



Rich's CCC Confer checklist - setup

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

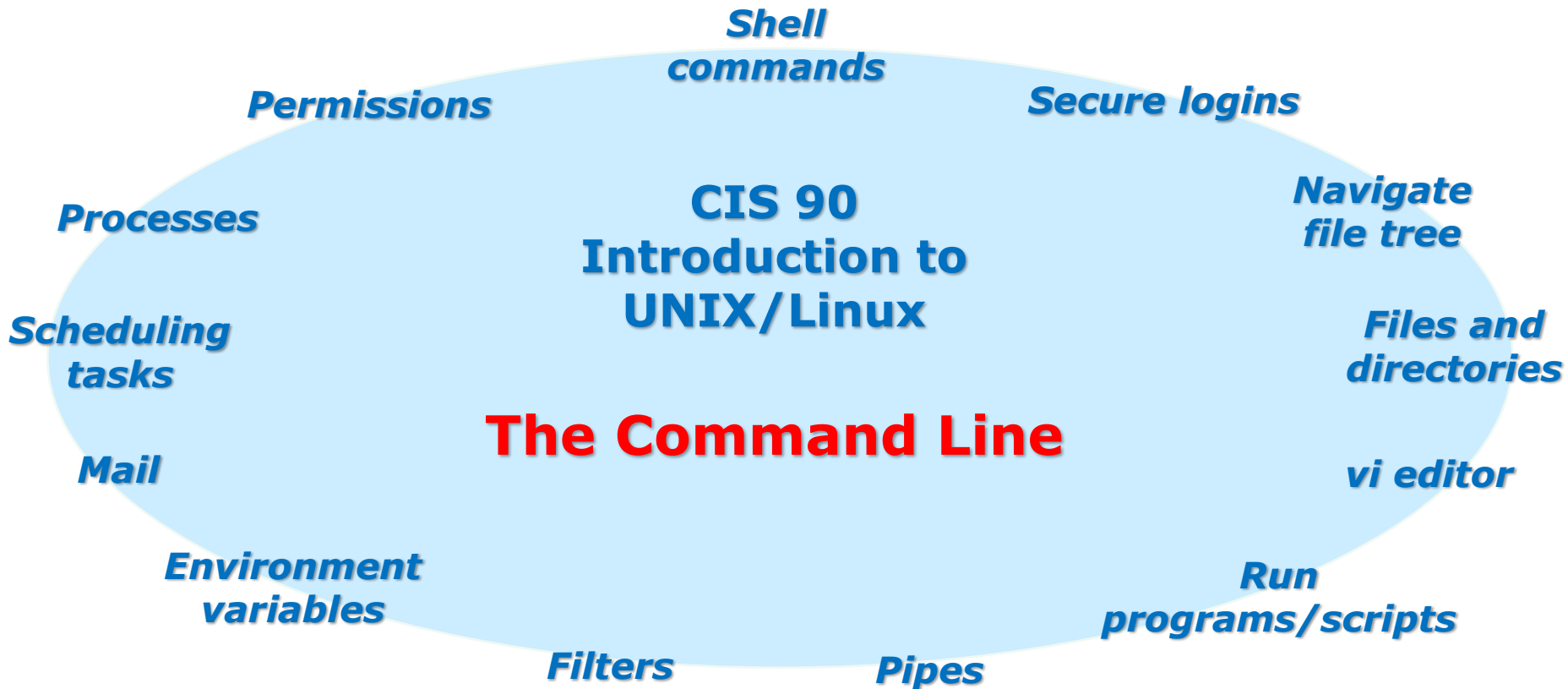
- ☐ Flash cards
- ☐ Page numbers
- ☐ 1st minute quiz - NA
- ☐ Web Calendar summary
- ☐ Web book pages
- ☐ Commands

