6KB/s   00:00
cis90@Arya-02:
```

FYI, from off-campus use this command copy to your home system:

```
scp -r -P 2220 simben90@oslab.cis.cabrillo.edu:poems .
```

Use your own Opus username and password when trying this

tar

tar command

- To simplify file transfers, Windows users typically "zip" multiple files together into a single "zipfile".
- UNIX/Linux users use the **tar** command to do this and "archive" multiple files into a single "tarball".

Basic tar command syntax

verbose

specify the archive file

```
tar -c -v -f <tarfile> <files-or-directory-to-archive>
```

***c**reates an archive*

```
tar -t -v -f <tarfile>
```

*views an archive's **t**able of contents*

```
tar -x -v -f <tarfile>
```

*e**x**tracts archive files to the current directory*

Basic tar command syntax

The tar command was written before POSIX command line conventions

tar -c -v -f *<tarfile> <files-or-directory-to-archive>*
tar cvf *<tarfile> <files-or-directory-to-archive>*

are equivalent

tar -t -v -f *<tarfile>*
tar tvf *<tarfile>*

are equivalent

tar -x -v -f *<tarfile>*
tar xvf *<tarfile>*

are equivalent

Example

Backup and restore a directory

Archive your Blake directory of poems

```
/home/cis90/simben $ cd poems/
/home/cis90/simben/poems $ ls -l Blake/
total 8
-r--r--r--. 1 simben90 cis90 582 Nov  7 06:40 jerusalem
-r--r--r--. 1 simben90 cis90 115 Nov  7 06:40 tiger
/home/cis90/simben/poems $ tar cvf blake.tar Blake/
Blake/
Blake/tiger
Blake/jerusalem
/home/cis90/simben/poems $
```

pathname to directory to archive

name of archive file (tarball)

***c**reate verbose file*

Example

Backup and restore a directory

table of contents
verbose
file

name of
archive file
(tarball)

```
/home/cis90/simben/poems $ tar tvf blake.tar
drwxr-xr-x simben90/cis90    0 2013-11-07 06:40 Blake/
-r--r--r-- simben90/cis90  115 2013-11-07 06:40 Blake/tiger
-r--r--r-- simben90/cis90  582 2013-11-07 06:40 Blake/jerusalem
/home/cis90/simben/poems $
```

View new archive's table of contents

Example

Backup and restore a directory

Clobber (remove) your directory of Blake poems

```
/home/cis90/simben/poems $ rm -rf Blake/
/home/cis90/simben/poems $ ls -l Blake
ls: cannot access Blake: No such file or directory
/home/cis90/simben/poems $
```

Uh oh, we just lost all of our Blake poems!

Example

Backup and restore a directory

No problem, we have a backup!

```
/home/cis90/simben/poems $ ls -l Blake
ls: cannot access Blake: No such file or directory
/home/cis90/simben/poems $ tar xvf blake.tar
Blake/
Blake/tiger
Blake/jerusalem
/home/cis90/simben/poems $
/home/cis90/simben/poems $ ls -l Blake
total 8
-r--r--r--. 1 simben90 cis90 582 Nov  7 06:40 jerusalem
-r--r--r--. 1 simben90 cis90 115 Nov  7 06:40 tiger
/home/cis90/simben/poems $
```

extract verbose file

name of archive file (tarball)

Restore your directory of Blake poems

tar

+

scp

Example

Copy archived directory to another system

Backup your bin directory

```
/home/cis90/simben $ ls bin
app     datecal         hi    I          myscript.v1   tryme
banner  enlightenment   home  myscript   treed         zoom

/home/cis90/simben $ tar cvf bin.tar bin/
bin/
bin/enlightenment
bin/treed
bin/zoom
bin/myscript.v1
bin/app
bin/home
bin/hi
bin/myscript
bin/I
bin/tryme
bin/datecal
bin/banner
/home/cis90/simben $
```

create verbose file

name of archive file (tarball)

pathname to directory to archive

Example

Copy archived directory to another system

View your bin archive

```
/home/cis90/simben $ ls -l bin.tar
-rw-rw----. 1 simben90 cis90 40960 Dec  2 07:47 bin.tar

/home/cis90/simben $ tar tvf bin.tar
drwxr-x--- simben90/cis90    0 2014-12-02 07:41 bin/
-r-xr-xr-- simben90/cis90 3442 2014-08-06 11:52 bin/enlightenment
-r-xr-x--- simben90/cis90  190 2001-07-20 15:04 bin/treed
-r-xr-x--- simben90/cis90   74 2001-07-20 15:18 bin/zoom
-rwxrwx--x simben90/cis90  546 2014-12-02 07:40 bin/myscript.v1
-r-xr-x--- simben90/cis90  220 2004-04-22 18:51 bin/app
-rwxr-xr-x simben90/cis90  103 2014-11-13 10:16 bin/home
-r-xr-x--- simben90/cis90  107 2001-07-20 21:06 bin/hi
-rwxrwxr-x simben90/cis90 10513 2014-12-02 07:41 bin/myscript
-r-xr-x--- simben90/cis90   375 2003-10-20 18:36 bin/I
-r-xr-x--- simben90/cis90   174 2004-03-04 13:02 bin/tryme
-r-xr-x--- simben90/cis90   519 2014-08-06 11:53 bin/datecal
-r-xr-x--- simben90/cis90  6160 2003-08-28 22:39 bin/banner
/home/cis90/simben $
```

Example

Copy archived directory to another system

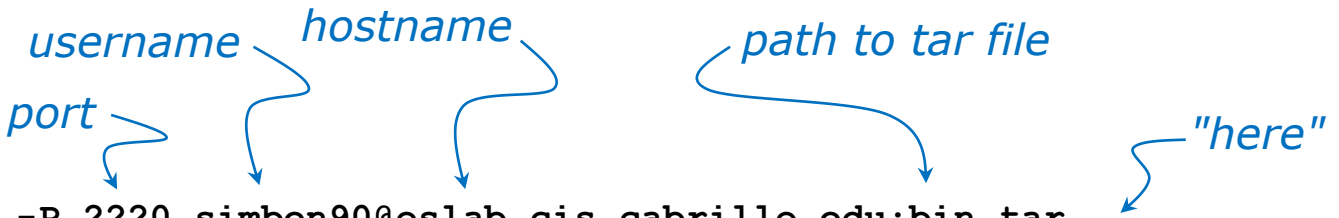
username *hostname*

```
/home/cis90/simben $ ssh cis90@arya-02
cis90@arya-02's password:
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-39-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

130 packages can be updated.
0 updates are security updates.

*** System restart required ***
   _____
  /     \_______ ___.__._____
 /  /_\  \_  __ <   |  |\__  \
/    |    \  | \/\___  | / __ \_
\____|__  /__|   / ____|(____  /
        \/       \/          \/
       Winter is coming

You have mail.
Last login: Tue Dec  2 07:21:57 2014 from opus.cis.cabrillo.edu
cis90@arya-02:~$
```

Login to your own Arya VM from Opus

Example

Copy archived directory to another system




```
cis90@arya-02:~$ scp -P 2220 simben90@oslab.cis.cabrillo.edu:bin.tar .
simben90@oslab.cis.cabrillo.edu's password:
bin.tar                                       100%   40KB  40.0KB/s
00:00

cis90@arya-02:~$ ls -l bin.tar
-rw-rw---- 1 cis90 cis90 40960 Dec  2 07:52 bin.tar
cis90@arya-02:~$
```

Note how archive files are shown in red

Copy your bin archive from Opus to Arya

Example

Copy archived directory to another system

```
cis90@arya-02:~$ tar xvf bin.tar
bin/
bin/enlightenment
bin/treed
bin/zoom
bin/myscript.v1
bin/app
bin/home
bin/hi
bin/myscript
bin/I
bin/tryme
bin/datecal
bin/banner
cis90@arya-02:~$
```

extract verbose file

name of archive file (tarball)

Extract your Opus bin directory to your Arya home directory

```
cis90@arya-02:~$ ls bin
app      datecal         hi    I          myscript.v1  tryme
banner   enlightenment   home  myscript   treed        zoom
cis90@arya-02:~$
```

Example

Copy archived directory to another system

```
cis90@Arya-02:~$ myscript
No command 'myscript' found, did you mean:
 Command 'pyscript' from package 'python-pyscript' (universe)
myscript: command not found
cis90@Arya-02:~$

cis90@arya-02:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/
usr/local/games
```

Oops, the local bin directory is not on the cis90 user's path!

Example

Copy archived directory to another system

```
cis90@Arya-02:~$ cd bin
cis90@Arya-02:~/bin$ ./myscript
/home/cis90/bin/myscript: line 44: finger: command not found
What is your first name? ^C
cis90@arya-02:~$
```

Hit Ctrl-C to abort myscript

Oops … the finger command used by Benji's script has not been installed on Arya

Example

Copy archived directory to another system

Use sudo to install finger as the root superuser

```
cis90@arya-02:~$ sudo apt-get install finger
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  finger
0 upgraded, 1 newly installed, 0 to remove and 145 not upgraded.
Need to get 0 B/17.3 kB of archives.
After this operation, 68.6 kB of additional disk space will be used.
Selecting previously unselected package finger.
(Reading database ... 290787 files and directories currently installed.)
Preparing to unpack .../finger_0.17-15_amd64.deb ...
Unpacking finger (0.17-15) ...
Processing triggers for man-db (2.6.7.1-1) ...
Setting up finger (0.17-15) ...
cis90@arya-02:~$
```

Example

Copy archived directory to another system

Run myscript file in the bin directory

```
cis90@Arya-02:~/bin$ ./myscript

    CIS, please Enter an option number from the list below:

    1) What is today?
    2) The users on Arya-02
    3) Warning, don't go here!!
    4) Sort current directory
    5) Back pat eCards
    6) Check IP forwarding status

    or enter Q to Quit

    Enter Your Choice:
```

We can ./ it so it will run without updating the path

Housekeeping

Next Class

Project is due next week!

1. No labs due today

2. There is a check script for Lab X2

3. There is no check script for Lab X1. To test permissions copy it to the /tmp directory and run it using the cis90 user account.

4. One week from now (see calendar)
   - Project due on by 11:59PM.
   - If you haven't started yet, now would be a good time!

5. Monday May 18th (see calendar)
   - Final Exam (Test #3) 7-9:50AM.
   - Extra credit labs are due by 11:59PM.

Make backup copies of your script

modify, debug, modify, debug, … rest

```
/home/cis90/simben/bin $ cp myscript myscript.v1
```

modify, debug, modify, debug, … rest

```
/home/cis90/simben/bin $ cp myscript myscript.v2
```

modify, debug, modify, debug, … rest

