## Rich's CCC Confer checklist - setup

- ❑ Slides and Project posted
- ❑ WB converted from PowerPoint
- ❑ Print out agenda slide and annotate page numbers

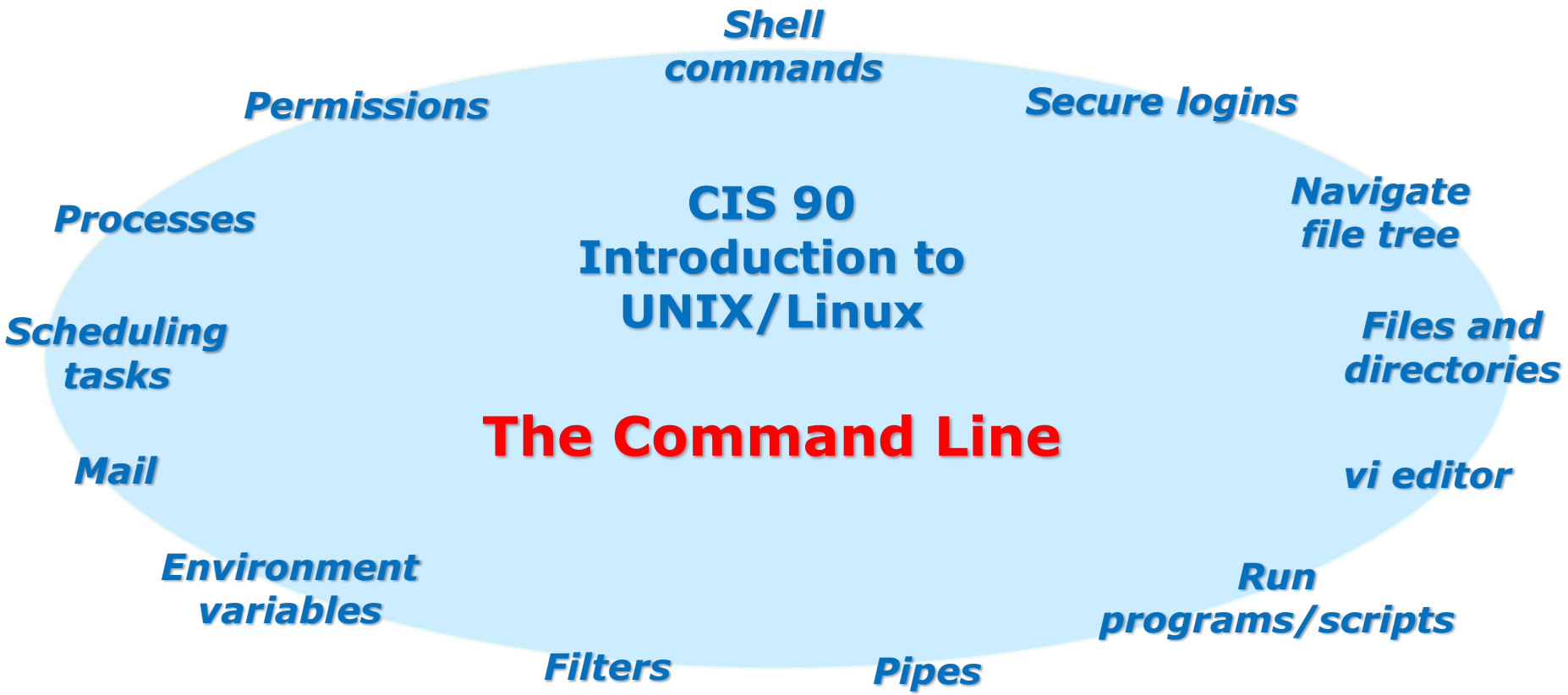
- ❑ Flash cards
- ❑ Page numbers
- ❑ 1st minute quiz - NA
- ❑ Web Calendar summary
- ❑ Web book pages
- ❑ Commands

- ❑ Dog script examples ready

- ❑ Backup slides, CCC info, handouts on flash drive
- ❑ Spare 9v battery for mic
- ❑ Key card for classroom door

- ❑ Update CCC Confer and 3C Media portals

*Last updated 11/28/2017*




## Student Learner Outcomes

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

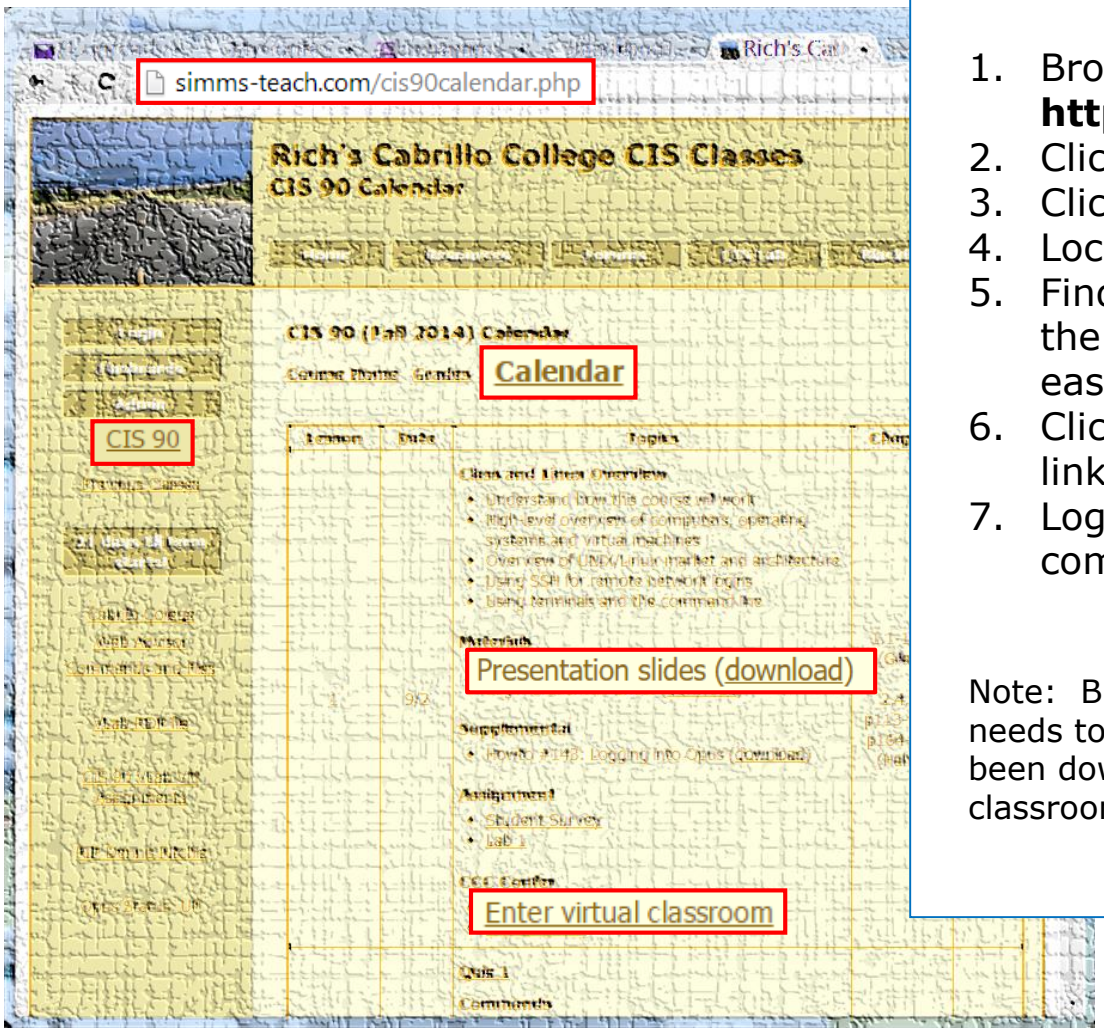
# Introductions and Credits

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

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

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

# Student checklist for attending class

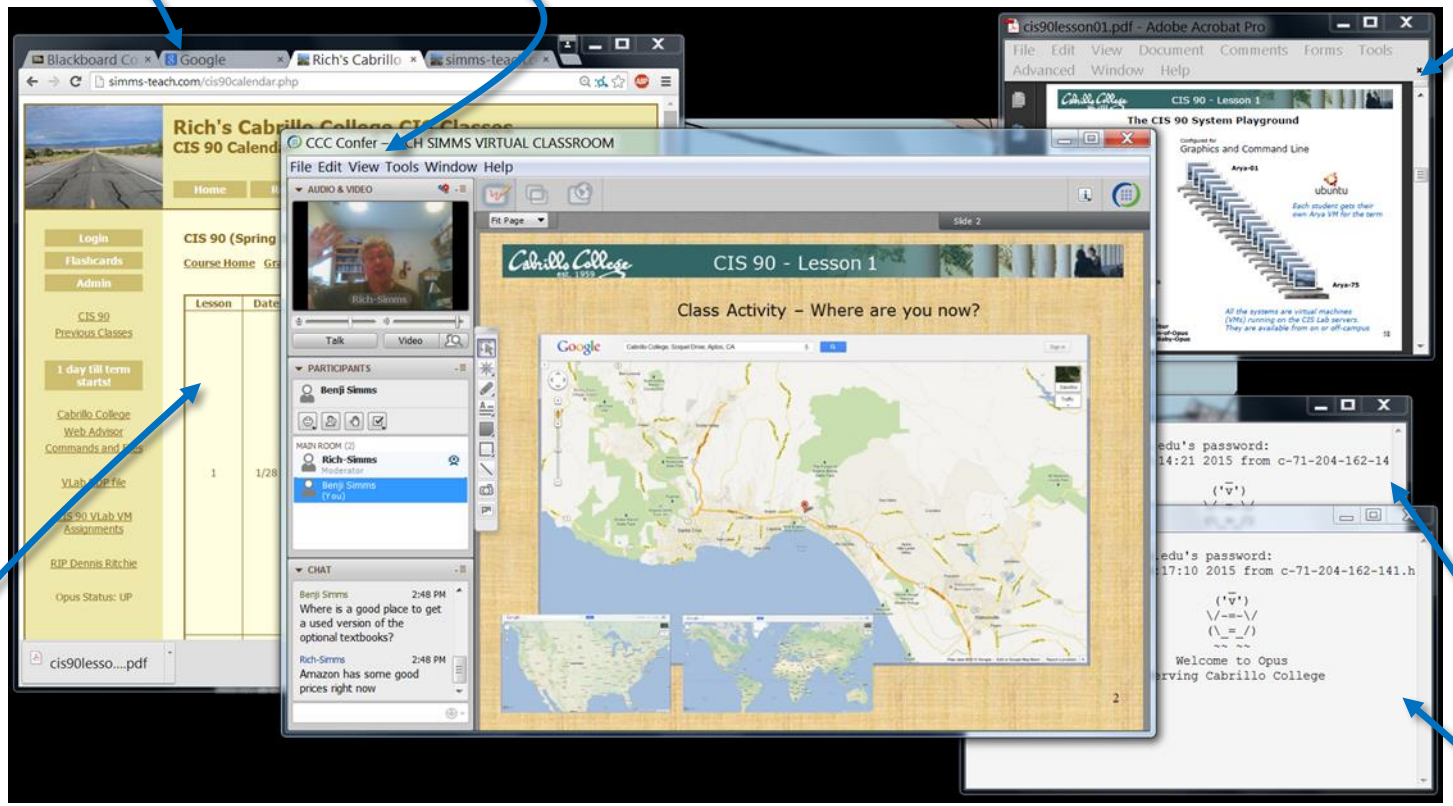



1. Browse to: **http://simms-teach.com**
2. Click the **CIS 90** link.
3. Click the **Calendar** link.
4. Locate today's lesson.
5. Find the **Presentation slides** for the lesson and **download** for easier viewing.
6. Click the **Enter virtual classroom** link to join CCC Confer.
7. Log into Opus-II with Putty or ssh command.

Note: Blackboard Collaborate Launcher only needs to be installed once. It has already been downloaded and installed on the classroom PC's.