- ☐ 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 5/2/2017



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

The screenshot shows a web browser at the URL simms-teach.com/cis90calendar.php. The page title is "Rich's Cabrillo College CIS Classes CIS 90 Calendar". On the left sidebar, the "CIS 90" link is highlighted. The main content area shows a calendar for "CIS 90 (Fall 2014) Calendar" with tabs for "Course Dates", "Grades", and "Calendar". The "Calendar" tab is selected, showing a table with columns for "Lesson", "Date", "Topics", and "Link". The first row is for Lesson 1 on 9/2, with topics including "Class and Linux Overview" and "Methods". A red box highlights the "Presentation slides (download)" link. Below the table, there are links for "Supplemental" (PowerPoint), "Assignments" (Student Survey, Lab 1), "CIS 90 Confer", and "Enter virtual classroom" (highlighted with a red box). Other links like "Quiz 1" and "Comments" are also visible.

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 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.

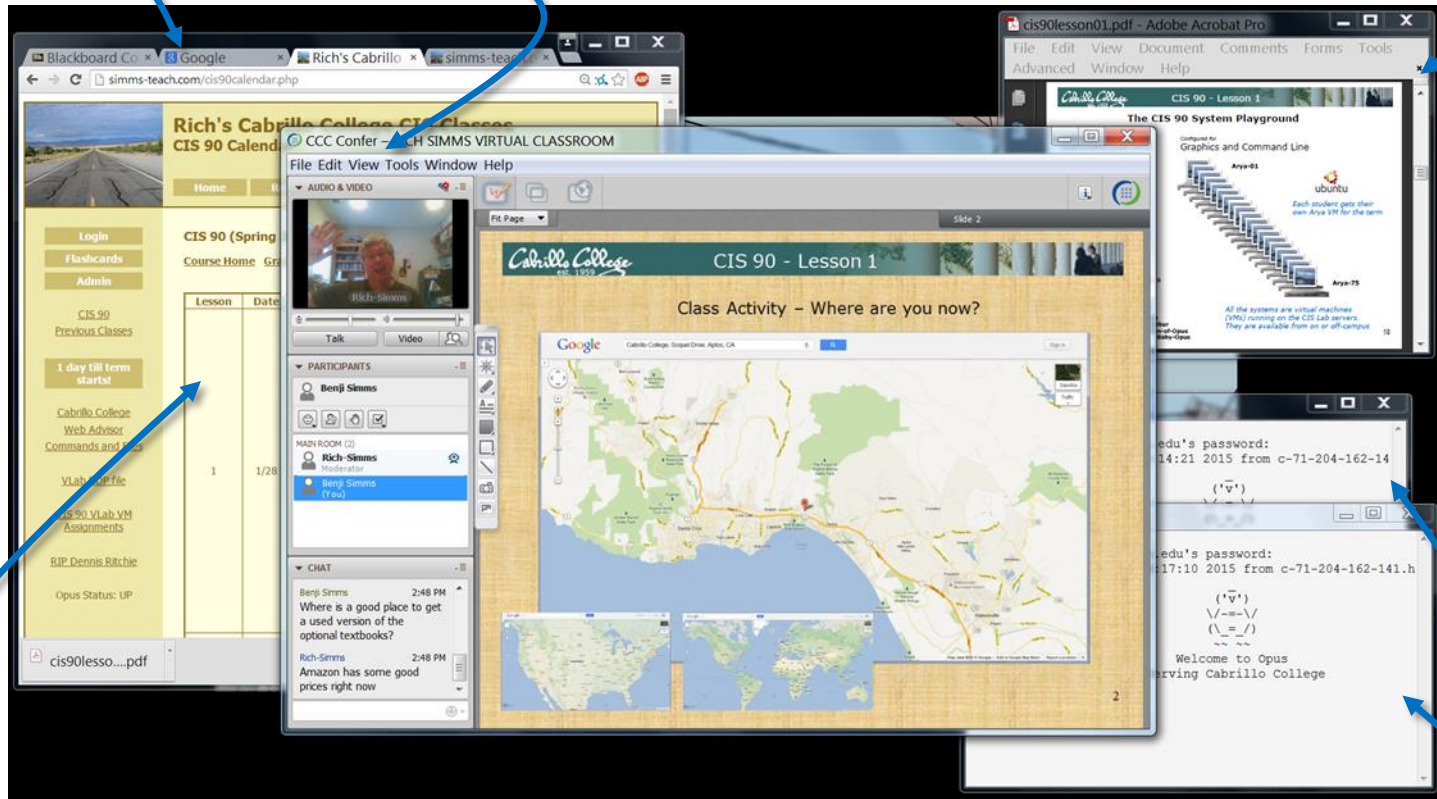


Student checklist for suggested screen layout

☐ Google

☐ CCC Confer

☐ Downloaded PDF of Lesson Slides



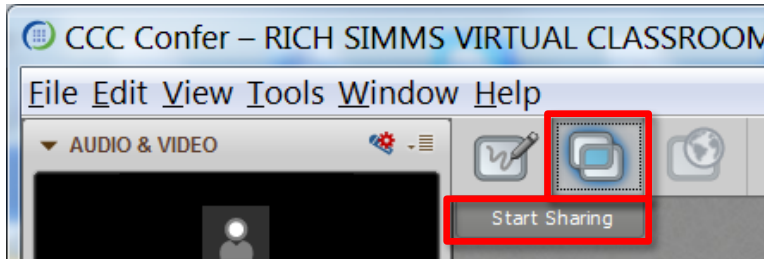
☐ CIS 90 website Calendar page

☐ One or more login sessions to Opus

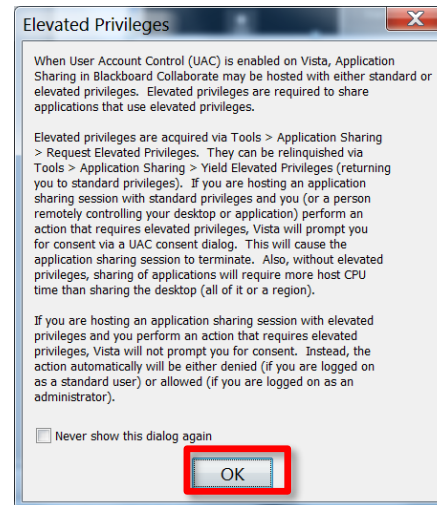


Student checklist for sharing desktop with classmates

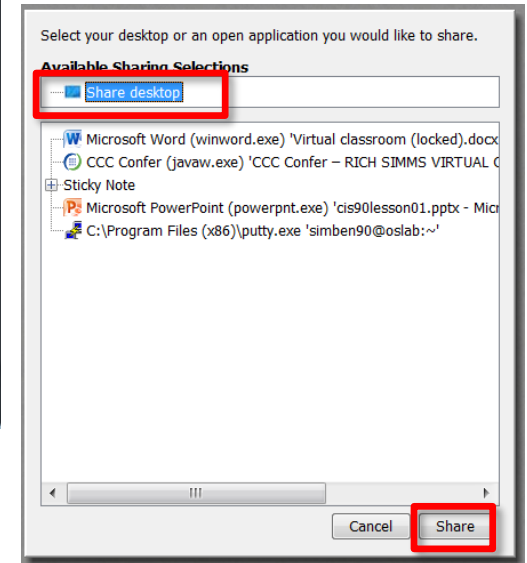
1) Instructor gives you sharing privileges



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



3) Click OK button.



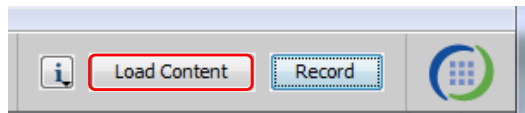
4) Select "Share desktop" and click Share button.