```
/home/cis90/simben/bin $ cp myscript myscript.v3
```

Fall 2015 Linux Classes and Prerequisites

CIS 90 Introduction to UNIX/Linux

Provides a technical overview of the UNIX/Linux operating system, including hands-on experience with commands, files, and tools.
Transfer Credit: CSU.

Section	Days	Times	Units	Instructor	Room
89005	W	01:00PM-04:05PM	3.00	R.Simms	OL
&	Arr.	Arr.		R.Simms	OL

Section 89005 is an ONLINE course. Meets weekly throughout the semester online during the scheduled times by remote technology with an additional 50 min online lab per week. For details, see instructor's web page at go.cabrillo.edu/online.

89006	W	01:00PM-04:05PM	3.00	R.Simms	828
&	Arr.	Arr.		R.Simms	OL

Section 89006 is a Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 50 min online lab per week. For details, see instructor's web page at go.cabrillo.edu/online.

CIS 191AB UNIX/Linux Installation, Configuration and Administration

Introduces skills required to administer UNIX/Linux systems. Prerequisite: CIS 90 or equivalent skills.

Section	Days	Times	Units	Instructor	Room
88994	Arr.	Arr.	4.00	M.Matera	OL

Section 88994 is an ONLINE course. For details, see instructor's web page at go.cabrillo.edu/online.

88995	TH	01:00PM-05:05PM	4.00	M.Matera	828
&	Arr.	Arr.		M.Matera	OL

Section 88995 is a Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 50 min online lab per week. For details, see instructor's web page at go.cabrillo.edu/online.

CIS 81 Networking Fundamentals and Theory (Cisco CCNA 1)

Presents networking protocols, standards, concepts, and terminology including Ethernet, ARP, ICMP, IP addressing, subnetting, switches, hubs, routers, TCP, UDP, OSI Model and other standards and protocols. Hybrid Requisite: Completion of or concurrent enrollment in CIS 72. Recommended Preparation: Eligibility for MATH 154.
Transfer Credit: CSU.

Section	Days	Times	Units	Instructor	Room
89002	M	09:30AM-01:35PM	4.00	R.Graziani	828
&	Arr.	Arr.		R.Graziani	OL

Section 89002 is a Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 50 min online lab per week. Students will be required to show that they meet the course prerequisites. For details, see instructor's web page at go.cabrillo.edu/online.

CIS 81 is a prerequisite to CIS 192 Linux Network Administration offered next spring

Final Exam

Test #3 (final exam) is MONDAY May 18 7-9:50AM

Monday	5/18	**Test #3 (the final exam)** **Time** • 7:00AM - 9:50AM in Room 828 **Materials** • Test (blackboard) **CCC Confer** • Enter virtual classroom • Class archives		5 posts Lab X1 Lab X2

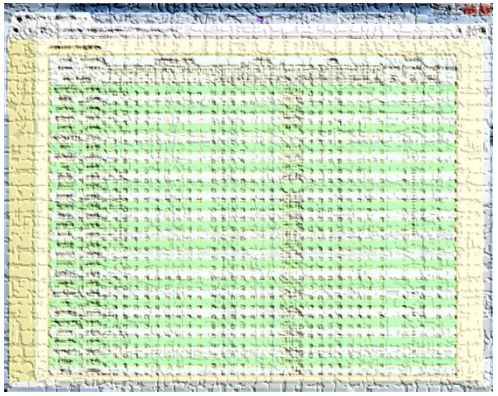
Extra credit labs and final posts due by 11:59PM

- All students will take the test at the same time. The test must be completed by 9:50AM.
- Working and long distance students can take the test online via CCC Confer and BlackBoard.
- Working students will need to plan ahead to take time off from work for the test.

Monitoring your grades

Points that could have been earned:	
10 quizzes:	30 points
10 labs:	300 points
2 tests:	60 points
3 forum quarters:	60 points
Total:	**450 points**

The CIS 90 website



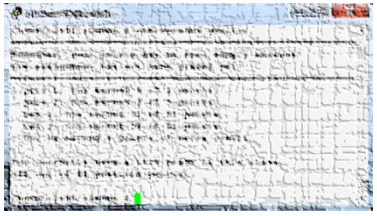
http://simms-teach.com/cis90grades.php

Send me your survey to get your LOR code name.

Or on Opus

`checkgrades` *`codename`*

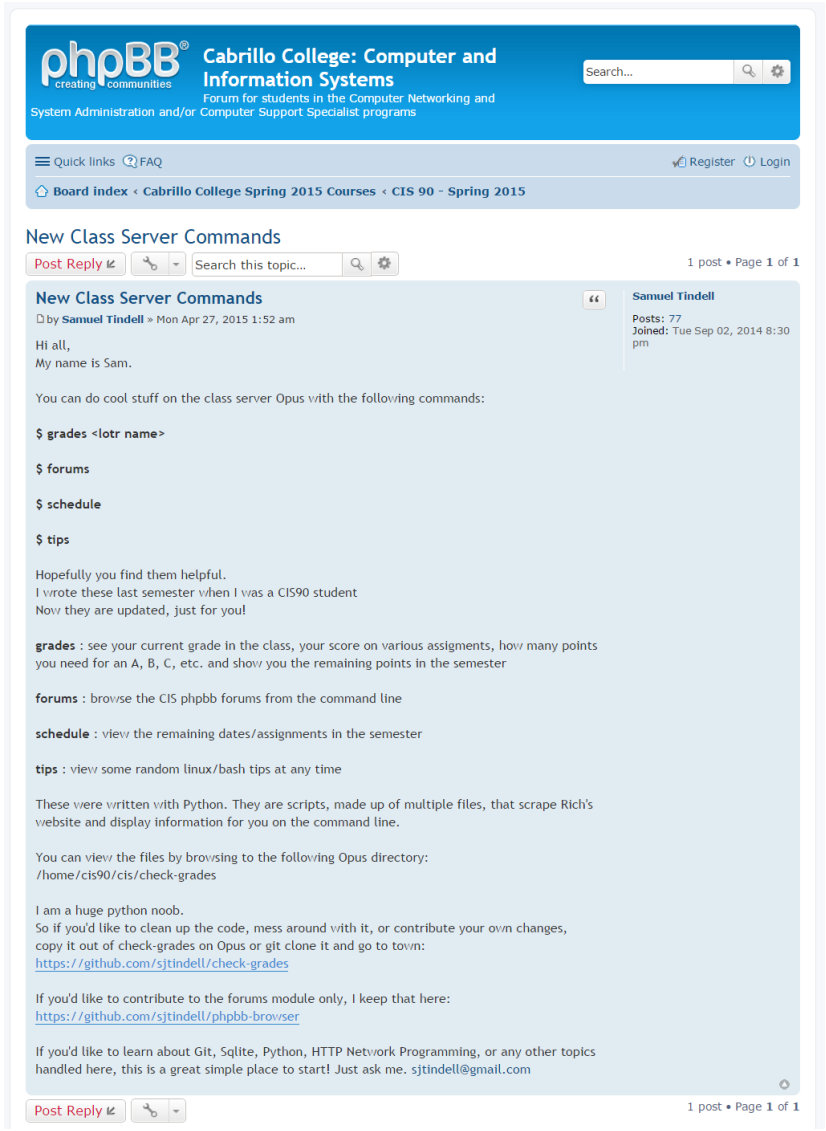
(where codename is your LOR codename)



The checkgrades script was written by Jessie a past CIS 90 Alumnus

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

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



phpBB creating communities

Cabrillo College: Computer and Information Systems

Forum for students in the Computer Networking and System Administration and/or Computer Support Specialist programs

Search...

Quick links FAQ Register Login

Board index ‹ Cabrillo College Spring 2015 Courses ‹ CIS 90 - Spring 2015

New Class Server Commands

Post Reply | Search this topic... | 1 post • Page 1 of 1

New Class Server Commands

by **Samuel Tindell** » Mon Apr 27, 2015 1:52 am

Samuel Tindell
Posts: 77
Joined: Tue Sep 02, 2014 8:30 pm

Hi all,
My name is Sam.

You can do cool stuff on the class server Opus with the following commands:

$ grades <lotr name>

$ forums

$ schedule

$ tips

Hopefully you find them helpful.
I wrote these last semester when I was a CIS90 student
Now they are updated, just for you!

grades : see your current grade in the class, your score on various assigments, how many points you need for an A, B, C, etc. and show you the remaining points in the semester

forums : browse the CIS phpbb forums from the command line

schedule : view the remaining dates/assignments in the semester

tips : view some random linux/bash tips at any time

These were written with Python. They are scripts, made up of multiple files, that scrape Rich's website and display information for you on the command line.

You can view the files by browsing to the following Opus directory:
/home/cis90/cis/check-grades

I am a huge python noob.
So if you'd like to clean up the code, mess around with it, or contribute your own changes, copy it out of check-grades on Opus or git clone it and go to town:
https://github.com/sjtindell/check-grades

If you'd like to contribute to the forums module only, I keep that here:
https://github.com/sjtindell/phpbb-browser

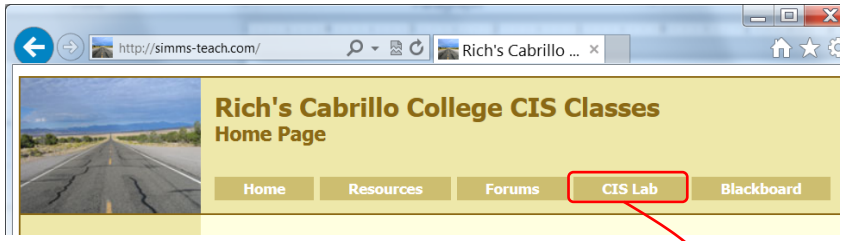
If you'd like to learn about Git, Sqlite, Python, HTTP Network Programming, or any other topics handled here, this is a great simple place to start! Just ask me. sjtindell@gmail.com

Post Reply | 1 post • Page 1 of 1

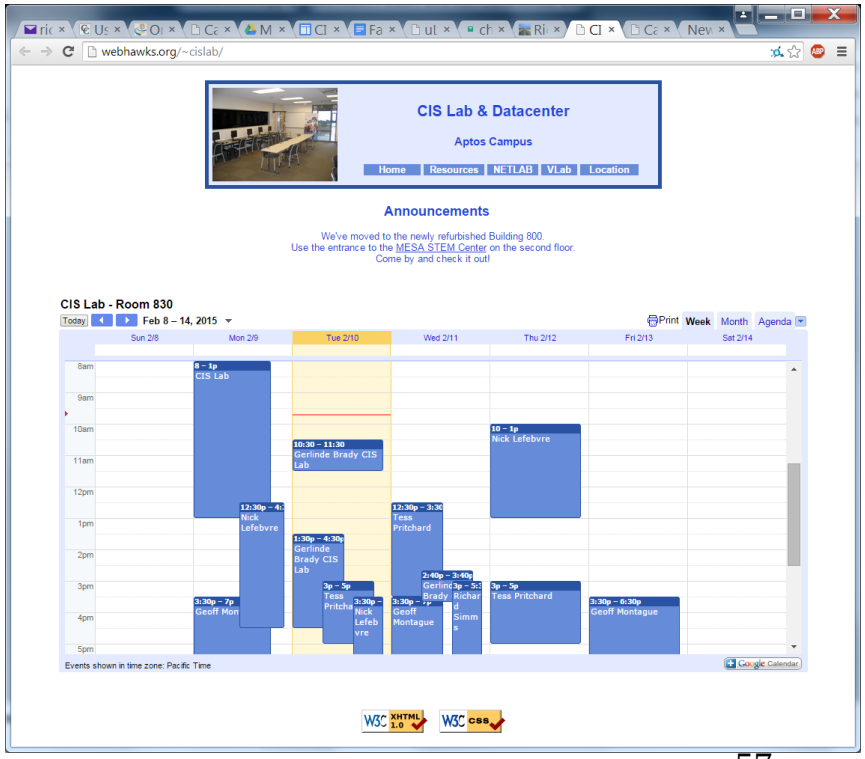
Sams' commands are available again:

grades *<LOR code name>*
forums
schedule
tips

Want some help working the labs?



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



Nick, Sean, and Geoff are CIS 90 Alumni.

Tess and Mike F. are in our class!

Michael M. is the other Linux instructor.

I'm in there Wednesdays 3:00-5:30pm.

CIS 90 Tutoring Available

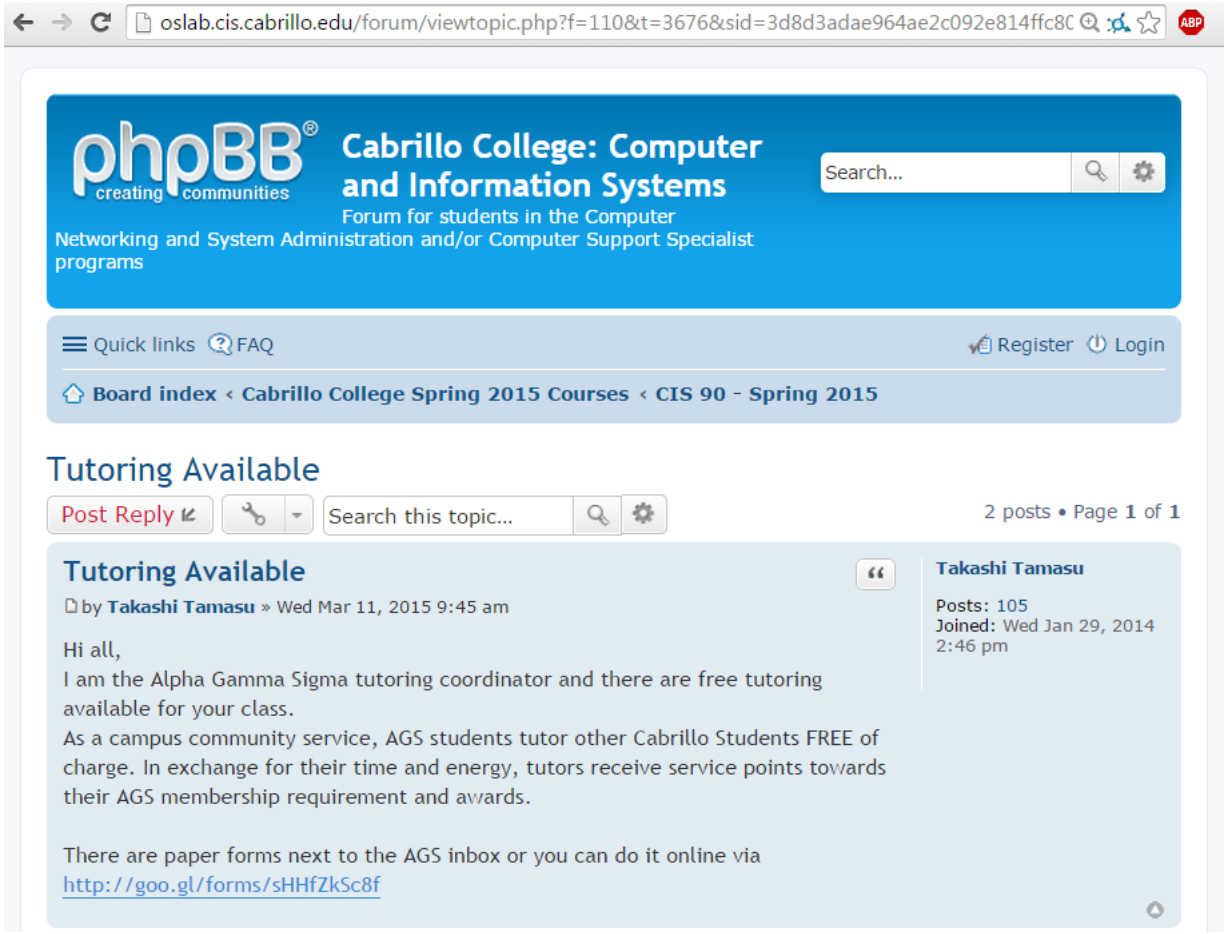
http://www.cabrillo.edu/services/tutorials/

Geoffrey Montague

All students interested in tutoring need to come directly to the Tutorials Center to schedule, register and fill out some paperwork. This is just a one-time visit.

The tutoring will take place at the STEM center.

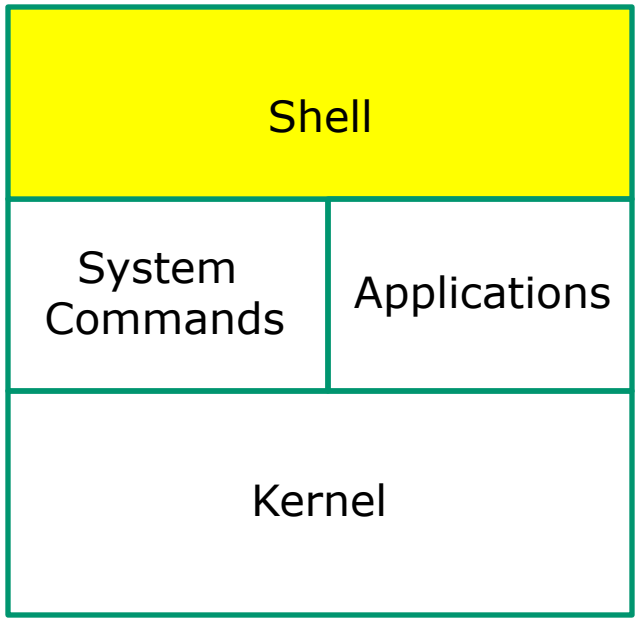
More CIS 90 Tutoring Available



Refresh

UNIX/Linux Architecture

The Shell



- Allows users to interact with the computer via a "command line".
- Prompts for a command, parses the command, finds the right program and gets that program executed.
- Called a "shell" because it hides the underlying operating system.
- Many shell programs are available: sh (Bourne shell), bash (Bourne Again shell), csh (C shell), ksh (Korn shell).
- **A user interface and a programming language (scripts).**
- GNOME and KDE desktops could be called graphical shells

Shell Scripts

Some scripts on opus

1) /home/cis90/bin/riddle1
2) /home/cis90/bin/allscripts
3) /etc/rc.d/init.d/network
4) /usr/bin/spell
5) /usr/bin/vimtutor
6) ~/bin/enlightenment

You have read permission for all these scripts. You can use cat, more, less, or even vi to view them

Many commands are scripts

Which commands in /bin are really scripts?