# Student checklist for suggested screen layout

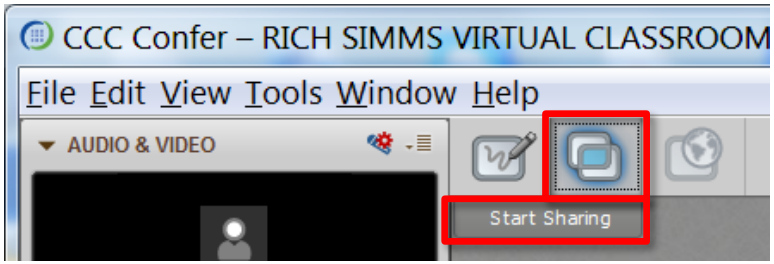
- ❑ *Google*
- ❑ *CCC Confer*
- ❑ *Downloaded PDF of Lesson Slides*



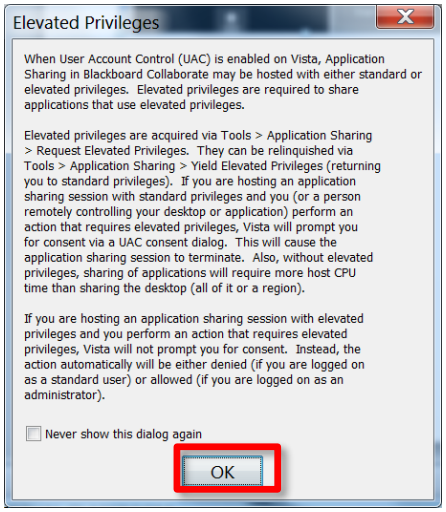

- ❑ *CIS 90 website Calendar page*
- ❑ *One or more login sessions to Opus-II*

# Student checklist for sharing desktop with classmates

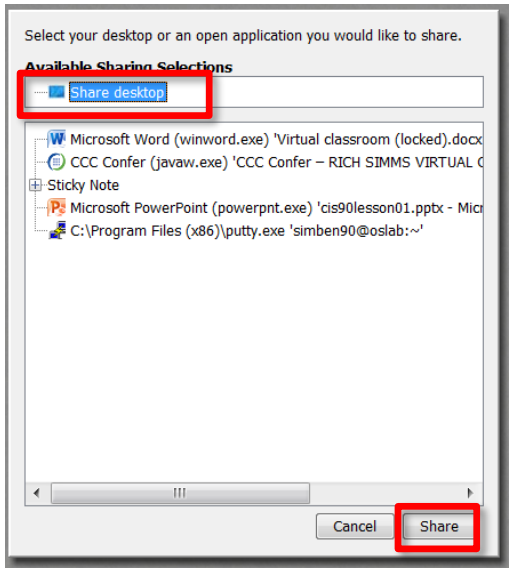
1) Instructor gives you sharing privileges




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



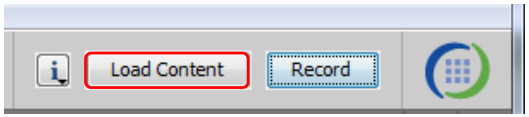

3) Click OK button.




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

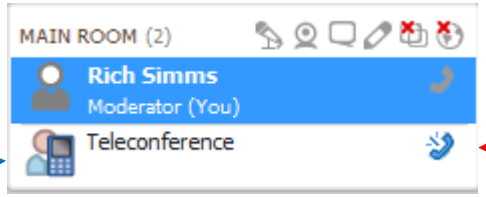
# Rich's CCC Confer checklist - setup

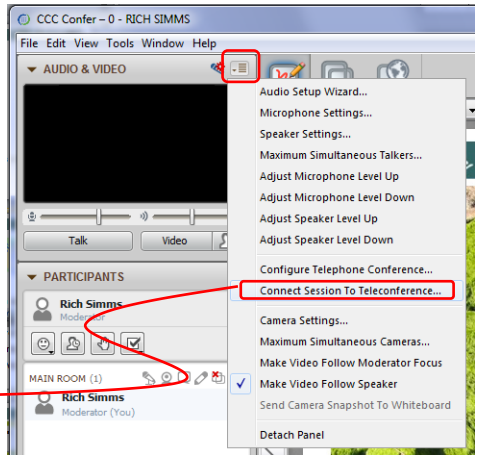
[ ] Preload White Board



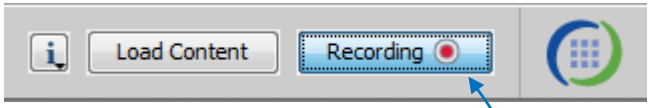

[ ] Connect session to Teleconference

*Session now connected to teleconference*



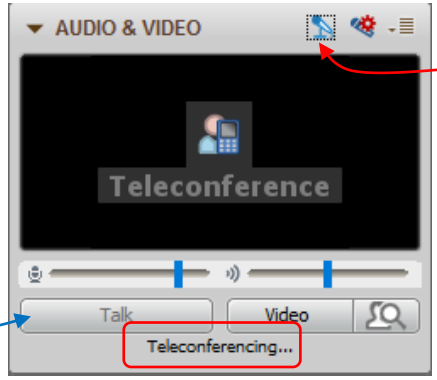

[ ] Is recording on?




*Red dot means recording*

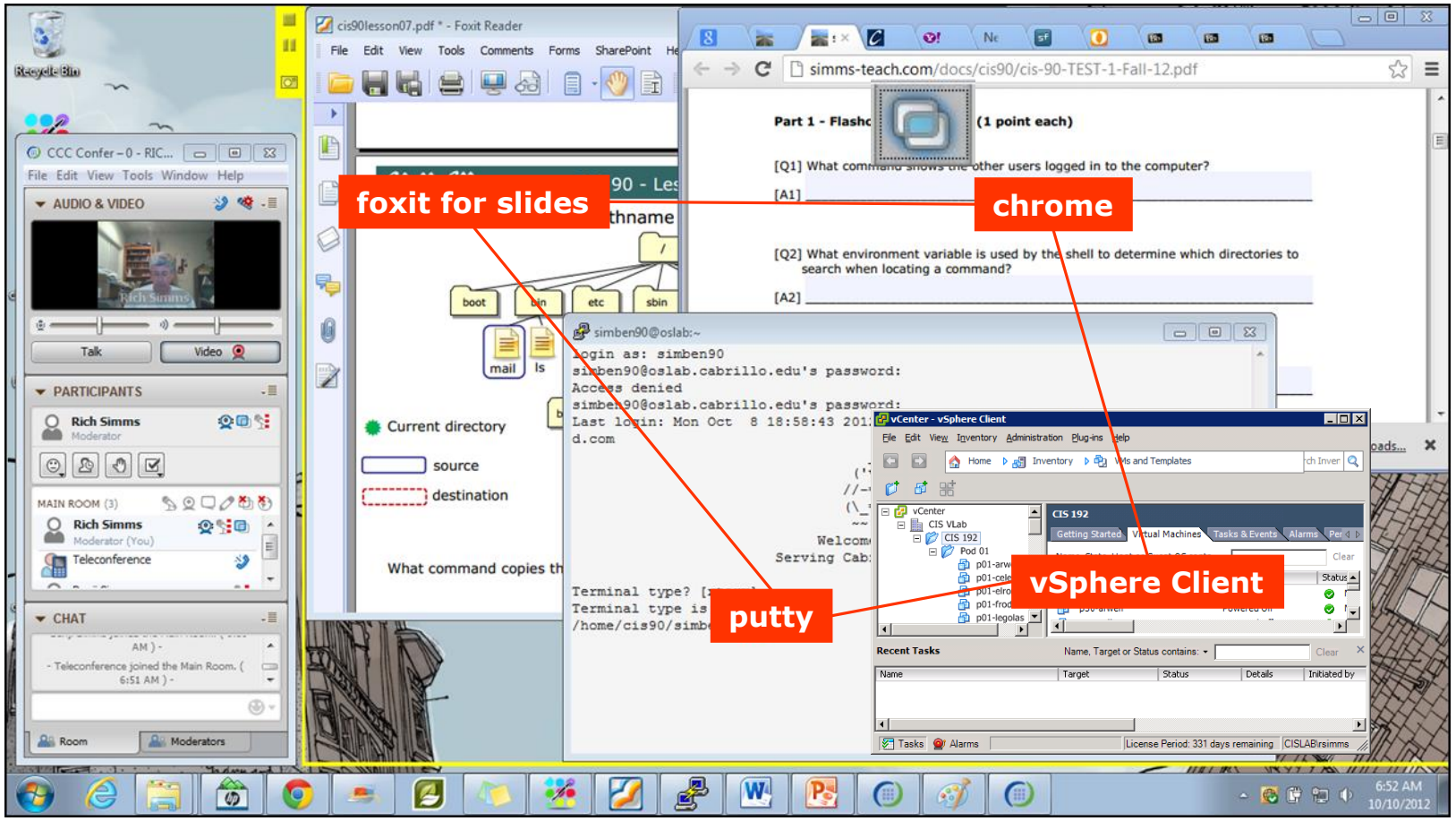
[ ] Use teleconferencing, not mic

*Should be grayed out*



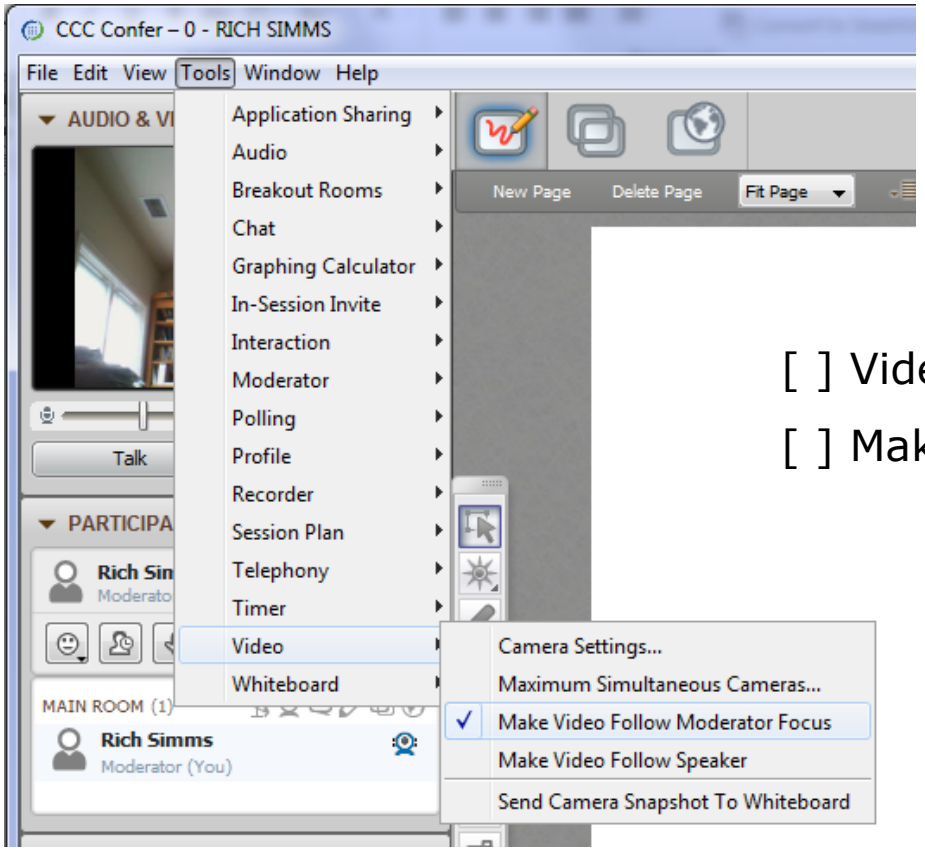

*Should change from phone handset icon to little Microphone icon and the Teleconferencing … message displayed*

# Rich's CCC Confer checklist - screen layout




[ ] layout and share apps

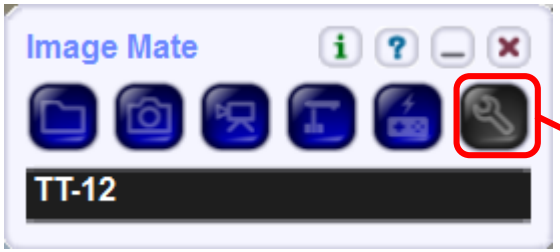
# Rich's CCC Confer checklist - webcam setup

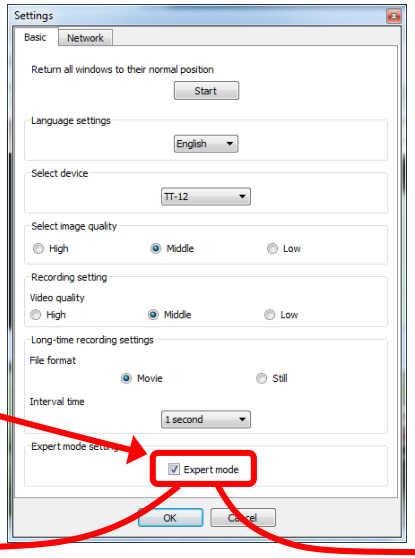



[ ] Video (webcam)

[ ] Make Video Follow Moderator Focus

# Rich's CCC Confer checklist - Elmo



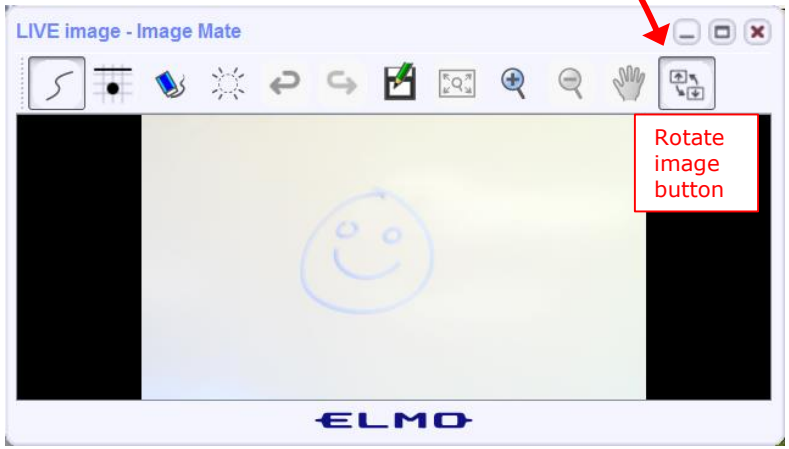




*The "rotate image" button is necessary if you use both the side table and the white board.*