Rich's CCC Confer checklist - setup

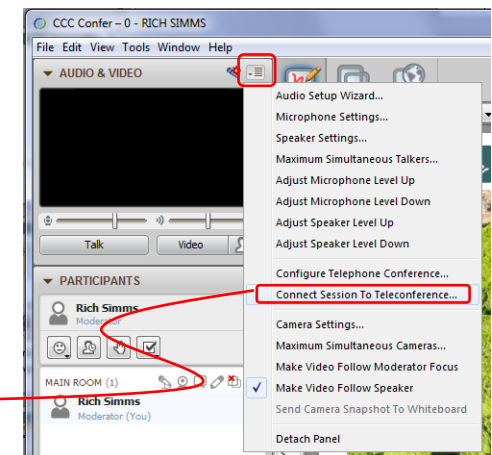
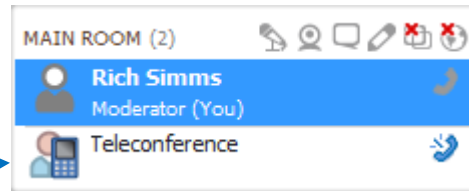


[] Preload White Board



[] Connect session to Teleconference

Session now connected to teleconference



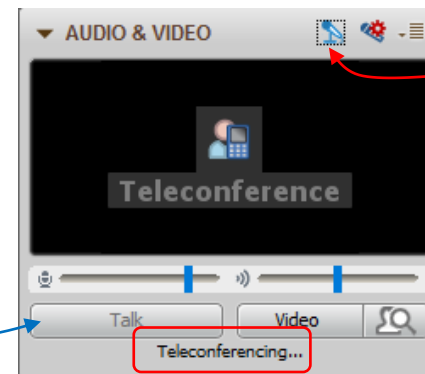
[] Is recording on?



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



The screenshot displays a Windows desktop environment during a CCC Confer session. The desktop includes several open applications:

- CCC Confer - 0 - RIC...:** A window showing the conferencing interface with a video feed of Rich Simms, a list of participants (Rich Simms, Moderator), and a chat window.
- simms-teach.com/docs/cis90/cis-90-TEST-1-Fall-12.pdf:** A web browser window displaying a quiz titled "Part 1 - Flashc (1 point each)". The quiz questions are:
 - [Q1] What command shows the other users logged in to the computer?
 - [A1]
 - [Q2] What environment variable is used by the shell to determine which directories to search when locating a command?
 - [A2]
- Terminal Window:** A terminal window showing a login session for "simben90@oslab:". The output includes:


```
login as: simben90
simben90@oslab.cabrillo.edu's password:
Access denied
simben90@oslab.cabrillo.edu's password:
Last login: Mon Oct 8 18:58:43 2011 from 10.10.10.10
```
- vSphere Client:** A window showing the vSphere Client interface, displaying a list of virtual machines (CIS 192, CIS 191, CIS 190, CIS 189, CIS 188, CIS 187, CIS 186, CIS 185, CIS 184, CIS 183, CIS 182, CIS 181, CIS 180, CIS 179, CIS 178, CIS 177, CIS 176, CIS 175, CIS 174, CIS 173, CIS 172, CIS 171, CIS 170, CIS 169, CIS 168, CIS 167, CIS 166, CIS 165, CIS 164, CIS 163, CIS 162, CIS 161, CIS 160, CIS 159, CIS 158, CIS 157, CIS 156, CIS 155, CIS 154, CIS 153, CIS 152, CIS 151, CIS 150, CIS 149, CIS 148, CIS 147, CIS 146, CIS 145, CIS 144, CIS 143, CIS 142, CIS 141, CIS 140, CIS 139, CIS 138, CIS 137, CIS 136, CIS 135, CIS 134, CIS 133, CIS 132, CIS 131, CIS 130, CIS 129, CIS 128, CIS 127, CIS 126, CIS 125, CIS 124, CIS 123, CIS 122, CIS 121, CIS 120, CIS 119, CIS 118, CIS 117, CIS 116, CIS 115, CIS 114, CIS 113, CIS 112, CIS 111, CIS 110, CIS 109, CIS 108, CIS 107, CIS 106, CIS 105, CIS 104, CIS 103, CIS 102, CIS 101, CIS 100, CIS 99, CIS 98, CIS 97, CIS 96, CIS 95, CIS 94, CIS 93, CIS 92, CIS 91, CIS 90, CIS 89, CIS 88, CIS 87, CIS 86, CIS 85, CIS 84, CIS 83, CIS 82, CIS 81, CIS 80, CIS 79, CIS 78, CIS 77, CIS 76, CIS 75, CIS 74, CIS 73, CIS 72, CIS 71, CIS 70, CIS 69, CIS 68, CIS 67, CIS 66, CIS 65, CIS 64, CIS 63, CIS 62, CIS 61, CIS 60, CIS 59, CIS 58, CIS 57, CIS 56, CIS 55, CIS 54, CIS 53, CIS 52, CIS 51, CIS 50, CIS 49, CIS 48, CIS 47, CIS 46, CIS 45, CIS 44, CIS 43, CIS 42, CIS 41, CIS 40, CIS 39, CIS 38, CIS 37, CIS 36, CIS 35, CIS 34, CIS 33, CIS 32, CIS 31, CIS 30, CIS 29, CIS 28, CIS 27, CIS 26, CIS 25, CIS 24, CIS 23, CIS 22, CIS 21, CIS 20, CIS 19, CIS 18, CIS 17, CIS 16, CIS 15, CIS 14, CIS 13, CIS 12, CIS 11, CIS 10, CIS 9, CIS 8, CIS 7, CIS 6, CIS 5, CIS 4, CIS 3, CIS 2, CIS 1, CIS 0).