```
file /bin/* | grep script
```

How many commands in /bin are scripts?

```
file /bin/* | grep script | wc -l
```

Class Activity
Scripting

Of all the UNIX/Linux commands in:

/bin

/usr/bin

/sbin

/usr/sbin

How many are scripts?

Write your answer in the chat window

Project

Get started on the project!
(If you haven't already)

1. Create a file in your bin directory named *myscript*:
   - Copy from */home/cis90/depot/myscript*
   - or copy and paste template code from:
     http://simms-teach.com/docs/cis90/cis90final-project.pdf

2. Give yourself full permissions and give CIS 90 group read and execute permissions
   - **chmod 750 myscript**

3. Run **allscripts** and verify your script will run without any errors

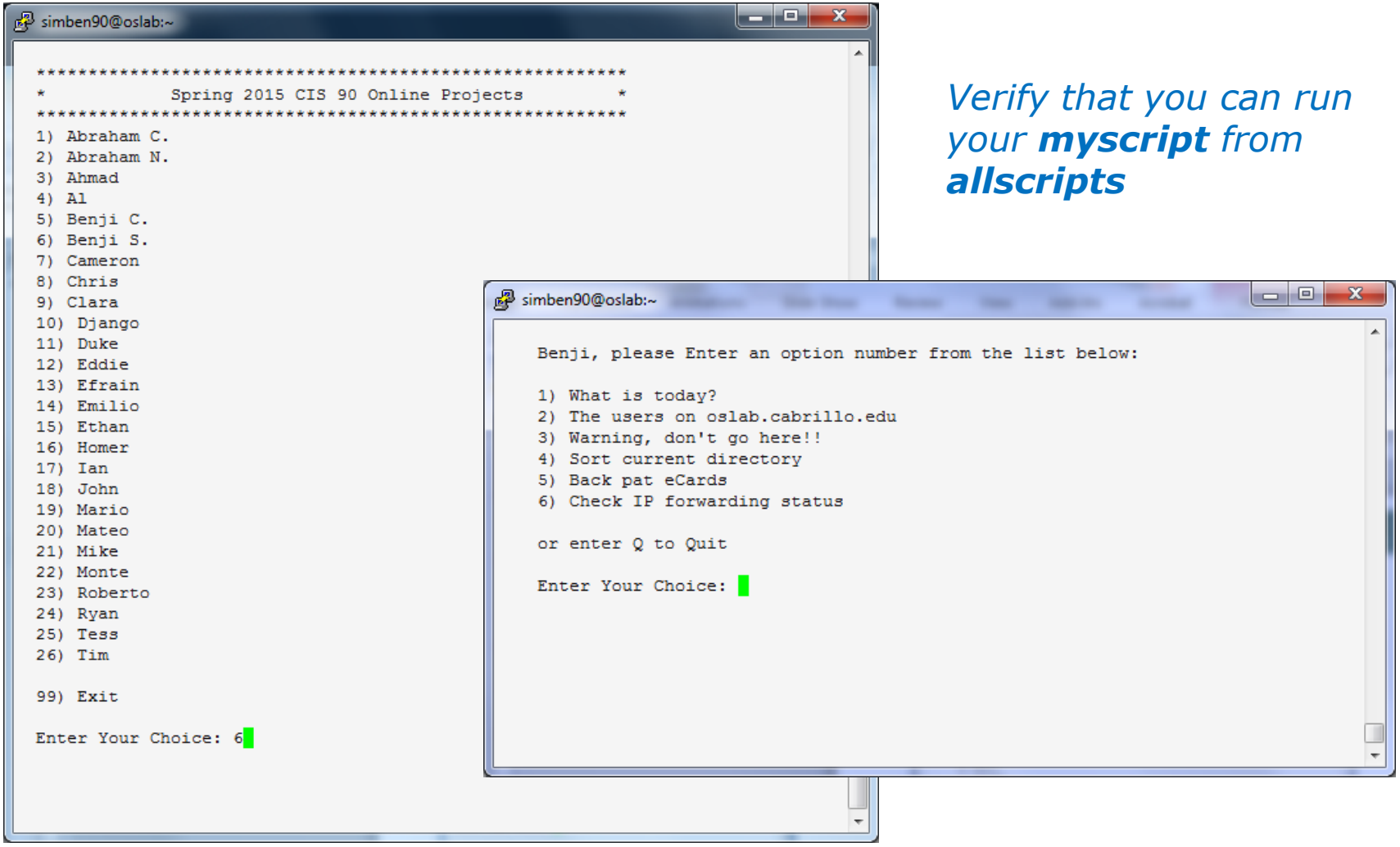
4. Do the example grep task shown in Lesson 13

Grading rubric (60 points maximum)

Possible Points	Requirements
30	Implementing all five tasks (6 points each): • Requirements for each task: – Minimum of 10 "original" script command lines – Has one or more non-generic comments to explain what it is doing – Has user interaction
25	You don't have to do all of these but do at least five: • Redirecting stdin (5 points) • Redirecting stdout (5 points) • Redirecting stderr (5 points) • Use of permissions (5 points) • Use of filename expansion characters (5 points) • Use of absolute path (5 points) • Use of relative path (5 points) • Use of a PID (5 points) • Use of inodes (5 points) • Use of links (5 points) • Use of scheduling (5 points) • Use of a GID or group (5 points) • Use of a UID or user (5 points) • Use of a /dev/tty device (5 points) • Use of a signal (5 points) • Use of piping (5 points) • Use of an environment variable (5 points) • Use of /bin/mail (5 points) • Use of a conditional (5 points) The maximum for this section is 25 points.
5	Present your script to the class
Points lost	
-15	Fails to run from **allscripts**
-15	Other students in the class are unable to read and execute your script.
-15	Error messages are displayed when running one or more tasks
-up to 90	No credit for any task which contains unoriginal script code that: • Doesn't give full credit to the original author • Doesn't indicate where the code was obtained from • Doesn't include licensing terms • Violates copyright or licensing terms
Extra credit	
30	Up to three additional tasks (10 points each)

This applies to each individual task

This applies to the project as a whole



*Verify that you can run your **myscript** from **allscripts***

Don't forget to do this!

chmod 750 ~/bin/myscript

Points lost	
-15	Fails to run from **allscripts**
-15	Other students in the class are unable to read and execute your script.
-15	Error messages are displayed when running one or more tasks
-up to 90	No credit for any task which contains unoriginal script code that: • Doesn't give full credit to the original author • Doesn't indicate where the code was obtained from • Doesn't include licensing terms • Violates copyright or licensing terms

Project Status

```
ls -l /home/cis90/*/bin/myscript
```

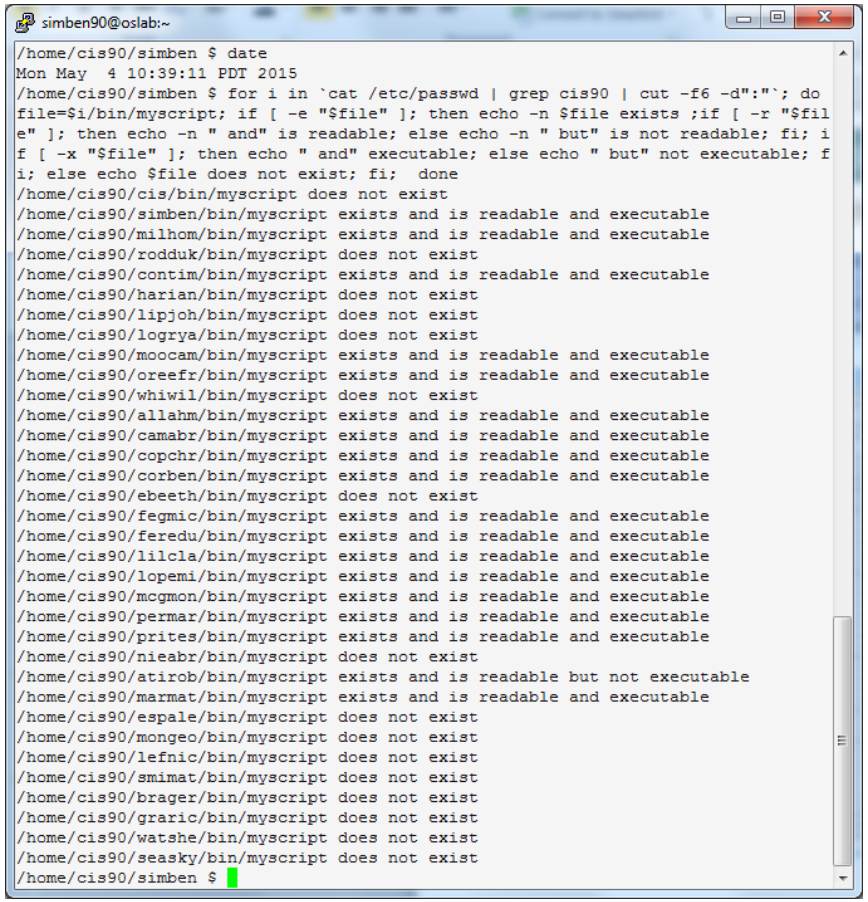
simben90@oslab:~