*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



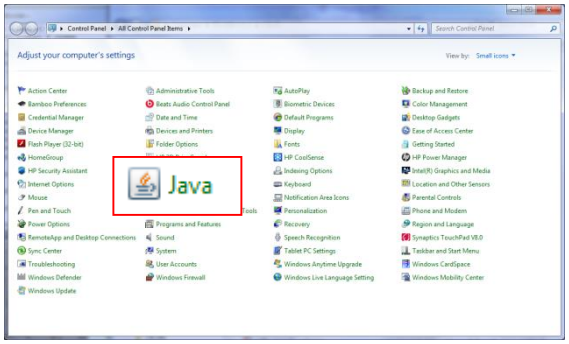

*Run and share the Image Mate program just as you would any other app with CCC Confer*

# Rich's CCC Confer checklist - universal fixes

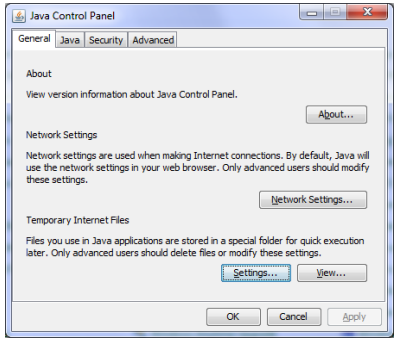
Universal Fix for CCC Confer:

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

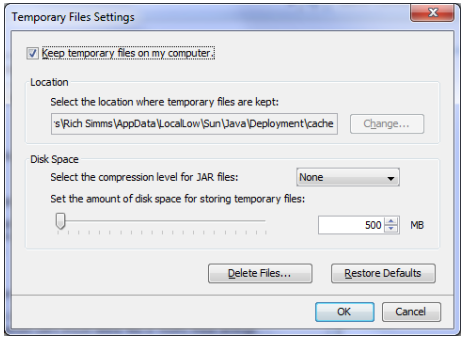
Control Panel (small icons)



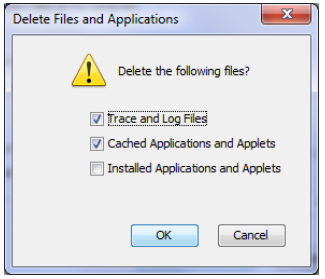

General Tab > Settings…




500MB cache size

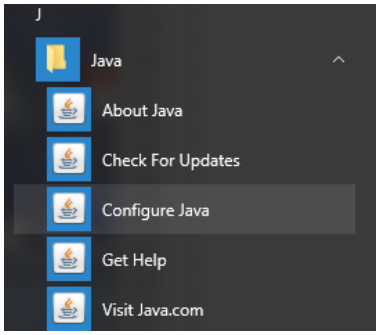



Delete these

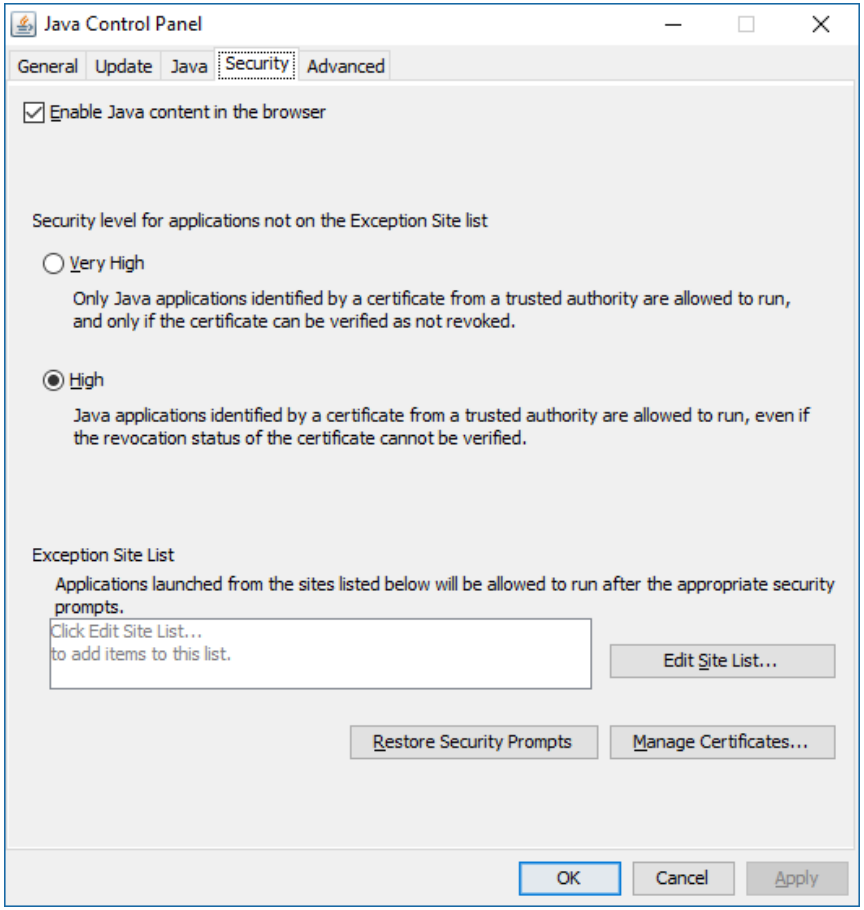



Google Java download

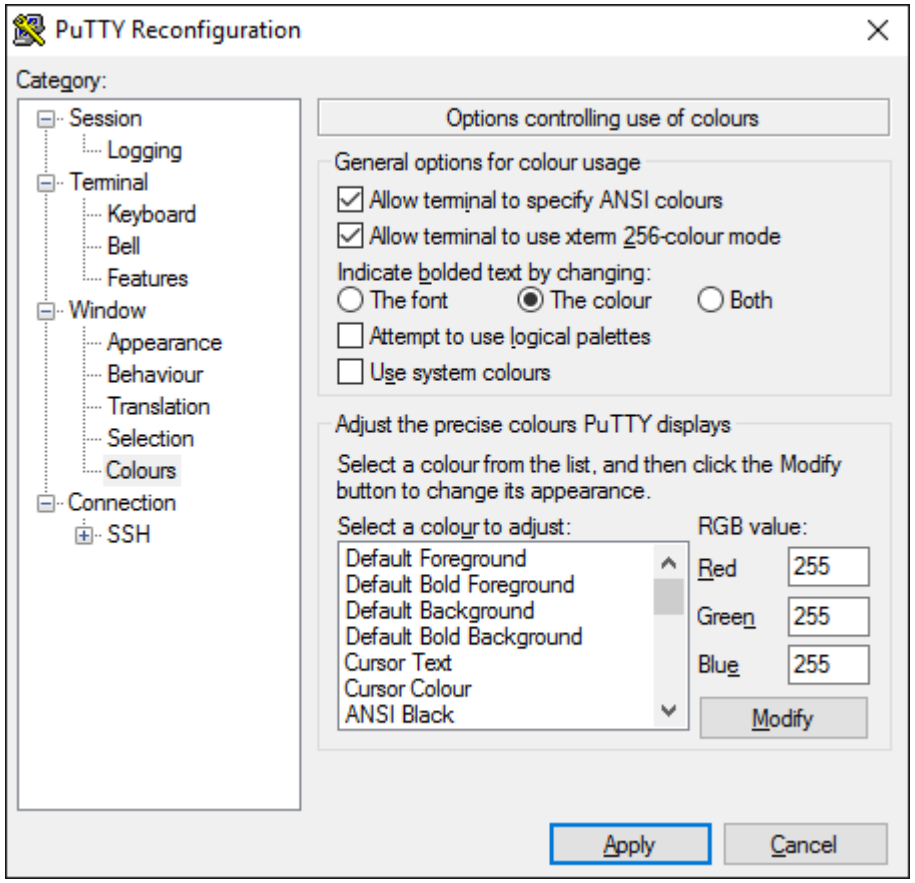
## Rich's CCC Confer checklist - digital certificate work around




1. Open the Java Control Panel
2. Select the **Security** tab
3. Select **Edit Site List...**
4. Select **Add**
5. Click into the white box next to the red exclamation mark and type **https://na-downloads.elluminate.com**
6. Press OK
7. Press **Continue** on the pop-up message
8. Press OK
9. Access your session or recording once more



# Rich's CCC Confer checklist - Putty Colors




http://looselytyped.blogspot.com/2013/02/zenburn-pleasant-color-scheme-for-putty.html

**Putty Colors**

Default Foreground 255 255 255
Default Bold Foreground 255 255 255
Default Background 51 51 51
Default Bold Background 255 2 85
Cursor Text 0 0 0
Cursor Color 0 255 0
ANSI Black 77 77 77
ANSI Black Bold 85 85 85
ANSI Red 187 0 0
ANSI Red Bold 255 85 85
ANSI Green 152 251 152
ANSI Green Bold 85 255 85
ANSI Yellow 240 230 140
ANSI Yellow Bold 255 255 85
ANSI Blue 205 133 63
ANSI Blue Bold 135 206 235
ANSI Magenta 255 222 173
ANSI Magenta Bold 255 85 255
ANSI Cyan 255 160 160
ANSI Cyan Bold 255 215 0
ANSI White 245 222 179
ANSI White Bold 255 255 255

# Start

# Sound Check

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

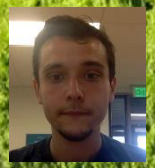
*Instructor can use *96 to mute all student lines.*

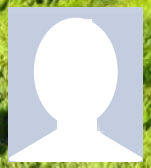
*Volume*
**4 - increase conference volume.*
**7 - decrease conference volume.*
**5 - increase your voice volume.*
**8 - decrease your voice volume.*

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


Marvin


William

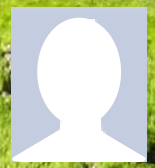

Vinny


Hayden


Nick


Ramon

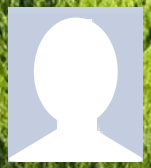

Nicholas


Manuel

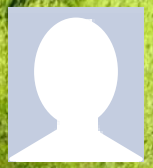

Oscar


Daniel P.


Jason


Brian

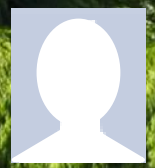

Vincent


Kyle


Sam


Jacobs


Emmanuel


Sean


Alejandro


Moises


Joseph


David


Ben

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

Quiz

# No Quiz Today !

# More Shell Scripting

| Objectives | Agenda |
|---|---|
| • Transfer files between computers<br>• Archive files using tar<br>• Learn some scripting techniques | • No Quiz<br>• Questions<br>• Raspberry Pi demos<br>• ssh and scp<br>• tar<br>• tar + scp<br>• Housekeeping<br>• Refresh on shell scripts<br>• Project<br>• Scripting tips - vi<br>• Scripting tips - sleep<br>• Scripting tips $(cmd) and `cmd`<br>• Scripting tips - field extraction<br>• Scripting tips - simple if<br>• Scripting tips - or logic<br>• Scripting tips - and logic<br>• Scripting tips - file types<br>• Scripting tips - if-then-else<br>• Scripting tips - set command<br>• Scripting tips - color<br>• Scripting tips - username <-> home directory<br>• Scripting tips - simple for loop<br>• Assignment<br>• Wrap up |

# Denise Moss

# Apprenticeships and Internships

# Computer Information Systems (CIS)

Gerlinde Brady, Dean of Career Technical Education

Matt Weis, Internship & Work Experience Instructor

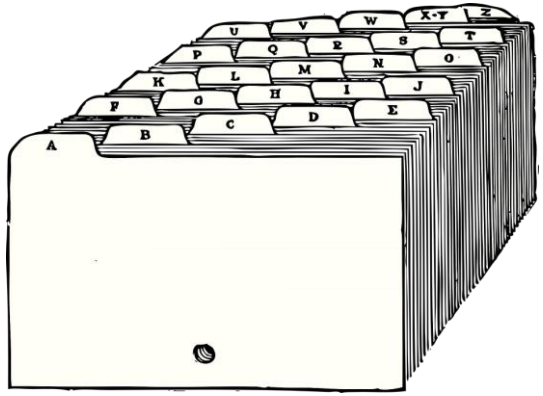
Denise Moss, Apprenticeship Job Developer

# On the Job Training (OJT) & Work Experience

Developing employment, internship and On the-Job-Training (OJT) opportunities in IT sector

*Examples of OJT opportunities*:

- Short-Term Contract
- Part-time/Full Time Employment
- Paid/Unpaid Internships
- Volunteer
- Department of Labor Registered Apprenticeship



# Examples of Placement Opportunities

Help Desk Technician / Computer Support Specialist (Windows and Linux)

System Analyst

Web Developer

Software Developer

Cyber Security

# Help Desk Technician / Computer Support Specialist

Test and evaluate existing network systems

Perform regular maintenance to ensure networks operate correctly

Troubleshoot LANs, WANs, and Internet systems

Provide help and advice to computer users and organizations

# Systems Analyst

Research emerging technologies for potential increases in organizational efficiency and effectiveness

Devise ways to add new functionality to existing computer systems

Oversee installation/configuration of new systems to customize for the organization

# Web Developer

Design and create websites

Create and test applications for a website

Write code for websites using HTML, XML, etc

Work with graphics/designers to develop website layout

Integrate graphics, audio, and video into websites

# Software Developer

Creative minds behind computer programs

Develop applications for underlying systems that run devices or control networks

Analyze users' needs and design/test/develop software to meet those needs

Ensure programs continue to run normally through software maintenance and testing

# Cyber Security

Encrypt data transmissions and establish firewalls

Monitor use of data files and regulate access

Monitor current reports of computer viruses and determine necessary upgrades

# Student Preparation and Placement Services

We assist with Preparation and placement:

Technical training - CIS program

Employment Portfolio development

- Resume development
- Interview coaching
- Social Media (LinkedIn)

Pre-screening

Placement

# Employers & Workforce Partners

- Cabrillo college IT dept
- Cloud Brigade / Launch Brigade
- Bay Federal
- Second Harvest
- Digital Nest
- Workforce Development Board
- And more

# What next?

Email Questions:

Matt Weis maweis@cabrillo.edu

Denise Moss denise.moss.ed@gmail.com

Complete Interest Form (https://goo.gl/forms/0BJfhHDFmZbOhNFh2)

# 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。<br>*He who asks a question is a fool for five minutes; he who does not ask a question remains a fool forever.* |

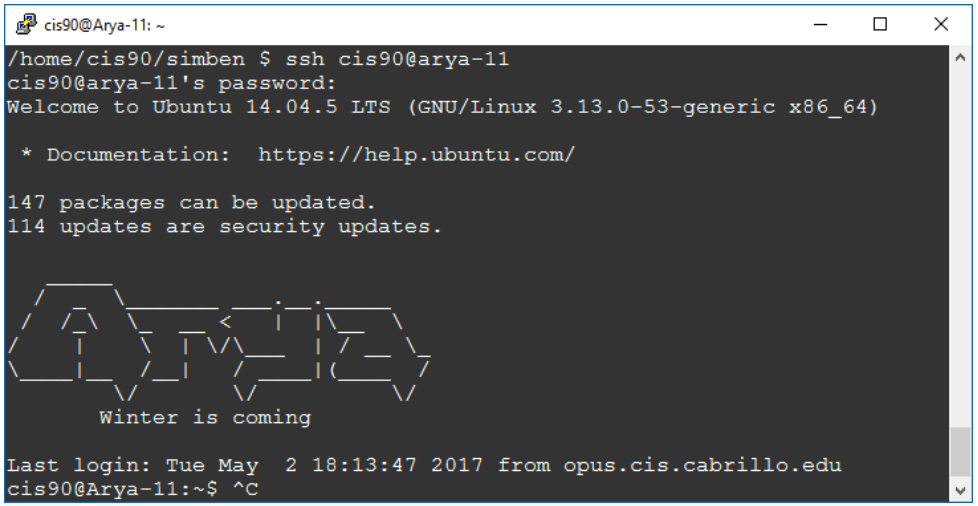
# More on ssh

Running a command on a remote system

# Did you know?

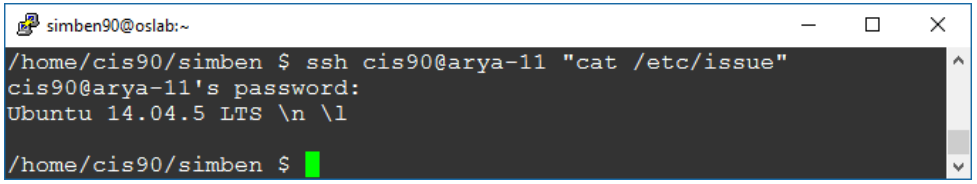
*You can add a command to the end of an ssh command*

**ssh cis90@arya-xx**




*This ssh command logs you into arya-11*

**ssh cis90@arya-xx "cat /etc/issue"**




*This ssh command runs a **cat /etc/issue** command on arya-11*

# Log into your Arya VM using ssh

**All these work from Opus-II:**

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

```
/home/cis90/simben $ ssh cis90@arya-xx
cis90@arya-11's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 3.13.0-53-generic x86_64)

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

81 packages can be updated.
58 updates are security updates.


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

Last login: Sun Mar 12 18:01:01 2017 from opus.cis.cabrillo.edu
cis90@Arya-11:~$
```

*Log into your own Arya VM*

*We've just logged into the Arya VM from Opus-II*

# From Arya run a remote command on Opus-II

Example 1

*This who command will be run on Opus-II*

```
cis90@Arya-11:~$ ssh simben90@opus-ii "who -Hu"
simben90@opus-ii's password:
NAME     LINE         TIME             IDLE          PID COMMENT
rsimms   pts/0        2016-05-03 06:37 02:35        2625 (c-50-174-12-20.hsd1.ca.comcast.net)
rsimms   pts/2        2016-05-01 19:47 00:03       24285 (c-50-174-12-20.hsd1.ca.comcast.net)
jordan90 pts/4        2016-05-03 15:14 00:40       11093 (50.247.74.213)
rsimms   pts/5        2016-05-03 16:34   .         23372 (c-50-174-12-20.hsd1.ca.comcast.net)
pajste90 pts/7        2016-05-03 15:24 01:12       30054 (47-32-184-65.dhcp.snlo.ca.charter.com)
soramr90 pts/8        2016-05-03 15:59 00:02       26035 (63.249.94.142)
soramr90 pts/9        2016-05-03 15:55 00:02       18935 (63.249.94.142)
cis90@Arya-11:~$
```

Example 2

*This variable will be set to the output of the ssh command*

*This pipeline command will be run on Opus-II*

```
cis90@Arya-11:~$ opusUsers=$(ssh simben90@opus-ii "who -s | cut -f1 -d' '")
simben90@opus's password:
cis90@Arya-11:~$ echo $opusUsers
rsimms rsimms jordan90 rsimms farsha154 pajste90 soramr90 soramr90
cis90@Arya-11:~$
```

# More on ssh

Using public/private key instead of a password

# Look Ma, no password

*On Opus-II*

```
/home/cis90/simben $ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/cis90/simben/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/cis90/simben/.ssh/id_rsa.
Your public key has been saved in /home/cis90/simben/.ssh/id_rsa.pub.
The key fingerprint is:
27:d2:ff:0e:ed:01:8a:b3:7e:aa:86:a5:5a:8c:83:79 simben90@oslab.cis.cabrillo.edu
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|                 |
|        .        |
|       . S o     |
|.+   .   o = o   |
|= E+   o . o o   |
| +o .   o.   + . |
|.. ..o+o    .+   |
+-----------------+
/home/cis90/simben $ ls .ssh
id_rsa  id_rsa.pub  known_hosts
```

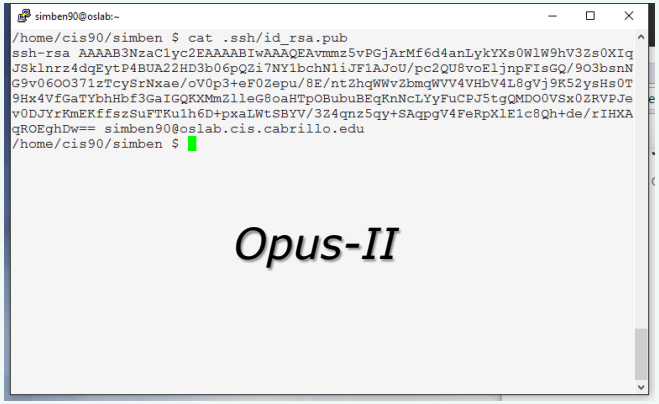
*Your public key (can share with anyone)*

*Your private key (NEVER EVER share with anyone)*

# Look Ma, no password

## Method 1

```
/home/cis90/simben $ cat .ssh/id_rsa.pub
```




```
cis90@Arya-xx:~$ mkdir .ssh
cis90@Arya-xx:~$ chmod 700 .ssh
cis90@Arya-xx:~$ vi .ssh/authorized_keys
```


*Copy and paste your public key on Opus-II into a file named authorized_keys in your .ssh directory on Arya*

## Method 2

```
/home/cis90/simben $  ssh-copy-id cis90@arya-xx
```

# Look Ma, no password

```
/home/cis90/simben $ ssh cis90@arya-03
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.13.0-53-generic x86_64)

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

172 packages can be updated.
115 updates are security updates.


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

Last login: Tue May  3 16:54:19 2016 from opus.cis.cabrillo.edu
cis90@Arya-03:~$
```

*Now you don't need to enter a password when you login to Arya from Opus!*

# 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-xx
cis90@arya-11'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-11:~$
```

*Log into your own Arya VM*

*We've just logged into the Arya VM from Opus-II*

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

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

# Copy one file from Opus-II

**Syntax:**

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

```
cis90@Arya-11:~$ scp simben90@opus-ii:letter .
simben90@opus-ii's password:
letter                                        100% 1044     1.0KB/s   00:00
cis90@Arya-11:~$
```

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

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

*Use your own Opus-II username and password when trying this*

# Copy several files from Opus-II

**Syntax:**

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

```
cis90@Arya-11:~$ scp simben90@opus-ii:poems/Shakespeare/sonnet* .
simben90@opus-ii'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-11:~$
```

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

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

*Use your own Opus-II username and password when trying this*

# Copy (recursively) an entire file tree branch from Opus-II

**Syntax:**

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

```
cis90@Arya-03:~$ scp -r simben90@opus-ii:poems .
simben90@opus-ii'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-03:
```

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

**scp -r -P 2220 simben90@opus-ii.cis.cabrillo.edu:poems .**

*Use your own Opus-II 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

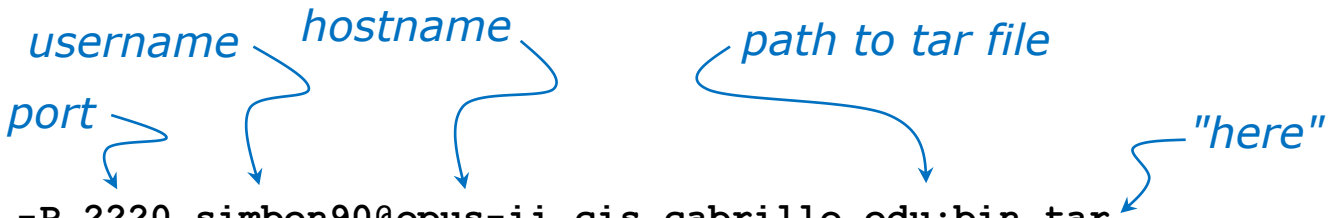
*username* *hostname*

```
/home/cis90/simben $ ssh cis90@arya-xx
cis90@arya-xx'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-xx:~$
```

*Login to your own Arya VM from Opus-II*

# Example

## Copy archived directory to another system




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

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

*Note how archive files are shown in red*

*Copy your bin archive from Opus-II to Arya*

# Example

## Copy archived directory to another system

```
cis90@Arya-xx:~$ 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-xx:~$
```

*extract verbose file*

*name of archive file (tarball)*

*Extract your Opus-II bin directory to your Arya home directory*

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

# Example

## Copy archived directory to another system

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

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

*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-xx:~$ 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-xx:~$
```

# Example

## Copy archived directory to another system

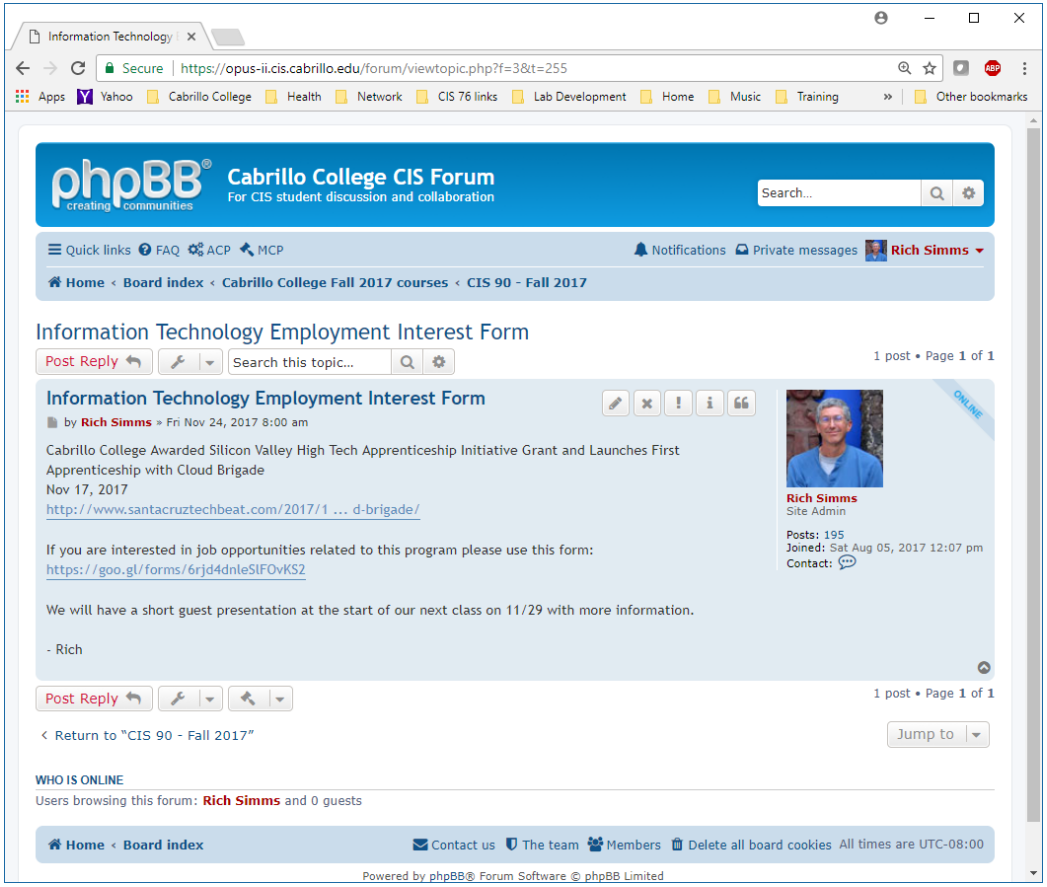
*Run myscript file in the bin directory*

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

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

    1) What is today?
    2) The users on Arya-03
    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

# Internships and Apprenticeships




*If you are interested in CIS related internships then read more on the forum*

https://opus-ii.cis.cabrillo.edu/forum/viewtopic.php?f=3&t=255

# 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 your labx1 file to a different directory and run it using the cis90 user account.