```
/home/cis90/simben $ date
Mon May  4 10:37:44 PDT 2015
/home/cis90/simben $ ls -l /home/cis90/*/bin/myscript
-rwxr-x---. 1 allahm90 cis90  3449 May  3 19:31 /home/cis90/allahm/bin/myscript
-rwxrw----. 1 atirob90 cis90   546 Apr 29 00:08 /home/cis90/atirob/bin/myscript
-rwxr-x---. 1 camabr90 cis90   716 Apr 22 10:56 /home/cis90/camabr/bin/myscript
-rwxr-x---. 1 contim90 cis90  1215 Apr 29 11:08 /home/cis90/contim/bin/myscript
-rwxr-xr-x. 1 copchr90 cis90  1285 Apr 22 10:38 /home/cis90/copchr/bin/myscript
-rwxr-x---. 1 corben90 cis90  1376 Apr 29 12:28 /home/cis90/corben/bin/myscript
-rwxr-x---. 1 fegmic90 cis90  5236 May  3 20:15 /home/cis90/fegmic/bin/myscript
-rwxr-x---. 1 feredu90 cis90   886 Apr 30 21:58 /home/cis90/feredu/bin/myscript
-rwxr-x---. 1 lilcla90 cis90   784 Apr 22 10:37 /home/cis90/lilcla/bin/myscript
-rwxrwxr-x. 1 lopemi90 cis90   710 Apr 22 10:36 /home/cis90/lopemi/bin/myscript
-rwxr-x---. 1 marmat90 cis90   698 Apr 22 10:44 /home/cis90/marmat/bin/myscript
-rwxrwxr-x. 1 mcgmon90 cis90   737 Apr 22 10:38 /home/cis90/mcgmon/bin/myscript
-rwxr-x---. 1 milhom90 cis90  4729 Apr 21 09:19 /home/cis90/milhom/bin/myscript
-rwxr-x---. 1 moocam90 cis90   748 Apr 22 10:37 /home/cis90/moocam/bin/myscript
-rwxr-x---. 1 oreefr90 cis90  1806 May  1 15:11 /home/cis90/oreefr/bin/myscript
-rwxr-x---. 1 permar90 cis90  1177 May  4 10:28 /home/cis90/permar/bin/myscript
-rwxr-x---. 1 prites90 cis90  8621 May  2 09:57 /home/cis90/prites/bin/myscript
-rwxr-x---. 1 simben90 cis90 10512 Apr 21 09:17 /home/cis90/simben/bin/myscript
/home/cis90/simben $
```

Is your script "hackable" by others classmates?

Project Status

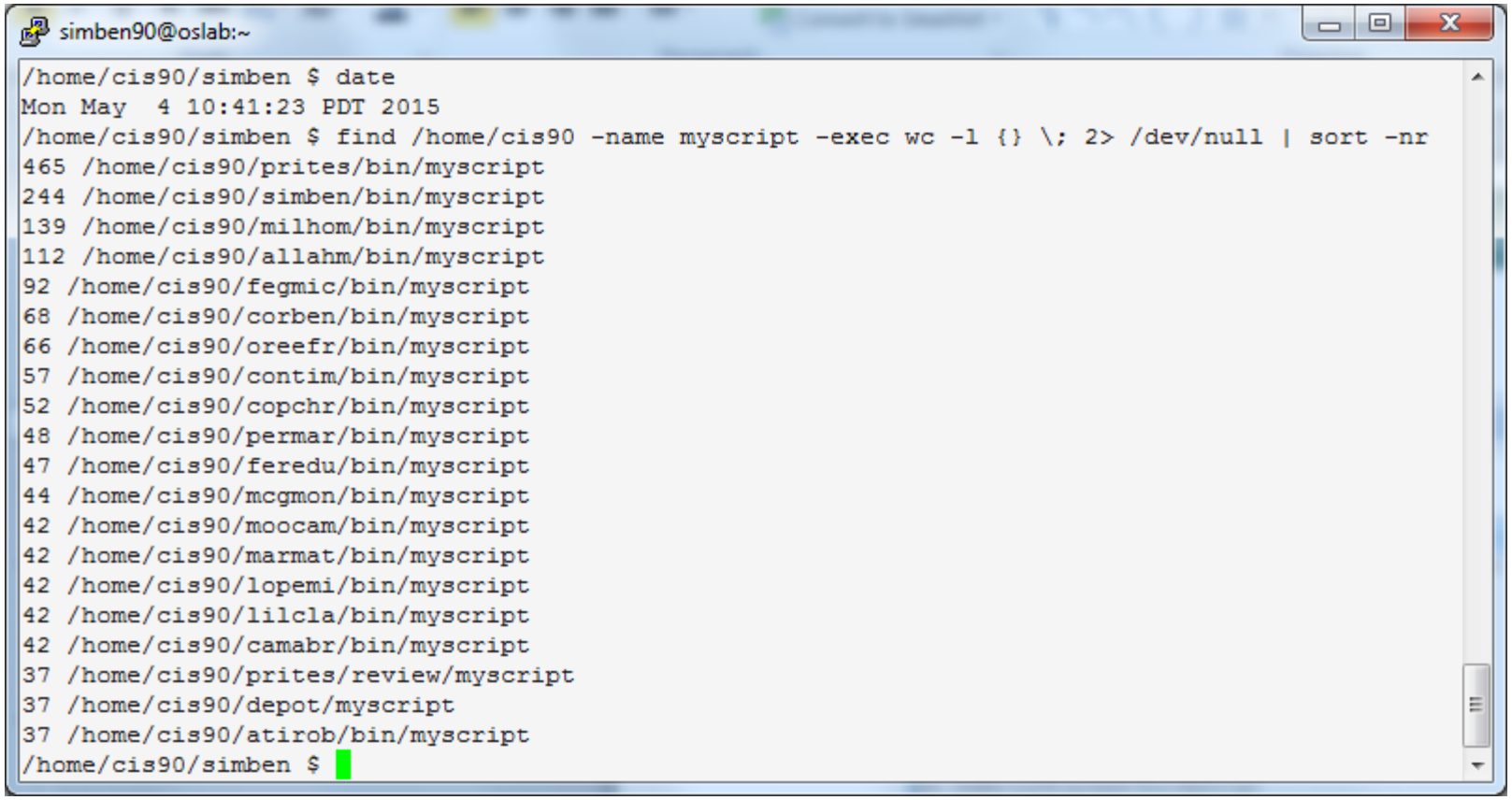
```
for i in `cat /etc/passwd | grep cis90 | cut -f6 -d":"`; do file=$i/bin/myscript; if [ -e
"$file" ]; then echo -n $file exists ;if [ -r "$file" ]; then echo -n " and" is readable;
else echo -n " but" is not readable; fi; if [ -x "$file" ]; then echo " and" executable;
else echo " but" not executable; fi; else echo $file does not exist; fi;  done
```



a one line command using semi-colons!

Project Status

```
find /home/cis90 -name myscript -exec wc -l {} \; 2> /dev/null | sort -nr
```



Review

```
function runningScript ()
{
```

The rules of the road for variables

- Rule 1: A child process can only see variables the parent has exported.
- Rule 2: A child process cannot change the parent's variables.

Running a Script

```
/home/cis90/simben $ cat mydate
#!/bin/bash
echo "Hola $LOGNAME"
date +'%m/%d/%Y'
echo $myvar1 $myvar2 $myvar3
```

Add this line to the last script we made

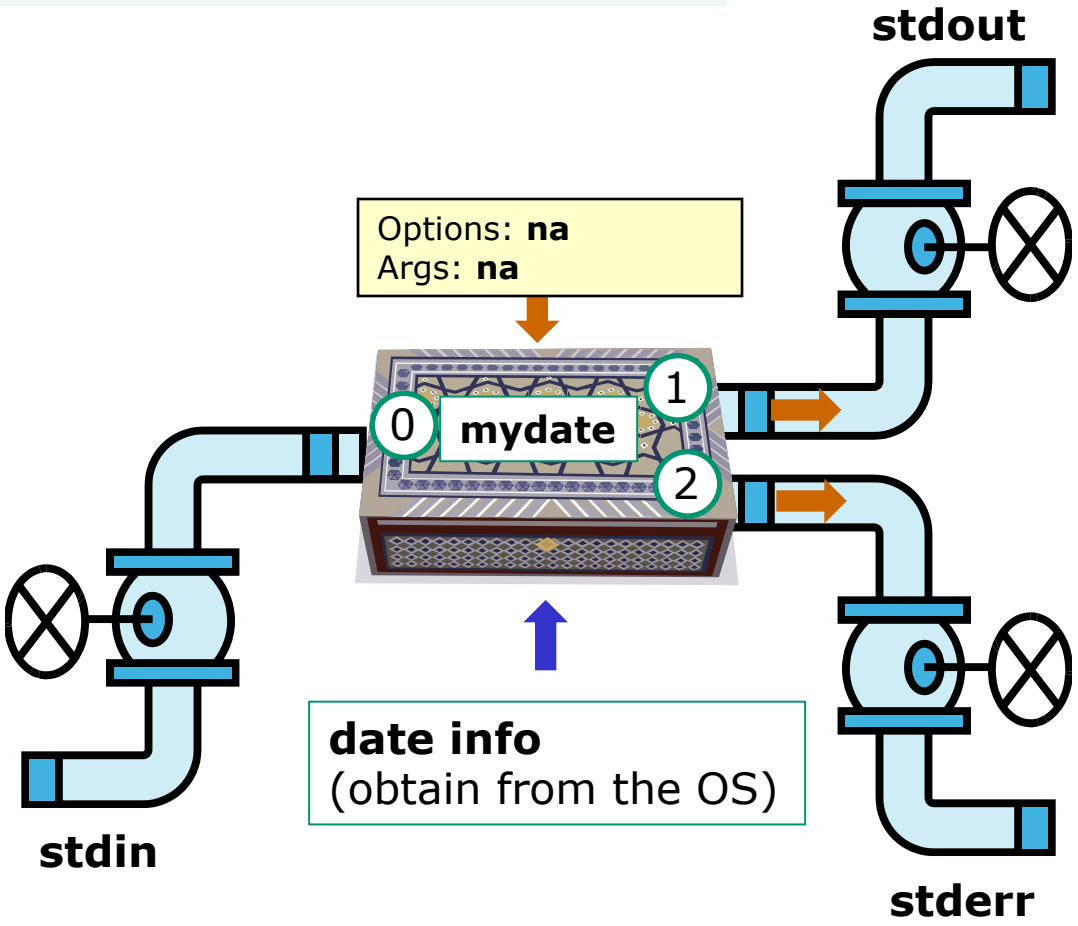
Don't initialize them yet

```
/home/cis90/simben $ mydate
Hola simben90
12/02/2014

/home/cis90/simben $
```

Because the variables don't exist yet the last echo statement prints a blank line

Running a Script

$ mydate



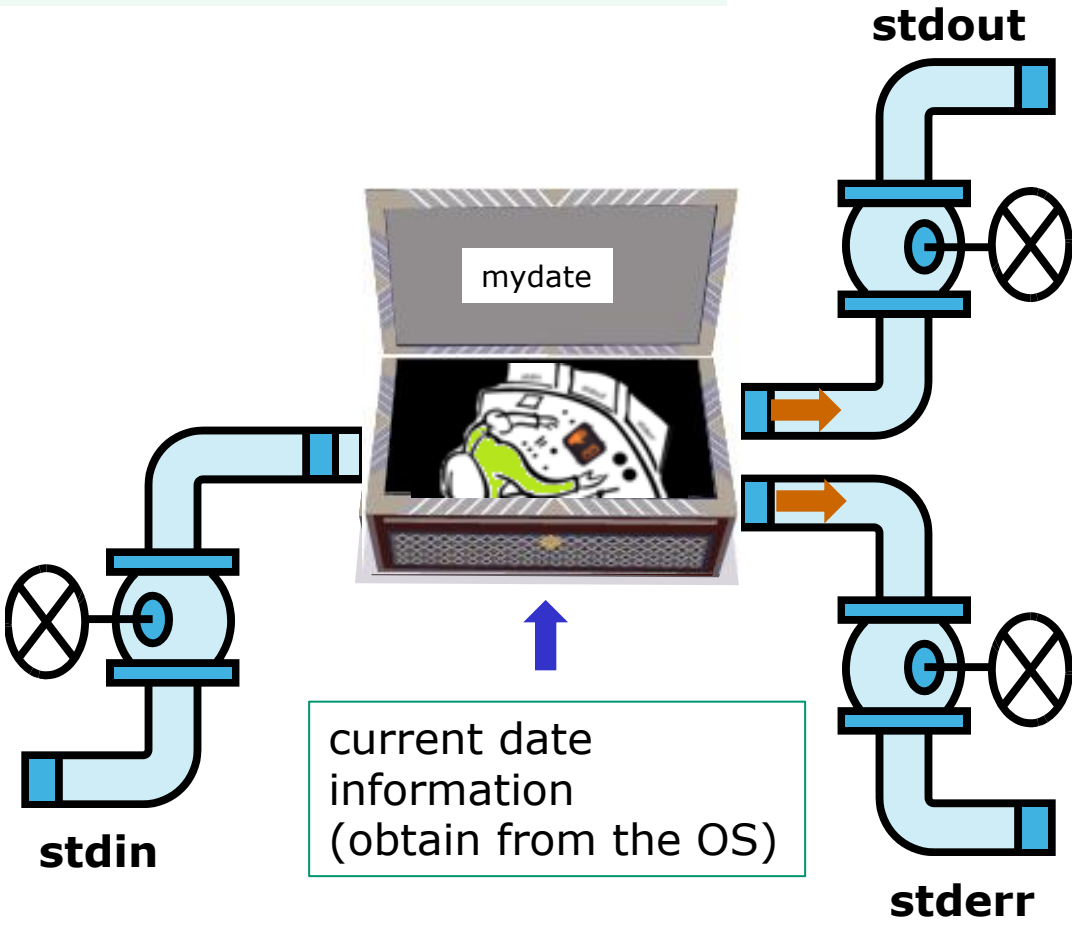