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

5. Extra credit labs are due on the day of the final exam (Test #3). See the calendar page for exact date.

# 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
```

# Heads up on Final Exam

Test #3 (final exam) is MONDAY December 11th 1-3:50PM

| | | | | |
|---|---|---|---|---|
| **Mon** | 12/11 | **Test #3 (the final exam)**<br>**Time**<br>• MONDAY 1:00PM - 3:50PM in Room 828<br>**Materials**<br>• Test (canvas)<br>**CCC Confer**<br>• Enter virtual classroom<br>• Class archives | | 5 posts<br>Lab X1<br>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 3:50PM.
- Working and long distance students can take the test online via CCC Confer and Canvas.
- Working students will need to plan ahead to arrange time off from work for the test.
- Test #3 is **mandatory** (even if you have all the points you want)

## FALL 2017 FINAL EXAMINATIONS SCHEDULE
## DECEMBER 11 TO DECEMBER 16

### DAYTIME FINAL SCHEDULE

**Daytime Classes**: All times in bold refer to the beginning times of classes. **MW/Daily** means Monday alone, Wednesday alone, Monday and Wednesday **or any 3** or more days in any combination. **TTH** means Tuesday alone, Thursday alone, or Tuesday and Thursday. **Classes meeting other combinations of days and/or hours not listed must have a final schedule approved by the Division Dean.**

| STARTING CLASS TIME / DAY(S) | EXAM HOUR | EXAM DATE |
|---|---|---|
| ***Classes starting between:*** | | |
| 6:30 am and 8:55 am, MW/Daily | 7:00 am-9:50 am | Monday, December 11 |
| 9:00 am and 10:15 am, MW/Daily | 7:00 am-9:50 am | Wednesday, December 13 |
| 10:20 am and 11:35 am, MW/Daily | 10:00 am-12:50 pm | Monday, December 11 |
| 11:40 am and 12:55 pm, MW/Daily | 10:00 am-12:50 pm | Wednesday, December 13 |
| 1:00 pm and 2:15 pm, MW/Daily | 1:00 pm-3:50 pm | Monday, December 11 |
| 2:20 pm and 3:35 pm, MW/Daily | 1:00 pm-3:50 pm | Wednesday, December 13 |
| 3:40 pm and 5:30 pm, MW/Daily | 4:00 pm-6:50 pm | Monday, December 11 |
| | | |
| 6:30 am and 8:55 am, TTh | 7:00 am-9:50 am | |
| 9:00 am and 10:15 am, TTh | 7:00 am-9:50 am | |
| 10:20 am and 11:35 am, TTh | 10:00 am-12:50 pm | |
| 11:40 am and 12:55 pm, TTH | 10:00 am-12:50 pm | |
| 1:00 pm and 2:15 pm, TTh | 1:00 pm-3:50 pm | |
| 2:20 pm and 3:35 pm, TTh | 1:00 pm-3:50 pm | |
| 3:40 pm and 5:30 pm, TTh | 4:00 pm-6:50 pm | |
| | | |
| Friday am | 9:00 am-11:50 am | |
| Friday pm | 1:00 pm-3:50 pm | |
| | | |
| Saturday am | 9:00 am-11:50 am | |
| Saturday pm | 1:00 pm-3:50 pm | |

**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. Recommended Preparation: CIS 1L or CIS 72.

Transfer Credit: Transfers to CSU;UC

| Section | Days | Times | Units | Instructor | Room |
|---|---|---|---|---|---|
| 98169 | W | 1:00PM-4:05PM | 3.00 | R.Simms | OL |
| & | Arr. | Arr. | | R.Simms | OL |

Section 98169 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.

| | | | | | |
|---|---|---|---|---|---|
| 98170 | W | 1:00PM-4:05PM | 3.00 | R.Simms | 828 |
| & | Arr. | Arr. | | R.Simms | OL |

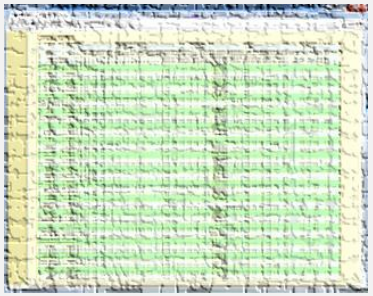
Section 98170 is a Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 50 min online lab per week. For details, see instructor's web page at go.cabrillo.edu/online.

# Where to find your grades

***Send me your survey to get your LOR code name.***

## The CIS 90 website Grades page

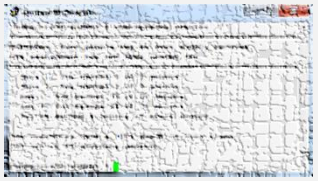
http://simms-teach.com/cis90grades.php



## Or check on Opus-II

**checkgrades** *codename*

*(where codename is your LOR codename)*



Written by Jesse Warren 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**

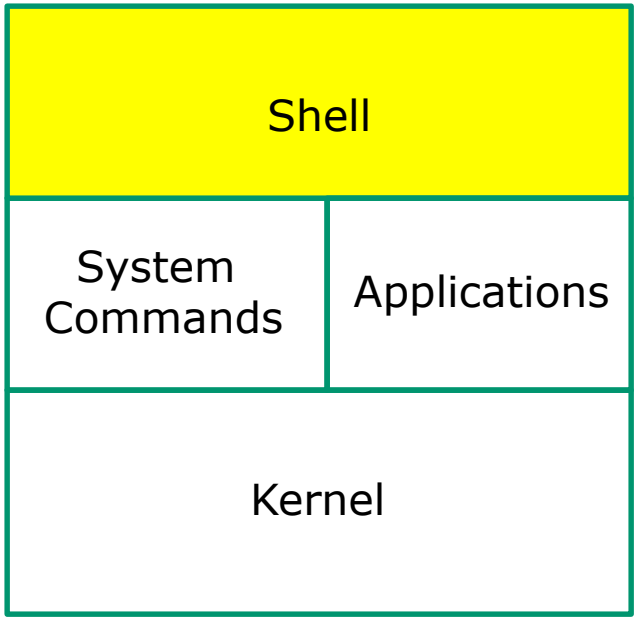
**Points that could have been earned:**

| | |
|---|---|
| 10 quizzes: | 30 points |
| 10 labs: | 300 points |
| 2 tests: | 60 points |
| 3 forum quarters: | 60 points |
| **Total:** | **450 points** |

# 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-ii

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**

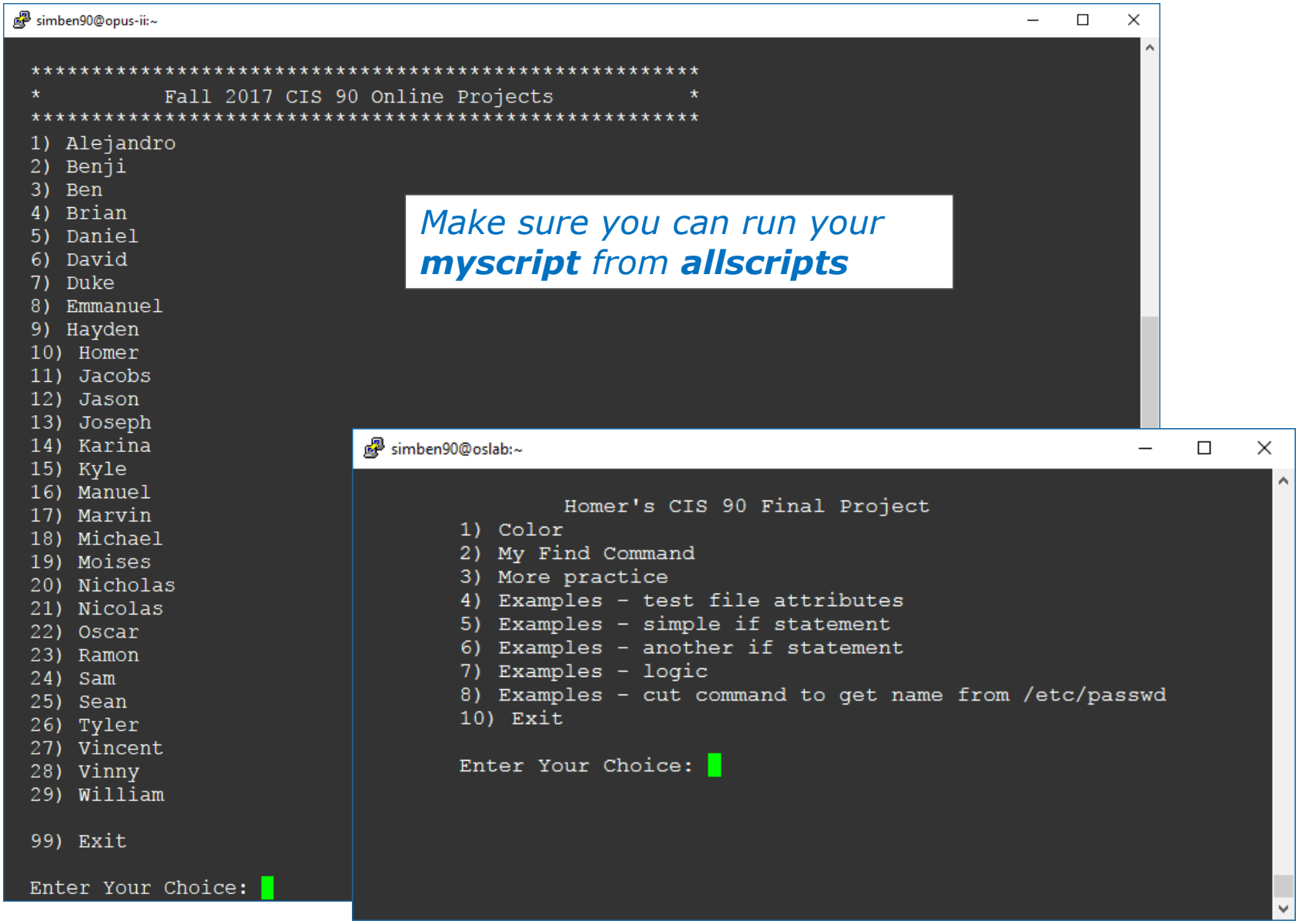
3. Run **allscripts** and verify your script will run without any errors

4. Do the example grep task shown in Lesson 13

| Possible Points | Requirements |
|---|---|
| 30 | Implementing all five tasks (6 points each):<br>• Requirements for each task:<br>– Minimum of 10 "original" script command lines<br>– Has one or more non-generic comments to explain what it is doing<br>– Has user interaction |
| 25 | You don't have to do all of these but do at least five:<br>• Redirecting stdin (5 points)<br>• Redirecting stdout (5 points)<br>• Redirecting stderr (5 points)<br>• Use of permissions (5 points)<br>• Use of filename expansion characters (5 points)<br>• Use of absolute path (5 points)<br>• Use of relative path (5 points)<br>• Use of a PID (5 points)<br>• Use of inodes (5 points)<br>• Use of links (5 points)<br>• Use of color (5 points)<br>• Use of scheduling (5 points)<br>• Use of a GID or group (5 points)<br>• Use of a UID or user (5 points)<br>• Use of a /dev/tty device (5 points)<br>• Use of a signal (5 points)<br>• Use of piping (5 points)<br>• Use of an environment variable (5 points)<br>• Use of /bin/mail (5 points)<br>• Use of a conditional (5 points)<br>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:<br>• Doesn't give full credit to the original author.<br>• Doesn't indicate where the code was obtained from.<br>• Doesn't include licensing terms.<br>• Violates copyright or licensing terms. |
| -up to 90 | For any "malware" scripts that steal credentials, exfiltrate confidential information, remove or encrypt a user's files or creates a denial of service condition on Opus. |
| **Extra credit** | |
| 30 | Up to three additional tasks (10 points each) |

*Plagiarizing another author's code is a NO-NO! All points lost!*

*Scripts that result in unauthorized hacking" is a NO-NO! All points lost!*


simben90@opus-ii:~
**********
* Fall 2017 CIS 90 Online Projects *
**********
1) Alejandro
2) Benji
3) Ben
4) Brian
5) Daniel
6) David
7) Duke
8) Emmanuel
9) Hayden
10) Homer
11) Jacobs
12) Jason
13) Joseph
14) Karina
15) Kyle
16) Manuel
17) Marvin
18) Michael
19) Moises
20) Nicholas
21) Nicolas
22) Oscar
23) Ramon
24) Sam
25) Sean
26) Tyler
27) Vincent
28) Vinny
29) William
99) Exit
Enter Your Choice:
Make sure you can run your myscript from allscripts
simben90@oslab:~
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
10) Exit
Enter Your Choice:

# Don't forget to do this!

*Make sure everyone can run your **myscript** from **allscripts***

**chmod 750 ~**
**chmod 750 ~/bin**
**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:<br>• Doesn't give full credit to the original author.<br>• Doesn't indicate where the code was obtained from.<br>• Doesn't include licensing terms.<br>• Violates copyright or licensing terms. |
| -up to 90 | For any "malware" scripts that steal credentials, exfiltrate confidential information, remove or encrypt a user's files or creates a denial of service condition on Opus. |
| **Extra credit** | |
| 30 | Up to three additional tasks (10 points each) |



*If you are not sure, log into Opus-II as the cis90 user and confirm*

## Project Status

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

rsimms@opus-ii:~