```
Hola simben90
12/02/2014
```

*In this example, ouptput from **myscript** goes to stdout.*

stdout has not been redirected so it goes to the default terminal device (your screen).

Running a Script

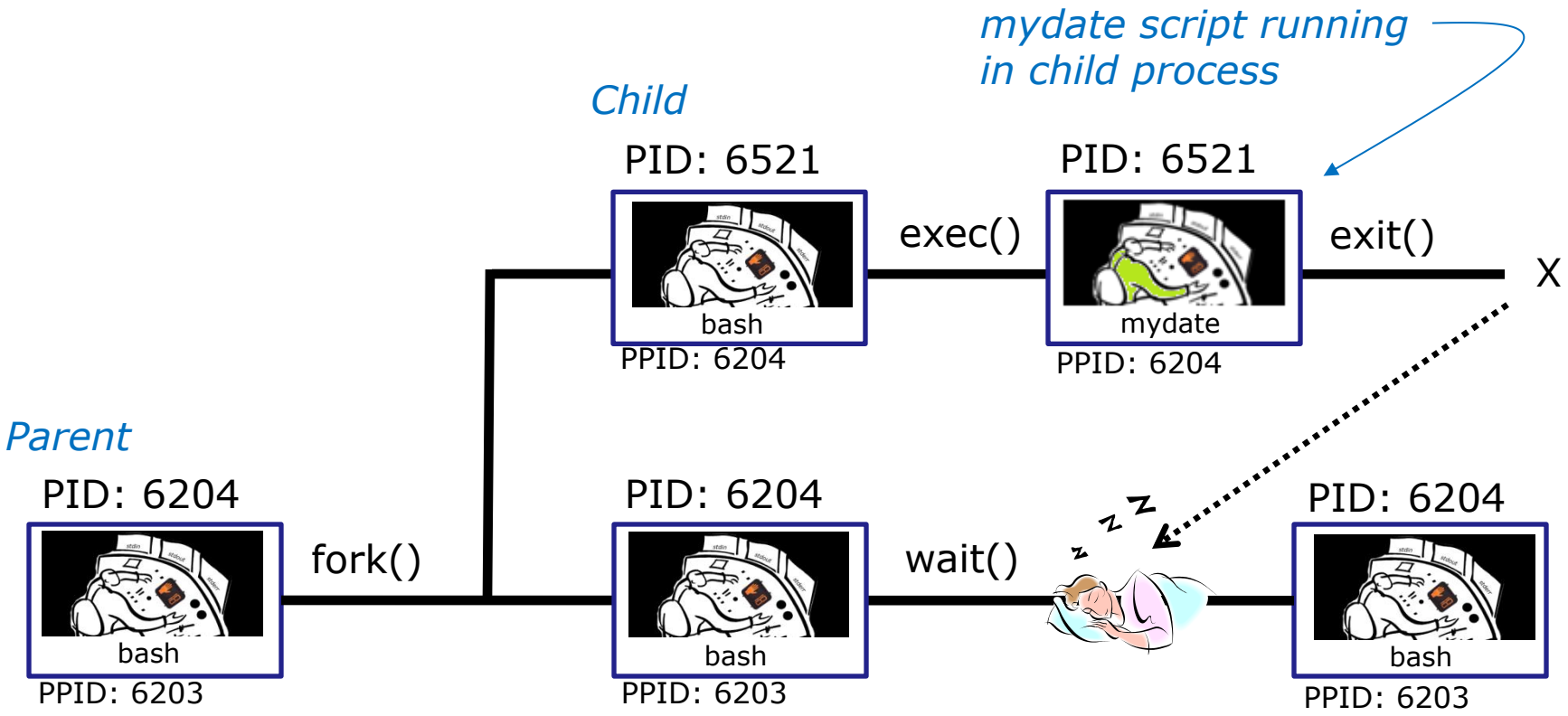
$ mydate




```
Hola simben90
12/02/2014
```

A sneak peek into memory to see what our process looks like!

Running a Script



*Whenever you run any command, program, or script it runs as a **child process***

Running a Script

```
/home/cis90/simben $ cat mydate
#!/bin/bash
echo "Hola $LOGNAME"
date +'%m/%d/%Y'
echo $myvar1 $myvar2 $myvar3
```

In the parent process, initialize the three variables

```
/home/cis90/simben $ myvar1=Tic; myvar2=Tac; myvar3=Toe
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3
Tic Tac Toe
```

*What happens if we run **mydate** now?*

Running a Script

```
/home/cis90/simben $ cat mydate
#!/bin/bash
echo "Hola $LOGNAME"
date +'%m/%d/%Y'
echo $myvar1 $myvar2 $myvar3

/home/cis90/simben $ myvar1=Tic; myvar2=Tac; myvar3=Toe
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3
Tic Tac Toe

/home/cis90/simben $ mydate
Hola simben90
12/02/2014

/home/cis90/simben $
```

*Running **mydate***
(as a child process)

Why no Tic Tac Toe output?

Running a Script

```
/home/cis90/simben $ export myvar1
/home/cis90/simben $ mydate
Hola simben90
12/02/2014
Tic

/home/cis90/simben $ export myvar2
/home/cis90/simben $ mydate
Hola simben90
12/02/2014
Tic Tac

/home/cis90/simben $ export myvar3
/home/cis90/simben $ mydate
Hola simben90
12/02/2014
Tic Tac Toe
```

Rule 1: A child process can only see variables the parent has exported

Running a Script

```
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3
Tic Tac Toe

/home/cis90/simben $ cat mydate
#!/bin/bash
echo "Hola $LOGNAME"
date +'%m/%d/%Y'
echo $myvar1 $myvar2 $myvar3
myvar1=red myvar2=white myvar3=blue
echo $myvar1 $myvar2 $myvar3

/home/cis90/simben $ mydate
Hola simben90
12/02/2014
Tic Tac Toe
red white blue

/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3
Tic Tac Toe
```

Add these new lines

Rule 2: A child process cannot change the parent's variables.

Running a Script

Unless we want them to

```
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3
Tic Tac Toe

/home/cis90/simben $ source mydate
Hola simben90
12/02/2014
Tic Tac Toe
red white blue

/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3
red white blue
```

Sourcing a script causes the instructions to be run in the parent process. A child process is not created

```
}
while no-comprende
do
  runningScript
done
```

Scripting Tips
vi

Line Numbers in errors and vi

```
milhom90@oslab:~/bin
Are you ready to search for beauty in the poems?

That thereby beauty's rose might never die,
        That beauty still may live in thine or thee.
Herein lives wisdom, beauty, and increase;
If I could write the beauty of your eyes,
And dig deep trenches in thy beauty's field,
Then being ask'd, where all thy beauty lies,
How much more praise deserv'd thy beauty's use,
Proving his beauty by succession thine.
Upon thyself thy beauty's legacy?
        Thy unus'd beauty must be tomb'd with thee,
Beauty's effect with beauty were bereft,
Yet mortal looks adore his beauty still,
But beauty's waste hath in the world an end,
And loved your beauty with love false or true,
Ready to count them?

14
Enter a new string to search for

searching for ""
./myscript: line 40: grab: command not found
Hit the Enter key to return to menu
```

```
milhom90@oslab:~/bin
        1)      # Task 1 - grep command explored

                # Simple grep for "beauty"
                echo "Are you ready to search for beauty in the poems?"
                read dummy
                grep -h beauty /home/cis90/milhom/poems/*/*

                                                tches too

                grep -h beauty /home/cis90/milhom/poems/*/* | wc -l

                # Prompt user to supply search string and use color
                echo "Enter a new string to search for"
                read string
                echo searching for '"'$string'"'
                grab -h --color $string /home/cis90/milhom/poems/*/*
                ;;
        2)      # Commands for Task 2
                ;;
        3)      # Commands for Task 3
                ;;
        4)      # Commands for Task 4
                                                        40,17         38%
```

line 40, column 17

Use the line number in error messages to locate the error in you script

Color Syntax

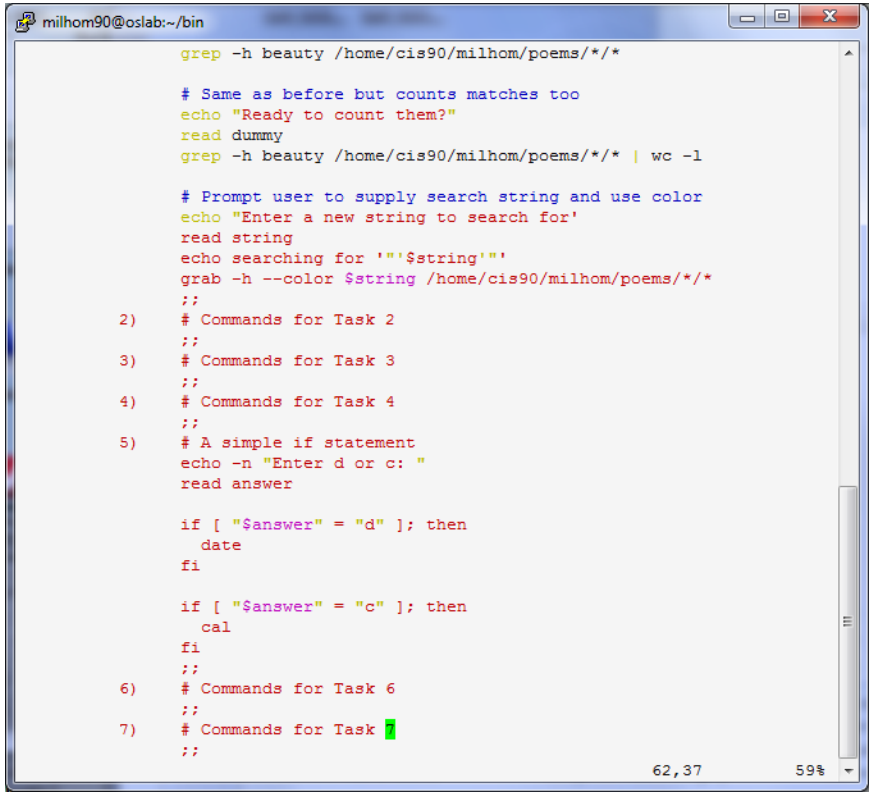
```
milhom90@oslab:~/bin
/home/cis90/milhom/bin $ ./myscript
./myscript: line 79: unexpected EOF while looking for matching `"'
./myscript: line 83: syntax error: unexpected end of file
/home/cis90/milhom/bin $
```

```
milhom90@oslab:~/bin
            grep -h beauty /home/cis90/milhom/poems/*/*

            # Same as before but counts matches too
            echo "Ready to count them?"
            read dummy
            grep -h beauty /home/cis90/milhom/poems/*/* | wc -l

            # Prompt user to supply search string and use color
            echo "Enter a new string to search for'
            read string
            echo searching for '"'$string'"'
            grab -h --color $string /home/cis90/milhom/poems/*/*
            ;;
        2)  # Commands for Task 2
            ;;
        3)  # Commands for Task 3
            ;;
        4)  # Commands for Task 4
            ;;
        5)  # A simple if statement
            echo -n "Enter d or c: "
            read answer

            if [ "$answer" = "d" ]; then
              date
            fi

            if [ "$answer" = "c" ]; then
              cal
            fi
            ;;
        6)  # Commands for Task 6
            ;;
        7)  # Commands for Task 7
            ;;
                                                    62,37         59%
```

Use color syntax to spot unmatched quotes

Is there a problem with this script? Where exactly is the problem?

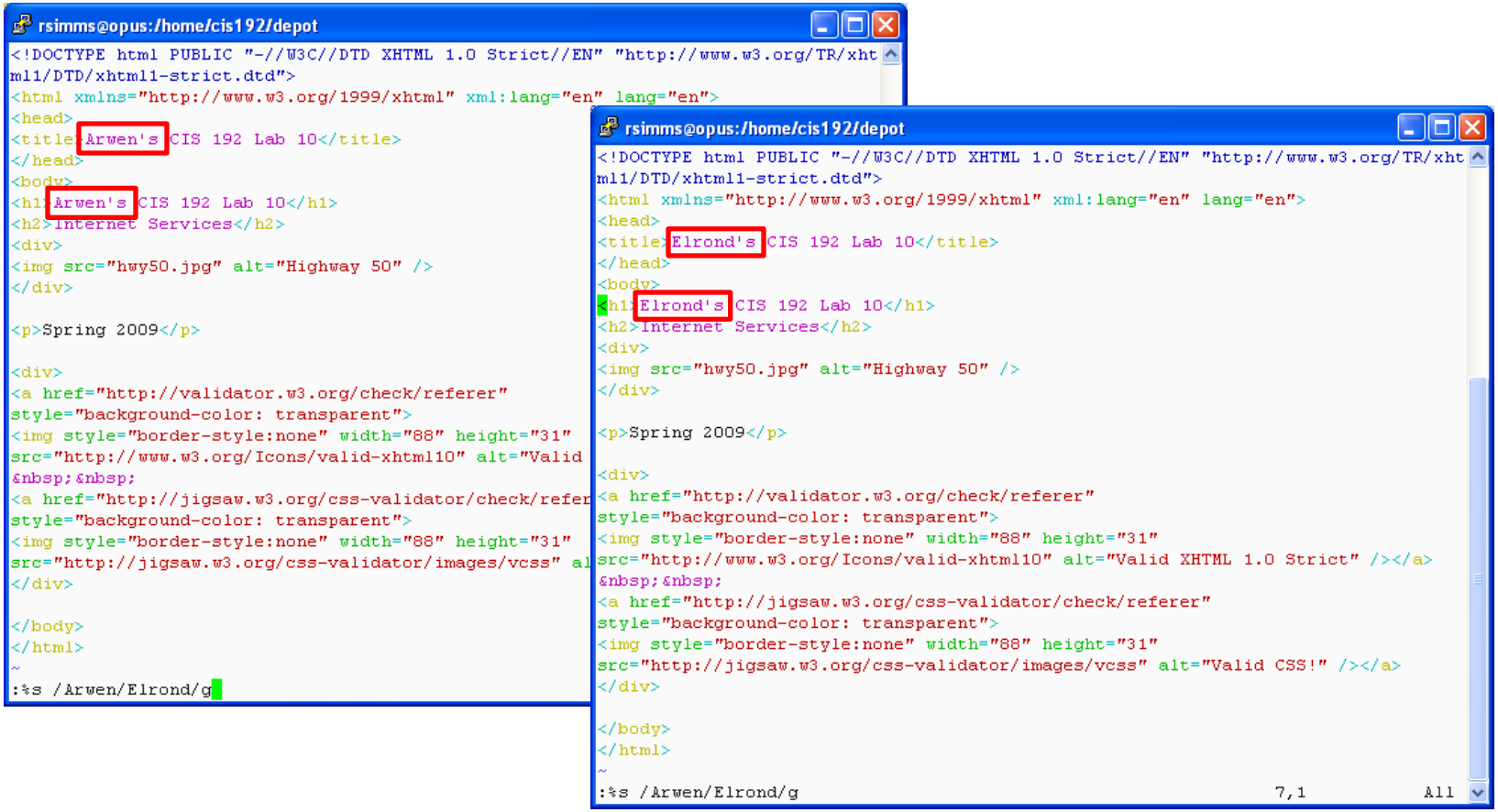
Color Syntax



One small change for script developer, one giant leap for script execution

Global search and replace with vi

<esc>:
%s /oldstring/newstring/g



Scripting Tips

sleep

(adding timed pauses)

Class Exercise

Make a new script in your bin directory

```
cd bin
vi example911
```

In vi add these lines then save:

```
echo help
sleep 3
echo Help
sleep 2
echo HELP
sleep 1
banner HELP
```

Prepare and run your script

```
chmod +x example911
example911
```

Scripting Tips

$(cmd) and `cmd`

Using $(cmd)

Sometimes you want to capture the output of a command and store in a variable or use as an argument

For example:

```
/home/cis90/simben $ find /bin | wc -l
113

/home/cis90/simben $ count=`find /bin | wc -l`

/home/cis90/simben $ echo "There are $count files in /bin"
There are 113 files in /bin
```

Using back tics around the command to evaluate

Using back tics

Sometimes you want to capture the output of a command and store in a variable or use as an argument

For example:

```
/home/cis90/simben $ find /bin | wc -l
113

/home/cis90/simben $ count=$(find /bin | wc -l)

/home/cis90/simben $ echo "There are $count files in /bin"
There are 113 files in /bin
```

Using $() instead of back tics is an alternate way to do the same thing

Class Activity
Scripting

```
/home/cis90/milhom/bin $ date +"%A"
Sunday
```

Which of the following commands makes a banner of the current day of the week?

a) **`date +"%A" | banner`**

b) **`banner date +"%A"`**

c) **`banner `date +"%A"``**

d) **`banner $(date +"%A")`**

e) **`date +"%A" | xargs banner`**

Put your answer in the chat window

Scripting Tips

extracting a field from a record

/etc/passwd

```
[rsimms@opus ~]$ cat /etc/passwd
< snipped >
apache:x:48:48:Apache:/var/www:/sbin/nologin
simben90:x:1001:190:Benji Simms:/home/cis90/simben:/bin/bash
milhom90:x:1002:190:Homer Miller:/home/cis90/milhom:/bin/bash

< snipped >
```

*The ":" serves as the field **delimiter***

The 5th field of each row has the user's first and last name

myscript

```
        8)     # Commands for Task 8
               date
               ;;
```

Let's start with something simple like printing the current date and time

```
                Homer's CIS 90 Final Project
        1) Color
        2) My Find Command
        3) More practice
        4) Examples - test file attributes
        5) Examples - simple if statement
        6) Examples - another if statement
        7) Examples - logic
        8) Examples - cut command to get name from /etc/passwd
        9) Exit

        Enter Your Choice: 8
Wed Dec  3 14:00:53 PST 2008
Hit the Enter key to return to menu
```

myscript

```
8)      # Commands for Task 8
        echo "Hello $LOGNAME"
        date
        ;;
```

Let's add a friendly Hello using the user logname

```
                Homer's CIS 90 Final Project
        1) Color
        2) My Find Command
        3) More practice
        4) Examples - test file attributes
        5) Examples - simple if statement
        6) Examples - another if statement
        7) Examples - logic
        8) Examples - cut command to get name from /etc/passwd
        9) Exit

        Enter Your Choice: 8
Hello milhom90
Wed Dec  3 14:07:07 PST 2008
Hit the Enter key to return to menu
```

myscript

```
8)      # Commands for Task 8
        echo "Hello $LOGNAME"
        echo $(cat /etc/passwd | grep $LOGNAME)
        date
        ;;
```

Now include the /etc/passwd info as well

```
                Homer's CIS 90 Final Project
        1) Color
        2) My Find Command
        3) More practice
        4) Examples - test file attributes
        5) Examples - simple if statement
        6) Examples - another if statement
        7) Examples - logic
        8) Examples - cut command to get name from /etc/passwd
        9) Exit

        Enter Your Choice: 8
Hello milhom90
milhom90:x:1156:103:Homer Miller:/home/cis90/milhom:/bin/bash
Wed Dec  3 14:07:07 PST 2008
Hit the Enter key to return to menu
```

myscript

```
8)      # Commands for Task 8
        echo "Hello $LOGNAME"
        echo $(cat /etc/passwd | grep $LOGNAME | cut -f5 -d":" )
        date
        ;;
```

*Cut the 5th field from the /etc/passwd record. The **-d** option specifies the delimiter to use.*

```
                Homer's CIS 90 Final Project
        1) Color
        2) My Find Command
        3) More practice
        4) Examples - test file attributes
        5) Examples - simple if statement
        6) Examples - another if statement
        7) Examples - logic
        8) Examples - cut command to get name from /etc/passwd
        9) Exit

        Enter Your Choice: 8
Hello milhom90
Homer Miller
Wed Dec  3 14:07:07 PST 2008
Hit the Enter key to return to menu
```

myscript

```
8)      # Commands for Task 8
        echo "Hello $LOGNAME"
        NAME=$(cat /etc/passwd | grep $LOGNAME | cut -f5 -d":" )
        echo "Hello $NAME"
        date
        ;;
```

Same as before, but save the user's name in a variable and then use it

```
                Homer's CIS 90 Final Project
        1) Color
        2) My Find Command
        3) More practice
        4) Examples - test file attributes
        5) Examples - simple if statement
        6) Examples - another if statement
        7) Examples - logic
        8) Examples - cut command to get name from /etc/passwd
        9) Exit

        Enter Your Choice: 8
Hello milhom90
Hello Homer Miller
Wed Dec  3 14:07:07 PST 2008
Hit the Enter key to return to menu
```

myscript

```
8)      # Commands for Task 8
        echo "Hello $LOGNAME"
        NAME=$(cat /etc/passwd | grep $LOGNAME | cut -f5 -d":" )
        echo "Hello $NAME"
        date
        ;;
```

Get rid of the old Hello $LOGNAME since we have something better now

```
                Homer's CIS 90 Final Project
        1) Color
        2) My Find Command
        3) More practice
        4) Examples - test file attributes
        5) Examples - simple if statement
        6) Examples - another if statement
        7) Examples - logic
        8) Examples - cut command to get name from /etc/passwd
        9) Exit

        Enter Your Choice: 8
```

Hello Homer Miller

```
Wed Dec  3 14:07:07 PST 2008
Hit the Enter key to return to menu
```

myscript

```
8)      # Commands for Task 8
        NAME=$(cat /etc/passwd | grep $LOGNAME | cut -f5 -d":" | cut -f1 -d" ")
        echo "Hello $NAME"
        date
        ;;
```

We can also cut out just the first name using a blank as the delimiter

```
                  Homer's CIS 90 Final Project
   1) Color
          2) My Find Command
          3) More practice
          4) Examples - test file attributes
          5) Examples - simple if statement
          6) Examples - another if statement
          7) Examples - logic
          8) Examples - cut command to get name from /etc/passwd
          9) Exit

          Enter Your Choice: 8
Hello Homer
Wed Dec  3 14:07:07 PST 2008
Hit the Enter key to return to menu
```

Class Exercise

Make a short script named example401 that emails a banner of your full name to yourself:

Make a new script in your bin directory

```
cd bin
vi example401
```

In vi add these lines then save:

```
name=$(cat /etc/passwd | grep $LOGNAME | cut -f5 -d":" )
banner $(echo $name) | mail -s "$name" $LOGNAME
```

Prepare and run your script

```
chmod +x example401
example401
```

Read your mail to view your new message

```
mail
```

Scripting Tips

simple if statement

myscript

If statements are used to test if a condition is true and if so execute a specific set of commands

```
5)      # Simple if statement
        echo -n "Enter d or c: "
        read answer

        if [ "$answer" = "d" ]; then
           date
        fi

        if [ "$answer" = "c" ]; then
           cal
        fi

        ;;
```

*The **date** command is executed only if the user typed a "d"*

*The **cal** command is executed only if the user typed a "c"*

*An **if** statement is ended with **fi** (if spelled backward)*

myscript

```
          Homer's CIS 90 Final Project
     1) My favorite color
     2) Getting started using grep command
     3) Task 3
     4) Task 4
     5) Simple if statement
     6) Task 6
     7) Task 7
     8) Getting your name
     9) Exit

     Enter Your Choice: 5
Enter d or c: d
Tue Dec  2 09:22:39 PST 2014
Hit the Enter key to return to menu
```

```
if [ "$answer" = "d" ]; then
   date
fi
```

*The **date** command runs because $answer = d*

myscript

```
          Homer's CIS 90 Final Project
     1) My favorite color
     2) Getting started using grep command
     3) Task 3
     4) Task 4
     5) Simple if statement
     6) Task 6
     7) Task 7
     8) Getting your name
     9) Exit

    Enter Your Choice: 5
Enter d or c: c
    December 2014
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