```
[rsimms@opus-ii ~]$ ls -l /home/cis90/*/bin/myscript
-rwxr-x---. 1 alvnic90 cis90  668 Nov 22 16:02 /home/cis90/alvnic/bin/myscript
-rwxrwxr-x. 1 bermoi90 cis90  550 Nov 22 14:41 /home/cis90/bermoi/bin/myscript
-rwxr-x---. 1 boysam90 cis90 1304 Nov 22 15:08 /home/cis90/boysam/bin/myscript
-rwxr-xr-x. 1 bresea90 cis90 2092 Nov 22 16:56 /home/cis90/bresea/bin/myscript
-rwxr-x---. 1 brokyl90 cis90  940 Nov 22 15:27 /home/cis90/brokyl/bin/myscript
-rwxrwxr-x. 1 cis90    cis90  549 Nov 22 15:09 /home/cis90/cis/bin/myscript
-rwxr-x---. 1 jarale90 cis90  787 Nov 15 14:25 /home/cis90/jarale/bin/myscript
-rwxr-x---. 1 lucram90 cis90  948 Nov 22 15:08 /home/cis90/lucram/bin/myscript
-rwxr-x---. 1 macdav90 cis90  988 Nov 22 15:13 /home/cis90/macdav/bin/myscript
-rwxrwxr-x. 1 mccben90 cis90 1141 Nov 23 14:58 /home/cis90/mccben/bin/myscript
-rwxr-x---. 1 ottjac90 cis90 1530 Nov 22 16:04 /home/cis90/ottjac/bin/myscript
-rwxr-x---. 1 panosc90 cis90 1244 Nov 22 16:22 /home/cis90/panosc/bin/myscript
-rwxr------. 1 privin90 cis90 1264 Nov 25 11:08 /home/cis90/privin/bin/myscript
-rwxr-xr-x. 1 samwil90 cis90 1021 Nov 22 15:19 /home/cis90/samwil/bin/myscript
-rwxr-x---. 1 siljas90 cis90 1327 Nov 22 15:35 /home/cis90/siljas/bin/myscript
-rwxr-x---. 1 simben90 cis90  549 Nov 12 17:59 /home/cis90/simben/bin/myscript
-rwxr-xr-x. 1 tbd0390  cis90  653 Nov 22 14:59 /home/cis90/tbd03/bin/myscript
[rsimms@opus-ii ~]$
```

*Is your script "hackable" by other classmates?*

## Project Status

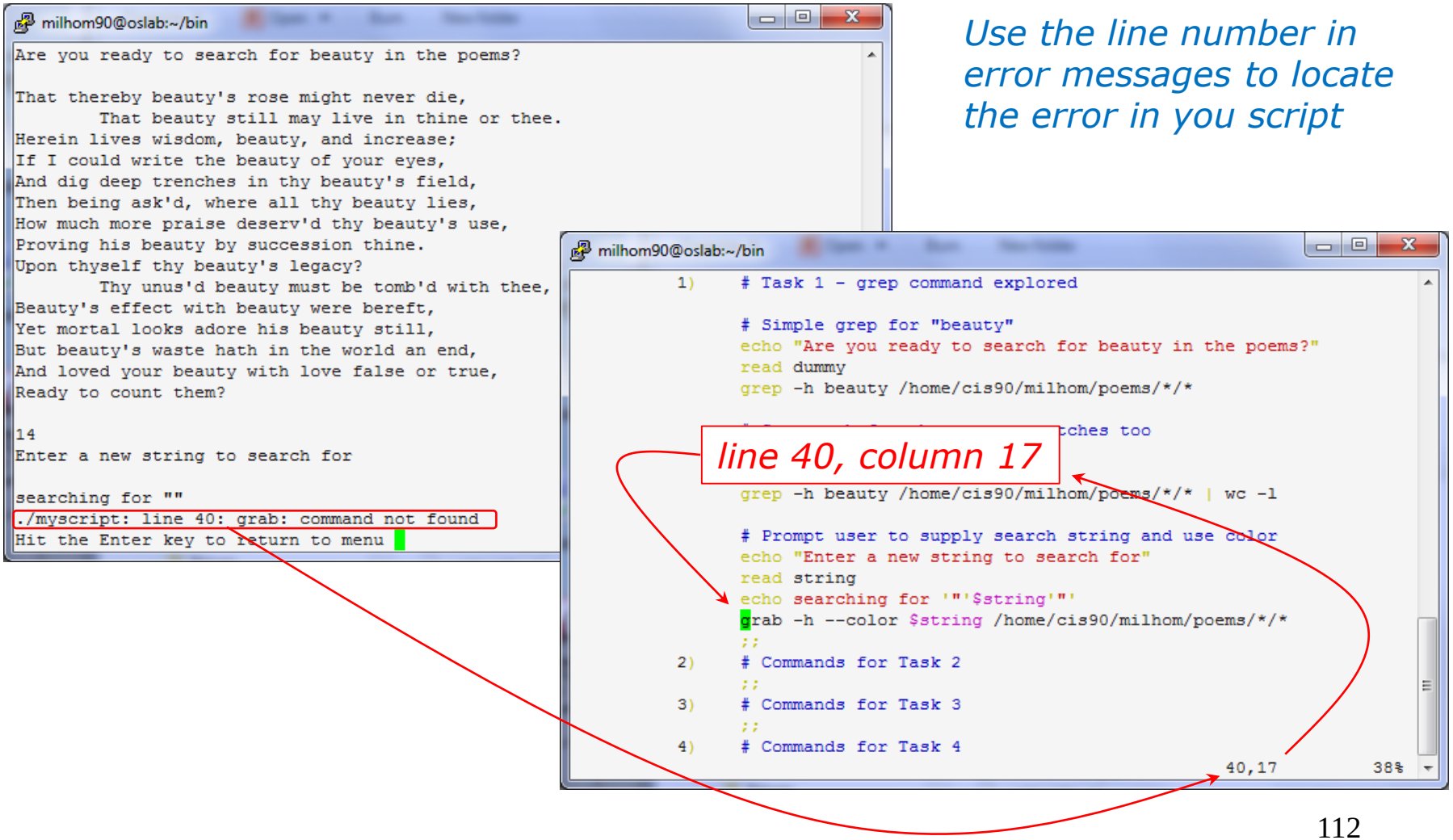
rsimms@opus-ii:~

```
[rsimms@opus-ii ~]$ date
Sat Nov 25 15:50:38 PST 2017
[rsimms@opus-ii ~]$ /home/cis90/bin/checkmyscripts
-rwxr-x---. 1 simben90 cis90 549 Nov 12 17:59 /home/cis90/simben/bin/myscript
ls: cannot access /home/cis90/milhom/bin/myscript: No such file or directory
ls: cannot access /home/cis90/rodduk/bin/myscript: No such file or directory
ls: cannot access /home/cis90/agunic/bin/myscript: No such file or directory
-rwxr-x---. 1 alvnic90 cis90 668 Nov 22 16:02 /home/cis90/alvnic/bin/myscript
-rwxrwxr-x. 1 bermoi90 cis90 550 Nov 22 14:41 /home/cis90/bermoi/bin/myscript
-rwxr-xr-x. 1 bresea90 cis90 2092 Nov 22 16:56 /home/cis90/bresea/bin/myscript
ls: cannot access /home/cis90/howdan/bin/myscript: No such file or directory
-rwxr-x---. 1 lucram90 cis90 948 Nov 22 15:08 /home/cis90/lucram/bin/myscript
ls: cannot access /home/cis90/malman/bin/myscript: No such file or directory
-rwxrwxr-x. 1 mccben90 cis90 1141 Nov 23 14:58 /home/cis90/mccben/bin/myscript
ls: cannot access /home/cis90/mouvin/bin/myscript: No such file or directory
-rwxr-xr-x. 1 samwil90 cis90 1021 Nov 22 15:19 /home/cis90/samwil/bin/myscript
ls: cannot access /home/cis90/wonmar/bin/myscript: No such file or directory
ls: cannot access /home/cis90/johemm/bin/myscript: No such file or directory
-rwxr-x---. 1 boysam90 cis90 1304 Nov 22 15:08 /home/cis90/boysam/bin/myscript
-rwxr-x---. 1 brokyl90 cis90 940 Nov 22 15:27 /home/cis90/brokyl/bin/myscript
-rwxr-x---. 1 jarale90 cis90 787 Nov 15 14:25 /home/cis90/jarale/bin/myscript
-rwxr-x---. 1 macdav90 cis90 988 Nov 22 15:13 /home/cis90/macdav/bin/myscript
ls: cannot access /home/cis90/ngujos/bin/myscript: No such file or directory
-rwxr-x---. 1 ottjac90 cis90 1530 Nov 22 16:04 /home/cis90/ottjac/bin/myscript
-rwxr-x---. 1 panosc90 cis90 1244 Nov 22 16:22 /home/cis90/panosc/bin/myscript
ls: cannot access /home/cis90/prihay/bin/myscript: No such file or directory
-rwxr-----. 1 privin90 cis90 1264 Nov 25 11:08 /home/cis90/privin/bin/myscript
-rwxr-x---. 1 siljas90 cis90 1327 Nov 22 15:35 /home/cis90/siljas/bin/myscript
-rwxr-xr-x. 1 tbd0390 cis90 653 Nov 22 14:59 /home/cis90/tbd03/bin/myscript
[rsimms@opus-ii ~]$
```

*Don't wait till the last minute!*

# Scripting Tips
# vi

# Line Numbers in errors and vi




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

# Color Syntax

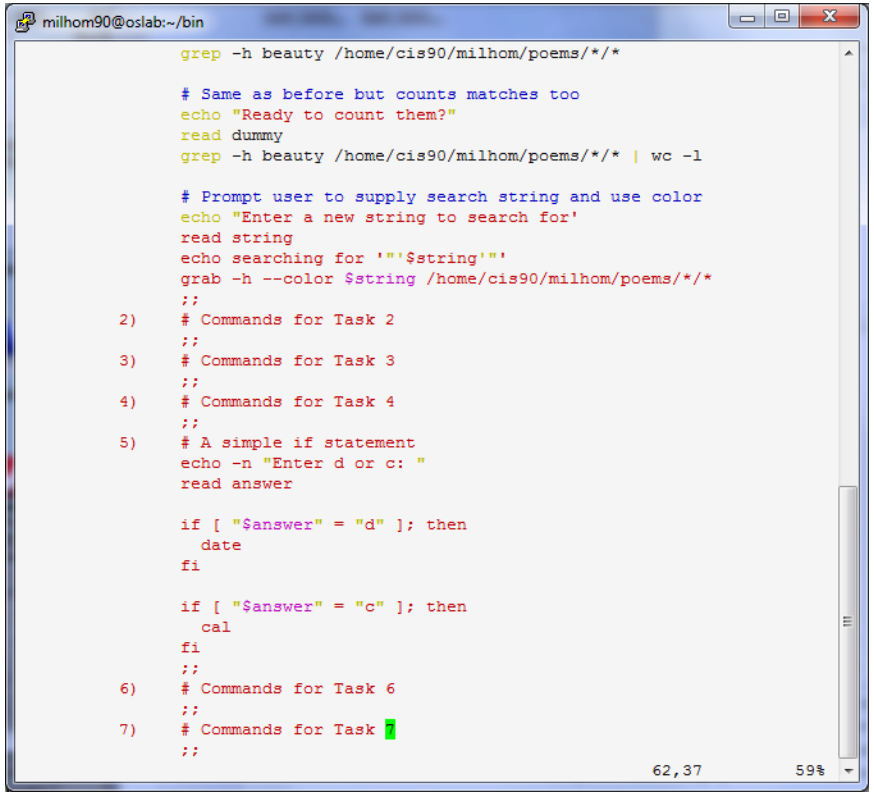
```
milhom90@oslab:~/bin
/home/cis90/milhom/bin $ ./myscript
./myscript: line 79: unexpected EOF while looking for matching `"'
./myscript: line 83: syntax error: unexpected end of file
/home/cis90/milhom/bin $
```