Hit the Enter key to return to menu
```

```
if [ "$answer" = "c" ]; then
   cal
fi
```

*The **cal** command runs because $answer = c*

Class Exercise

Run the previous example task

- run **allscripts**
- select Homer's script
- select Task **5** and enter **d** (for date)
- select Task **5** and enter **c** (for calendar)

Now look at Homer's code to see how it was done:

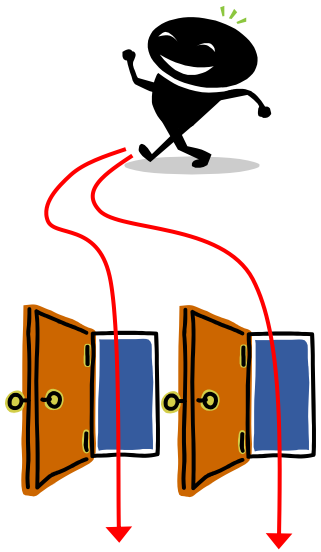
```
vi ~milhom90/bin/myscript
```

Scripting Tips

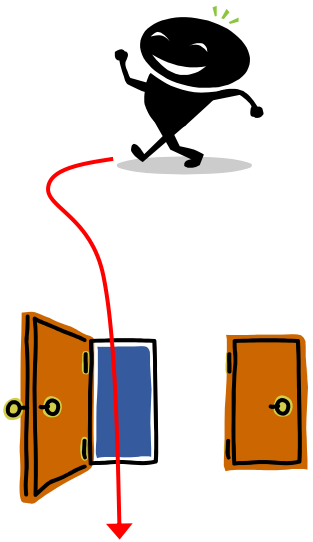
if statement with "or"

p	q	p or q
T	T	T
T	F	T
F	T	T
F	F	F

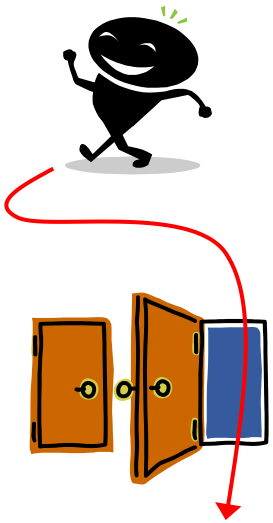
OR logic



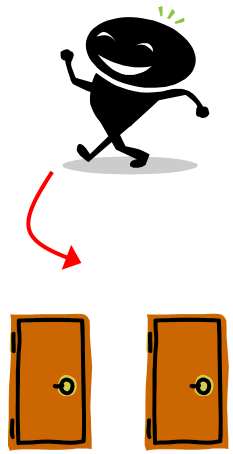
Yes



Yes



Yes



No

myscript

```
6)      # Another if statement
        echo -n "Enter d or c: "
        read answer

        if [ "$answer" = "d" ] || [ "$answer" = "D" ]; then
              date
        fi

        if [ "$answer" = "c" ] || [ "$answer" = "C" ]; then
              cal
        fi

        ;;
```

*Run **date** if the user types d or D*

*Run **cal** if the user types c or C*

*The **||** is the logical "or" operator*

myscript

```
          Homer's CIS 90 Final Project
     1) My favorite color
     2) Getting started using grep command
     3) Task 3
     4) Task 4
     5) Simple if statement
     6) Another if statement
     7) Task 7
     8) Getting your name
     9) Exit

     Enter Your Choice: 6
Enter d or c: d
Wed May 20 05:07:10 PDT 2009
Hit the Enter key to return to menu
```

```
if [ "$answer" = "d" ] || [ "$answer" =  "D" ]
then
      date
fi
```

***date** is run because user typed a "d"*

myscript

```
          Homer's CIS 90 Final Project
     1) My favorite color
     2) Getting started using grep command
     3) Task 3
     4) Task 4
     5) Simple if statement
     6) Another if statement
     7) Task 7
     8) Getting your name
     9) Exit

     Enter Your Choice: 6
 Enter d or c: D
Tue Dec  2 09:31:47 PST 2014
 Hit the Enter key to return to menu
```

```
if [ "$answer" = "d" ] || [ "$answer" = "D" ]
then
      date
fi
```

***date** is run because user typed a "D"*

Class Exercise

Make a new script in your bin directory

```
cd bin
vi example654
```

In vi add these lines then save:

```
echo -n "What is your name: "
read answer
if [ "$answer" = "Sylar" ] || [ "$answer" = "sylar" ]; then
    echo "I'm out of here"
fi
```

Prepare and run your script

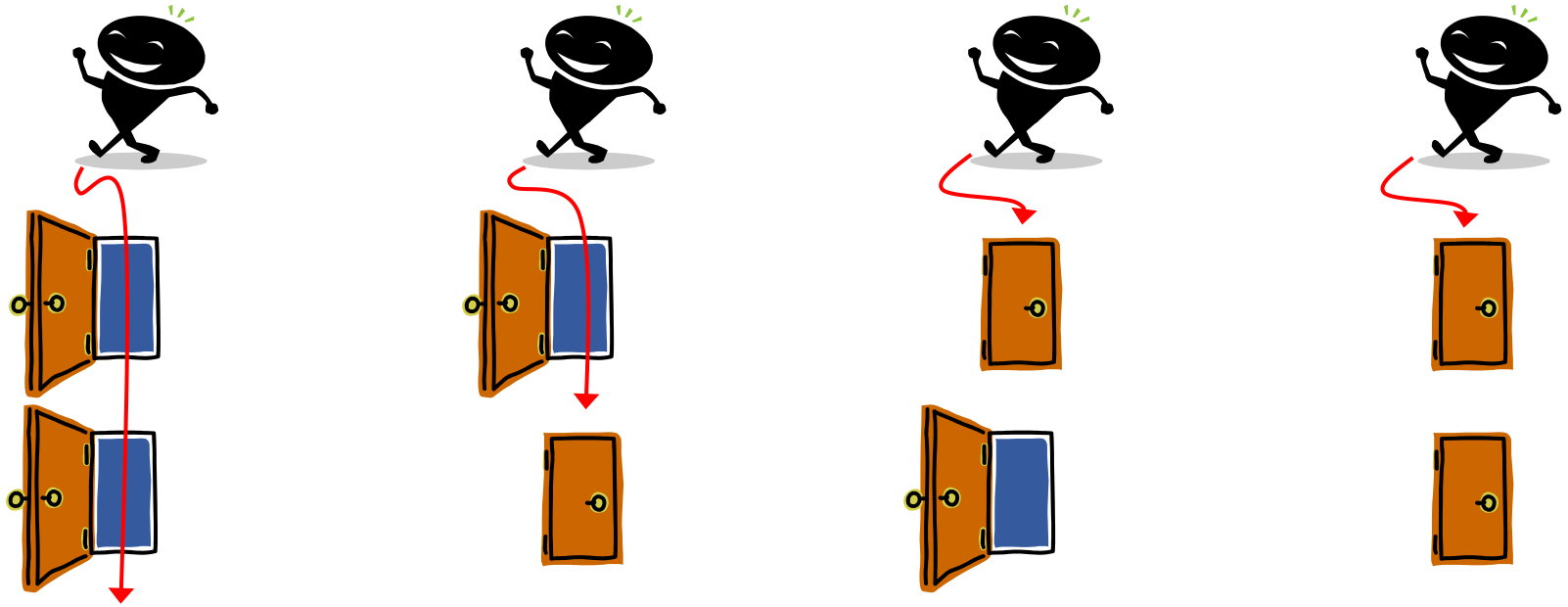
```
chmod +x example654
example654
```

Scripting Tips

if statements with "and"

p	q	p and q
T	T	T
T	F	F
F	T	F
F	F	F

AND logic



myscript

```
7)      # logic example
        echo -n "Is the furnace "on" or off? "
        read furnace
        echo -n "Is there a fire in the fireplace (yes or no)? "
        read fireplace

        if [ "$furnace" = "on" ] && [ "$fireplace" = "yes" ]; then
                echo "It is really hot in here"
        fi

        if [ "$furnace" = "off" ] && [ "$fireplace" = "yes" ]; then
                echo "It is warm and smoky in here"
        fi

        if [ "$furnace" = "on" ] && [ "$fireplace" = "no" ]; then
                echo "It is warm in here"
        fi

        if [ "$furnace" = "off" ] && [ "$fireplace" = "no" ]; then
                echo "It is really freezing in here"
        fi
        ;;
```

&& *means "and"*

myscript

```
                Homer's CIS90 Final Project
        1) My favorite color
        2) Getting started using grep command
        3) Task 3
        4) Task 4
        5) Simple if statement
        6) Another if statement
        7) Logic example
        8) Getting your name
        9) Exit

        Enter Your Choice: 7
Is the furnace on or off? off
Is there a fire in the fireplace (yes or no)? no
It is really freezing in here
Hit the Enter key to return to menu
```

```
if [ "$furnace" = "off" ] && [ "$fireplace" = "no" ]; then
   echo "It is really freezing in here"
fi
```

myscript

```
                Homer's CIS90 Final Project
        1) My favorite color
        2) Getting started using grep command
        3) Task 3
        4) Task 4
        5) Simple if statement
        6) Another if statement
        7) Logic example
        8) Getting your name
        9) Exit

        Enter Your Choice: 7
Is the furnace on or off? on
Is there a fire in the fireplace (yes or no)? no
It is warm in here
Hit the Enter key to return to menu
```

```
if [ "$furnace" = "on" ] && [ "$fireplace" = "no" ]; then
   echo "It is warm in here"
fi
```

Class Exercise

Run the previous example task
- run **allscripts**
- select Homer's script
- select Task **7** several times with different answers

Now look at Homer's code to see how it was done:

```
vi /home/cis90/milhom/bin/myscript
```

Scripting Tips

if

file types

myscript

```
4)      # More example IF statements
        echo "The files in this directory are: "
        ls -1
        echo -n "Which file are you interested in? : "
        read filename

        echo "Here are some details about $filename:"
        file $filename

        if [ -f $filename ]; then
            echo $filename is a regular file
            echo "Here is long listing of the $filename" file:
            ls -l $filename
        fi

        if [ -d $filename ]; then
            echo $filename is a directory
            echo "Here is a long listing of the $filename directory:"
            ls -ld $filename
        fi
        ;;
```

tests to see if it's a regular file

tests to see if it's a directory

myscript

```
                Homer's CIS 90 Final Project
          1) My favorite color
          2) Getting started using grep command
          3) Task 3
          4) More example IF statements
          5) Simple if statement
          6) Another if statement
          7) Logic example
          8) Getting your name
          9) Exit

Enter Your Choice: 4
The files in this directory are:
app
banner
enlightenment
< snipped >
Which file are you interested in? : enlightenment
Here are some details about enlightenment:
enlightenment: POSIX shell script text executable
enlightenment is a regular file
Here is long listing of the enlightenment file:
-rwxr-xr-x. 1 milhom90 cis90 3442 Aug  6 11:52 enlightenment
Hit the Enter key to return to menu
```

a file

myscript

```
 Homer's CIS 90 Final Project
        1) My favorite color
        2) Getting started using grep command
        3) Task 3
        4) More example IF statements
        5) Simple if statement
        6) Another if statement
        7) Logic example
        8) Getting your name
        9) Exit

Enter Your Choice: 4
The files in this directory are:
< snipped>
poems
< snipped>
Which file are you interested in? : poems
Here are some details about poems:
poems: directory
poems is a directory
Here is a long listing of the poems directory:
drwxr-xr-x. 8 milhom90 cis90 4096 Oct 28 15:48 poems
Hit the Enter key to return to menu
```

a directory

Additional file attributes to test for:

`-d file` = True if the file exists and is a directory.
`-e file` = True if the file exists.
`-f file` = True if the file exists and is a regular file
`-k file` = True if the files' "sticky" bit is set.
`-L file` = True if the file exists and is a symbolic link.
`-r file` = True if the file exists and is readable.
`-s file` = True if the file exists and is not empty.
`-u file` = True if the file exists and its set-user-id bit is set.
`-w file` = True if the file exists and is writable.
`-x file` = True if the file exists and is executable.
`-O file` = True if the file exists and is owned by the effective user id.
`-G file` = True if the file exists and is owned by the effective group id.
`file1 -nt file2` = True if file1 is newer, by modification date, than file2.
`file1 -ot file2` = True if file1 is older than file2.

Source: http://www.comptechdoc.org/os/linux/programming/script/linux_pgscripttest.html

Class Exercise

Run the previous example task
- run **allscripts**
- select Homer's script
- select Task **4**

Now look at Homer's code to see how it was done:

```
vi ~milhom90/bin/myscript
```

Scripting Tips

if then else statement

myscript

```
3)      # Commands for Task 3
        NAME=$(cat /etc/passwd | grep $LOGNAME | cut -f5 -d":" )
        echo "Hello $NAME"
        date '+%A'
        date '+%A, %B %d, %Y'
        ;;
```

```
                Homer's CIS 90 Final Project
          1) My favorite color
          2) Getting started using grep command
          3) An if-then-else statement
          4) More example IF statements
          5) Simple if statement
          6) Another if statement
          7) Logic example
          8) Getting your name
          9) Exit

          Enter Your Choice: 3
Hello Homer Miller
Wednesday
Wednesday, December 03, 2008
Hit the Enter key to return to menu
```

How can we do just one format or the other?

myscript

```
3)      # Commands for Task 3
        NAME=$(cat /etc/passwd | grep $LOGNAME | cut -f5 -d":" )
        echo "Hello $NAME"
        echo "$NAME, Do you like short or long dates?"
        echo -n "Enter 1 for short or 2 for long: "
        read ANSWER
        if [ "$ANSWER" = 1 ]; then
                date '+%A'
        else
                date '+%A, %B %d, %Y'
        fi
        ;;
```

Prompt user for choice then use if-then-else statement

```
Enter Your Choice: 3
Hello Homer Miller
Homer Miller, Do you like short or long dates?
Enter 1 for short or 2 for long: 1
Tuesday
Hit the Enter key to return to menu
```

```
Enter Your Choice: 3
Hello Homer Miller
Homer Miller, Do you like short or long dates?
Enter 1 for short or 2 for long: 2
Tuesday, December 02, 2014
Hit the Enter key to return to menu
```

Scripting Tips

Using the set command

```
[rsimms@opus scripts]$ set dogs cats birds humans

[rsimms@opus scripts]$ echo $1
dogs

[rsimms@opus scripts]$ echo $2
cats

[rsimms@opus scripts]$ echo $3
birds

[rsimms@opus scripts]$ echo $4
humans

[rsimms@opus scripts]$ echo $#
4

[rsimms@opus scripts]$ echo $*
dogs cats birds humans
```

*The **set** command parses the arguments it receives.*

$1 is set to the first argument
$2 is set to the second argument and so forth.

$# is set to the total number of arguments.