```
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

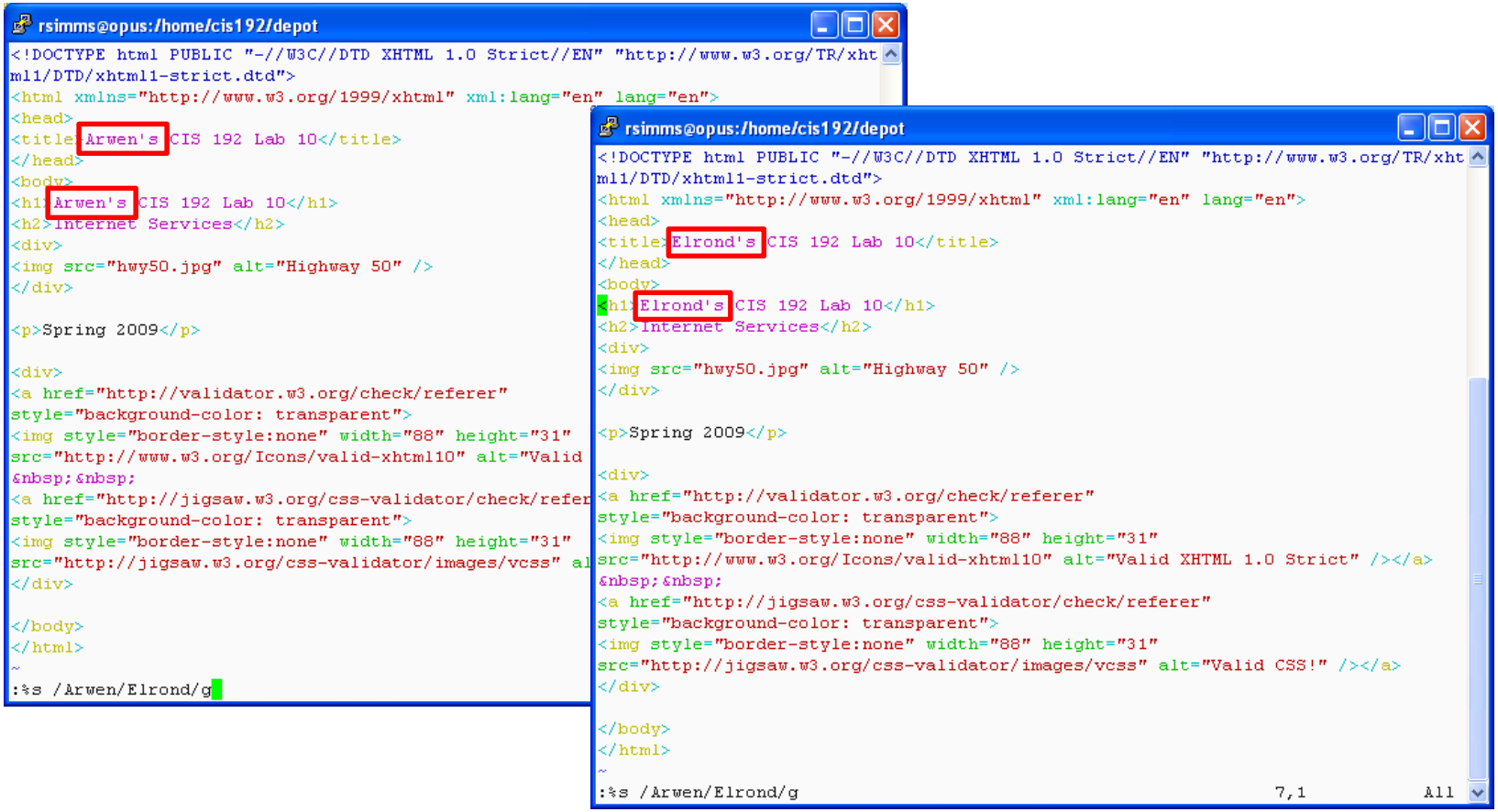




*One small change for script developer, one giant leap for script execution*

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

# 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 back tics around the command to evaluate*

# 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:

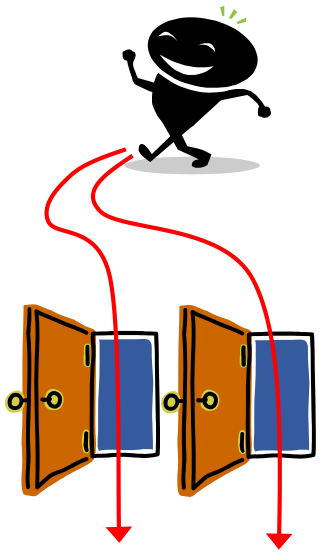
```
vi ~milhom90/bin/myscript
```

# Scripting Tips

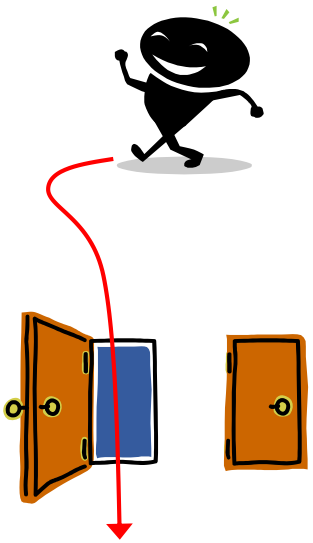
## if statement with "or"

| p | q | p or q |
|---|---|---|
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

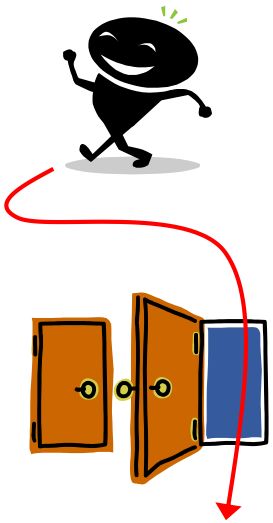
# OR logic



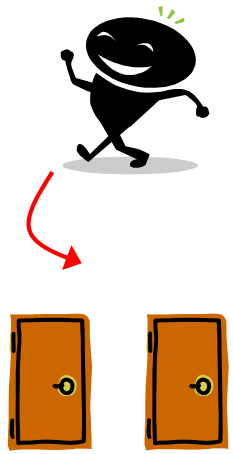
*Yes*



*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

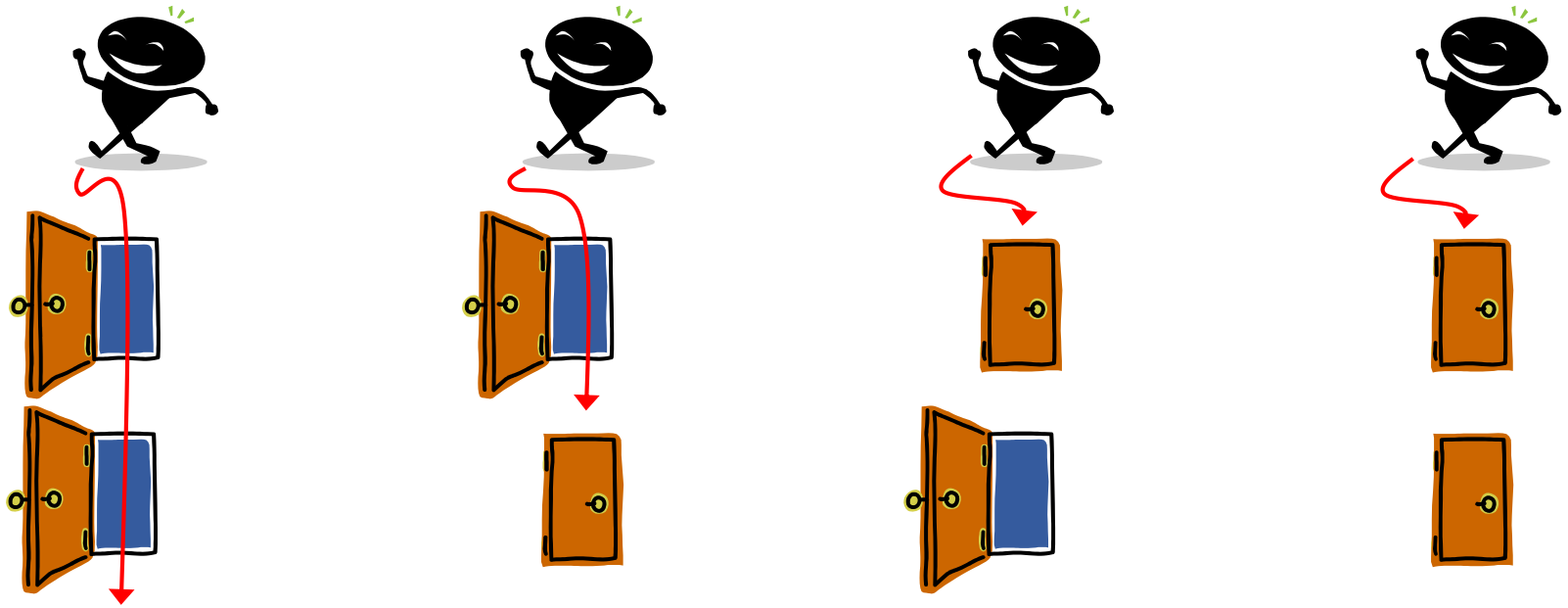
```
chmod +x example654
example654
```

# 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



*Yes* *No* *No* *No*

# 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

## More if statement examples

# Combining Conditionals

```
#!/bin/bash
while true; do
  echo; echo Some flowers: petunias roses tulips mums
  read -p "Enter one of the flowers above or q to end: " response

  if [ "$response" == "q" ] || [ "$response" == "Q" ]; then
    exit
  fi

  if [ "$response" == "petunias" ]; then
    echo "..  We have some red and blues ones left"
  fi

  if [ "$response" == "roses" ]; then
    echo "..  We have some yellow and peach ones left"
  fi

  if [ "$response" == "tulips" ]; then
    echo "..  Sorry we are all out"
  fi

  if [ "$response" == "mums" ]; then
    echo "..  All colors are available"
  fi

done
exit
```

*The developer wants to do something different for any choice the user selects.*

*This works but is not optimal because you have to execute all if statements even when an earlier match is found.*

# Combining Conditionals

```
#!/bin/bash
while true; do
  echo; echo Some flowers: petunias roses tulips mums
  read -p "Enter one of the flowers above or q to end: " response

  if [ "$response" == "q" ] || [ "$response" == "Q" ]; then
    exit
  else
    if [ "$response" == "petunias" ]; then
      echo ".. We have some red and blues ones left"
    else
      if [ "$response" == "roses" ]; then
        echo ".. We have some yellow and peach ones left"
      else
        if [ "$response" == "tulips" ]; then
          echo ".. Sorry we are all out"
        else
          if [ "$response" == "mums" ]; then
            echo ".. All colors are available"
          fi
        fi
      fi
    fi
  fi
done
exit
```

*The developer wants to do something different for any choice the user selects.*

*Using an else clause is a better way to do this.*

# Combining Conditionals

```
#!/bin/bash
while true; do
  echo; echo Some flowers: petunias roses tulips mums
  read -p "Enter one of the flowers above or q to end: " response

  if [ "$response" == "q" ] || [ "$response" == "Q" ]; then
    exit
  elif [ "$response" == "petunias" ]; then
    echo "..  We have some red and blues ones left"
  elif [ "$response" == "roses" ]; then
    echo "..  We have some yellow and peach ones left"
  elif [ "$response" == "tulips" ]; then
    echo "..  Sorry we are all out"
  elif [ "$response" == "mums" ]; then
    echo "..  All colors are available"
  fi

done
exit
```

*The developer wants to do something different for any choice the user selects.*

*Using elif is a little cleaner and easier to modify later.*

# Combining Conditionals

```
#!/bin/bash
while true; do
  clear
  echo -n "Flowers
   1) petunias
   2) roses
   3) tulips
   4) mums

  Select one from above (1-4) or q to quit: "

read response
  case $response in
  [qQ]) exit;;
     1) echo "..  We have some red and blues ones left";;
     2) echo "..  We have some yellow and peach ones left";;
     3) echo "..  Sorry we are all out";;
     4) echo "..  All colors are available";;
  esac
  sleep 2
done
exit
```

*The developer wants to do something different for any choice the user selects.*

*A case statement is another way to handle this.*

# Scripting Tips

## Shortcuts for conditionals

# Conditionals without "if", "then" or "else"

*To do something when command is successful*

```
/home/cis90/simben $ [ -e letter ] && echo file exists
file exists
/home/cis90/simben $ [ -e bogus ] && echo file exists
```

*To do something when command fails*

```
/home/cis90/simben $ [ -e letter ] || echo file does not exist
/home/cis90/simben $ [ -e bogus ] || echo file does not exist
file does not exist
```

# Conditionals without "if", "then" or "else"

*To do something either way*

```
/home/cis90/simben $ ping -c1 -W1 moogle.com > /dev/null && echo up || echo down
down

/home/cis90/simben $ ping -c1 -W1 google.com > /dev/null && echo up || echo down
up
```

*To do something either way*

```
/home/cis90/simben $ grep -r love poems/ > /dev/null && echo found || echo not found
found

/home/cis90/simben $ grep -r nasa poems/ > /dev/null && echo found || echo not found
not found
```

# Scripting Tips

## Parsing with set

# Parsing with set

```
[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.

$* is set to a concatenation of all aguments

# Parsing with set

```
[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 $1
1975.egg

[rsimms@opus bin]$ echo $#
20

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

# Parsing with set

```
[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 $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*

# Parsing with set

```
/home/cis90/simben $ cat /etc/passwd | grep $LOGNAME
simben90:x:1201:1090:Benji Simms:/home/cis90/simben:/bin/bash

/home/cis90/simben $ myAccount=$(cat /etc/passwd | grep $LOGNAME)
/home/cis90/simben $ echo $myAccount
simben90:x:1201:1090:Benji Simms:/home/cis90/simben:/bin/bash
/home/cis90/simben $ echo $myAccount | cut -f5 -d":"
Benji Simms


/home/cis90/simben $ echo $IFS

/home/cis90/simben $ echo '"'$IFS'"'
" "

/home/cis90/simben $ IFS=":"
/home/cis90/simben $ set $myAccount
/home/cis90/simben $ echo My name is $5 and my home directory is $6
My name is Benji Simms and my home directory is /home/cis90/simben
```

*Normally a blank is uses to separate arguments*

*That can be changed*

*Using set as an alternative to cut to extract strings from lines of text*

# 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 |

```
/home/cis90/simben/bin $ echo -e "\e[00;31mMy favorite color is RED\e[00m"
My favorite color is RED
/home/cis90/simben/bin $ echo -e "\e[00;34mMy favorite color is BLUE\e[00m"
My favorite color is BLUE
/home/cis90/simben/bin $ echo -e "\e[00;32mMy favorite color is GREEN\e[00m"
My favorite color is GREEN
/home/cis90/simben/bin $
```

*Use **echo -e "\e[0**n;nn**m"** to turn on color and **\e[00m** to turn it off.*

*(the -e option enables interpretation of backslash escapes)*

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

# Using Color

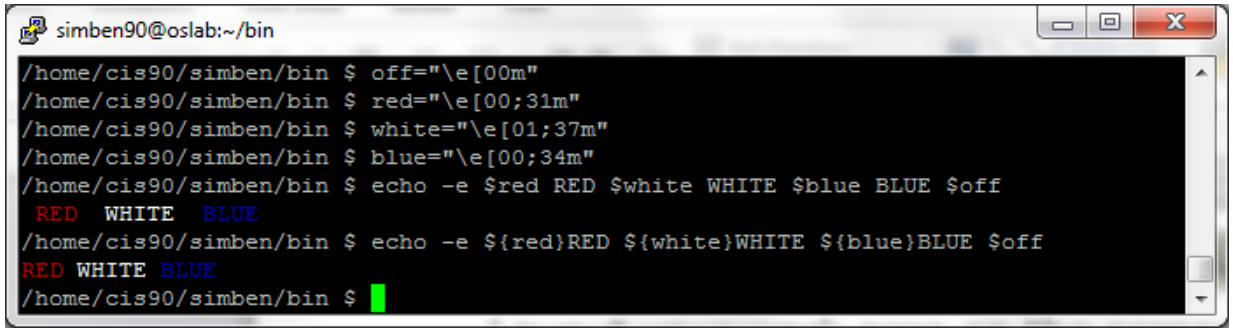
```
/home/cis90/simben/bin $ echo -e "\e[00;32m"

/home/cis90/simben/bin $ head -4 ~/letter
Hello Mother!  Hello Father!

Here I am at Camp Granada.  Things are very entertaining,
and they say we'll have some fun when it stops raining.
/home/cis90/simben/bin $ echo -e '\e[00m'

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

*Change to color green*

*Revert color back to normal*

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

# Using Color




```
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 separate the variable name from adjacent text strings:

- $redRED is null
- ${red}RED is "\e[00;31mRED"

# 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

## Opus-II usernames to home directories and vice-versa

# 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 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*

# Going from CIS 90 username ➔ home directory name

*Another way to do the same thing*

```
/home/cis90/simben/bin $ homeDir=$(grep $LOGNAME /etc/passwd | cut -f6 -d":")
/home/cis90/simben/bin $ echo The home of $LOGNAME is $homeDir
The home of simben90 is /home/cis90/simben
```

*The 6th field of every line in /etc/passwd is the that user's home directory*

# Scripting Tips

## Simple for loop

# for loop examples

*On command line*

```
/home/cis90/simben $ for name in hugo sun jin john charlie
> do
>   echo Hello $name
> done
Hello hugo
Hello sun
Hello jin
Hello john
Hello charlie
/home/cis90/simben $
```

*In script file*

```
/home/cis90/simben $ cat loop1
for name in hugo sun jin john charlie; do
  echo Hello $name
done
/home/cis90/simben $ ./loop1
Hello hugo
Hello sun
Hello jin
Hello john
Hello charlie
/home/cis90/simben $
```

# for loop examples

*On command line*

```
/home/cis90/simben $ for file in $(ls /usr/bin/pip*)
> do
>   echo I found a file named $file
> done
I found a file named /usr/bin/pip
I found a file named /usr/bin/pip2
I found a file named /usr/bin/pip2.6
```

*In script file*

```
/home/cis90/simben $ cat loop2
for file in $(ls /usr/bin/pip*); do
  echo I found a file named $file
done
/home/cis90/simben $ ./loop2
I found a file named /usr/bin/pip
I found a file named /usr/bin/pip2
I found a file named /usr/bin/pip2.6
/home/cis90/simben $
```

# for loop examples

*On command line*

```
/home/cis90/simben $ for (( i=1; i<10; i++ ))
> do
>   echo i=$i
> done
i=1
i=2
i=3
i=4
i=5
i=6
i=7
i=8
i=9
/home/cis90/simben $
```

*In script file*

```
/home/cis90/simben $ cat loop3
for ((i=1; i<10; i++)); do
  echo i=$i
done
/home/cis90/simben $ ./loop3
i=1
i=2
i=3
i=4
i=5
i=6
i=7
i=8
i=9
/home/cis90/simben $
```

# 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
```

# Scripting Tips

## Pulling integers from files and adding them

# Adding integers with let

```
/home/cis90/simben $ cat datafile
apples 20
oranges 25

/home/cis90/simben $ costApples=$(cat datafile | grep apples | cut -f2 -d" ")
/home/cis90/simben $ echo "TRACE costApples=$costApples"
TRACE costApples=20

/home/cis90/simben $ costOranges=$(cat datafile | grep oranges | cut -f2 -d" ")
/home/cis90/simben $ echo "TRACE costOranges=$costOranges"
TRACE costOranges=25

/home/cis90/simben $ let sum=$costApples+$costOranges
/home/cis90/simben $ echo The total cost is $sum
The total cost is 45
```

*We get the cost of apples and oranges from a text file and add them*

# Adding integers with double parentheses

```
/home/cis90/simben $ cat datafile
apples 20
oranges 25

/home/cis90/simben $ costApples=$(cat datafile | grep apples | cut -f2 -d" ")
/home/cis90/simben $ echo "TRACE costApples=$costApples"
TRACE costApples=20

/home/cis90/simben $ costOranges=$(cat datafile | grep oranges | cut -f2 -d" ")
/home/cis90/simben $ echo "TRACE costOranges=$costOranges"
TRACE costOranges=25

/home/cis90/simben $ sum2=$((costApples+costOranges))
/home/cis90/simben $ echo The total cost is $sum2
The total cost is 45
```

*We get the cost of apples and oranges from a text file and add them*

# Bash Arrays

# Arrays

```
/home/cis90/simben/bin $ cat example9881
#!/bin/bash

names[0]="Homer"
names[1]="Benji"
names[2]="Sky"

echo "names[1] = ${names[1]}"
echo "size of names = ${#names[*]}"

for (( i=0; i<${#names[*]}; i++ )); do
  echo "names[$i] = ${names[$i]}"
done
exit
```

*Bash support one dimensional arrays*

```
/home/cis90/simben/bin $ ./example9881
names[1] = Benji
size of names = 3
names[0] = Homer
names[1] = Benji
names[2] = Sky
```

# shift

# shift

```
/home/cis90/simben/bin $ cat example9872
#!/bin/bash

poets=$(ls $HOME/poems)
set $poets

while [ "$1" != "" ]; do
  echo "Poet = $1"
  shift
done

exit
```

*Shifting off parsed arguments on the left.*

```
/home/cis90/simben/bin $ ./example9872
Poet = Angelou
Poet = Anon
Poet = Blake
Poet = Dickenson
Poet = Neruda
Poet = Shakespeare
Poet = Yeats
```

# functions

# functions

```
/home/cis90/simben/bin $ cat example6599
#!/bin/bash

function userInfo() {
  userID=$1
  name=$(grep $userID /etc/passwd | cut -f5 -d":")
  shell=$(grep $userID /etc/passwd | cut -f7 -d":")
  echo "Username: $userID"
  echo "  Name = $name"
  echo "  Shell = $shell"
}

read -p "Enter username: " id
userInfo $id

exit
```

*A simple function example*

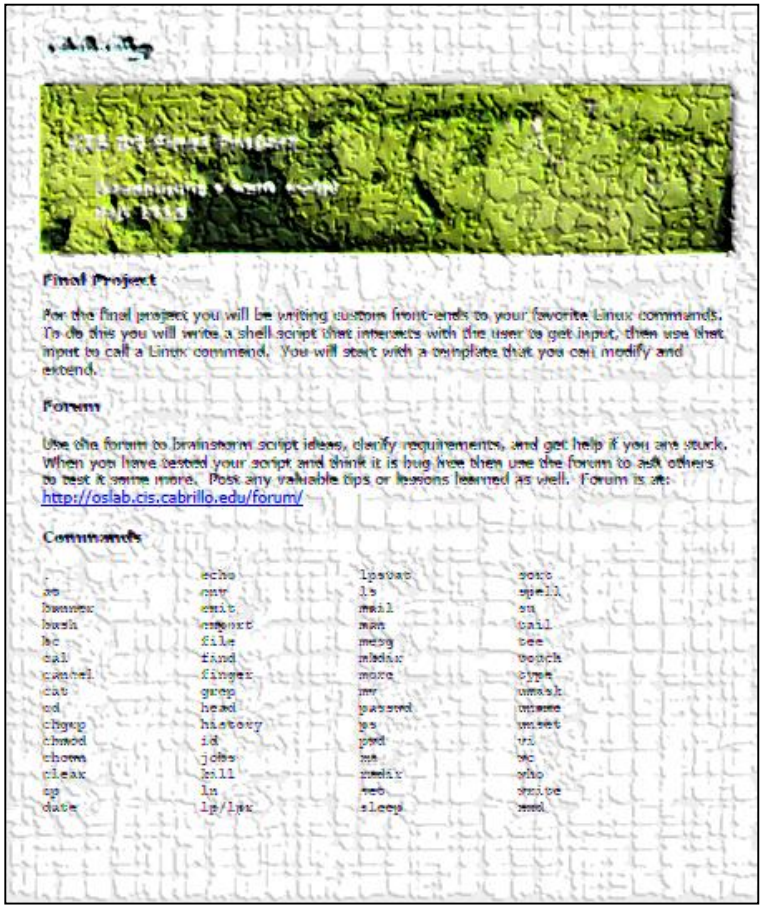
```
/home/cis90/simben/bin $ ./example6599
Enter username: milhom90
Username: milhom90
  Name = Homer Miller
  Shell = /bin/bash
```


Assignment

# Next Class

**Project is due next week!**

# Finish your project!



**Final Project**

For the final project you will be writing custom front-ends to your favorite Linux commands. To do this you will write a shell script that interacts with the user to get input, then use that input to call a Linux command. You will start with a template that you can modify and extend.

**Forum**

Use the forum to brainstorm script ideas, clarify requirements, and get help if you are stuck. When you have tested your script and think it is bug free then use the forum to ask others to test it some more. Post any valuable tips or lessons learned as well. Forum is at: http://oslab.cis.cabrillo.edu/forum/

**Commands**

| | | | |
|---|---|---|---|
| . | echo | lpstat | sort |
| ar | env | ls | spell |
| banner | exit | mail | su |
| bash | export | man | tail |
| bc | file | mesg | tee |
| cal | find | mkdir | touch |
| cancel | finger | more | tput |
| cat | grep | mv | umask |
| cd | head | passwd | uname |
| chgrp | history | ps | unset |
| chmod | id | pwd | vi |
| chown | jobs | rm | wc |
| clear | kill | rmdir | who |
| cp | ln | set | write |
| date | lp/lpr | sleep | xmkl |

*Be sure to review the grading rubric to make sure you didn't miss anything.*


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

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

No Quiz
Project due

**Work on final project - due in one week**

Optional extra credit labs

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

# Backup