```
[rsimms@opus bin]$ echo $(ls)
1975.egg app banner datecal enlightenment hi I myscript myscript.milhom90
myscript.v1 newscript old program quiet quiet.bak script treed tryme
typescript zoom

[rsimms@opus bin]$ set $(ls)

[rsimms@opus bin]$ echo $3
banner

[rsimms@opus bin]$ echo $7
I

[rsimms@opus bin]$ echo $11
1975.egg1

[rsimms@opus bin]$ echo $#
20

[rsimms@opus bin]$ echo "The fifth file in this directory is $5"
The fifth file in this directory is enlightenment
[rsimms@opus bin]$
```

A nice way to be able to reference specific files in a directory

```
[rsimms@opus scripts]$ finger $LOGNAME
Login: rsimms                           Name: Rich Simms
Directory: /home/rsimms                 Shell: /bin/bash
On since Mon May 18 14:38 (PDT) on pts/1 from 207.62.186.30
Mail last read Mon May 18 16:09 2009 (PDT)
No Plan.

[rsimms@opus scripts]$ finger $LOGNAME | head -1
Login: rsimms                           Name: Rich Simms

[rsimms@opus scripts]$ set $(finger $LOGNAME | head -1)

[rsimms@opus scripts]$ echo $1
Login:

[rsimms@opus scripts]$ echo $2
rsimms

[rsimms@opus scripts]$ echo $3
Name:

[rsimms@opus scripts]$ echo $4
Rich

[rsimms@opus scripts]$ echo $5
Simms

[rsimms@opus scripts]$ firstname=$4

[rsimms@opus bin]$ echo My first name is $firstname
My first name is Rich
```

Another way to get a user's first name

Class Exercise

Make a new script in your bin directory

```
cd bin
    vi example777
```

In vi add these lines to your script then save:

```
    set $(finger $LOGNAME | head -1)
    firstname=$4
    echo My first name is $firstname
```

Prepare and run your script

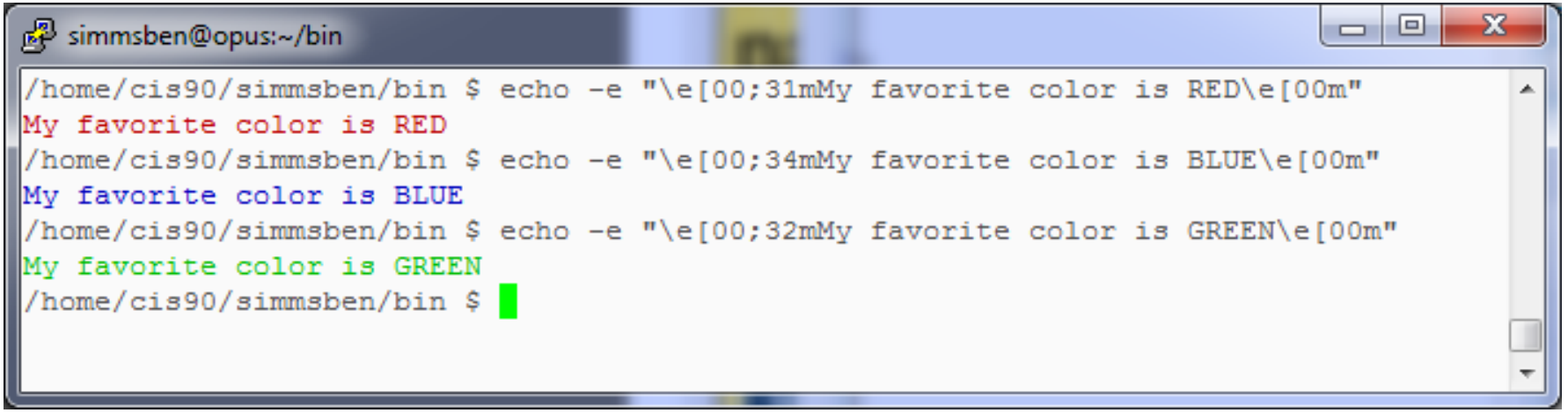
```
    chmod +x example777
    example777
```

Scripting Tips

color

Using Color

Black 0;30	Green 0;32	Red 0;31	Brown 0;33
Dark Gray 1;30	Light Green 1;32	Light Red 1;31	Yellow 1;33
Blue 0;34	Cyan 0;36	Purple 0;35	Light Gray 0;37
Light Blue 1;34	Light Cyan 1;36	Light Purple 1;35	White 1;37



*Use **echo -e "\e[0**n;nn**m"** to turn on color*
(the -e option enables interpretation of backslash escapes)

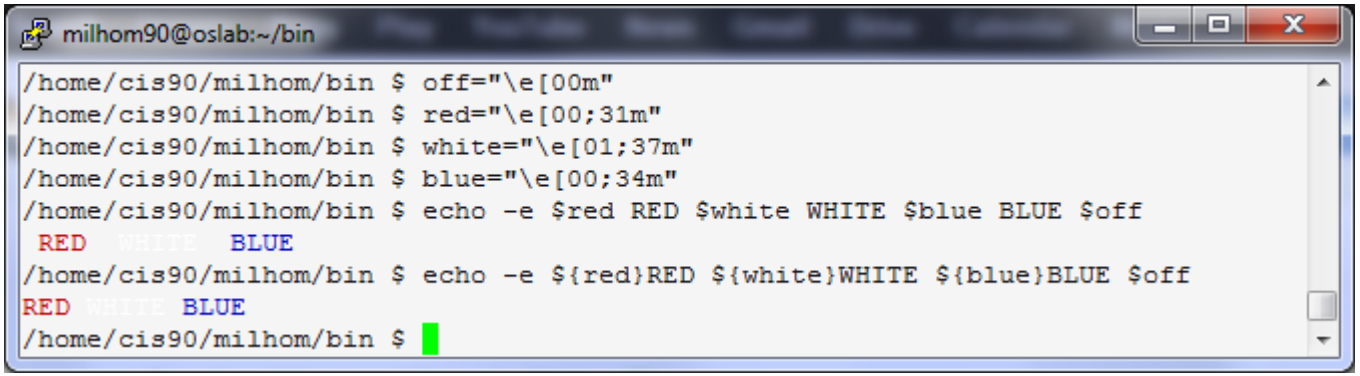
Source: http://hacktux.com/bash/colors

Using Color

```
echo -e "\e[00;32m"
```

*Use **echo -e '\e[00m'** to revert back to normal*

Source: http://hacktux.com/bash/colors




```
off="\e[00m"
red="\e[00;31m"
white="\e[01;37m"
blue="\e[00;34m"
echo -e $red RED $white WHITE $blue BLUE $off
 RED  WHITE  BLUE
echo -e ${red}RED ${white}WHITE ${blue}BLUE $off
RED WHITE BLUE
```

Demonstrating the use of variables and curly braces to make color easier to use.

Curly braces are used to clearly delineate the variable name when there is no blank used as a separator from the next string

Class Exercise

Make a new script in your bin directory

```
cd bin
vi example4271
```

In vi add these lines to your script then save:

```
off="\e[00m"
green="\e[00;32m"
echo -e Hi there, you look a little ${green}GREEN${off} today!
```

Prepare and run your script

```
chmod +x example4271
example4271
```

Scripting Tips

home directories and user names

Going from CIS 90 home directory name ➔ username

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

/home/cis90/simben $ basename $HOME
simben
```

*The **basename** command extracts the filename from the end of a pathname*

```
/home/cis90/simben $ echo $(basename $HOME)
simben

/home/cis90/simben $ echo $(basename $HOME)90
simben90
```

This is how you tack 90 on to the home directory filename

```
/home/cis90/simben $ userid=`echo $(basename $HOME)90`
/home/cis90/simben $ echo The home directory of $userid is $HOME
The home directory of simben90 is /home/cis90/simben
```

Going from CIS 90 home directory name ➔ username

```
/home/cis90/simben $ finger $(basename $HOME)90
Login: simben90                         Name: Benji Simms
Directory: /home/cis90/simben           Shell: /bin/bash
On since Wed May 16 08:09 (PDT) on pts/2 from 50-0-68-
235.dsl.dynamic.fusionbroadband.com
No mail.
Plan:
To pass this course with flying colors!
```

*Determining the username from the home directory name and then using it as an argument to the **finger** command*

Going from CIS 90 username ➔ home directory name

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

This variable holds your username

```
/home/cis90/simben $ echo ${LOGNAME%90}
simben
```

This is how you strip text off the end of a string

```
/home/cis90/simben $ file=`echo ${LOGNAME%90}`
/home/cis90/simben $ echo $file
simben
```

*This sets a new variable named **file** to hold the filename*

```
/home/cis90/simben $ echo The home of $LOGNAME is /home/cis90/$file
The home of simben90 is /home/cis90/simben
```

And this is how you could use it

Scripting Tips

Simple for loop

for loop example

```
/home/cis90/milhom/bin $ for i in hugo sun jin john charlie
> do
> echo Hello $i
> done
Hello hugo
Hello sun
Hello jin
Hello john
Hello charlie
/home/cis90/milhom/bin $
```

for loop example

```
/home/cis90/milhom/bin $ for file in $(ls /usr/bin/pi*)
> do
> echo I found a file named $file
> done
I found a file named /usr/bin/pic
I found a file named /usr/bin/pic2graph
I found a file named /usr/bin/piconv
I found a file named /usr/bin/pidstat
I found a file named /usr/bin/pinentry
I found a file named /usr/bin/pinentry-curses
I found a file named /usr/bin/pinfo
I found a file named /usr/bin/pinky
/home/cis90/milhom/bin $
```

Class Exercise

Make a new script in your bin directory

```
cd bin
vi example808
```

In vi add these lines to your script then save:

```
for name in $(grep cis90 /etc/passwd | cut -f5 -d":" | cut -f1 -d" ")
do
  echo My classmate is named $name
done
```

Prepare and run your script

```
chmod +x example808
example808
```

Wrap up

Commands:

basename	- extract filename form pathname
scp	- secure copy command
tar	- archive command
if then else	- conditionals in scripts
[ ]	- for logic tests in scripts

Next Class

Project is due next week!

Project Workshop

- Make sure you get one "starter" task scripted and working before leaving class today.
- Grade your starter script using the Final Project rubric

Implementing all five tasks (6 points each): • Requirements for each task: –Minimum of 10 "original" script command lines –Has one or more non-generic comments to explain what it is doing –Has user interaction
You don't have to do all of these but do at least five: • Redirecting stdin (5 points) • Redirecting stdout (5 points) • Redirecting stderr (5 points) • Use of permissions (5 points) • Use of filename expansion characters (5 points) • Use of absolute path (5 points) • Use of relative path (5 points) • Use of a PID (5 points) • Use of inodes (5 points) • Use of links (5 points) • Use of scheduling (5 points) • Use of a GID or group (5 points) • Use of a UID or user (5 points) • Use of a /dev/tty device (5 points) • Use of a signal (5 points) • Use of piping (5 points) • Use of an environment variable (5 points) • Use of /bin/mail (5 points) • Use of a conditional (5 points) The maximum for this section is 25 points.

Backup