Red callout boxes with arrows point to specific elements:

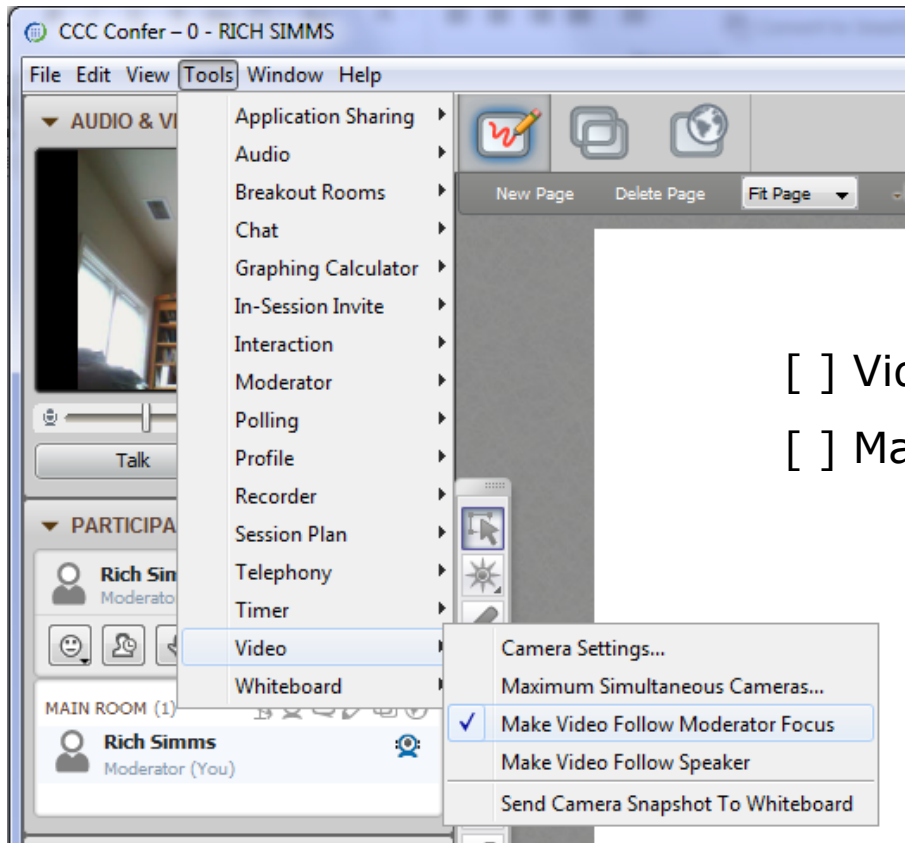
- foxit for slides:** Points to the "cis90lesson07.pdf" window.
- chrome:** Points to the "simms-teach.com" browser window.
- putty:** Points to the terminal window.
- vSphere Client:** Points to the vSphere Client window.

[] layout and share apps





Rich's CCC Confer checklist - webcam setup

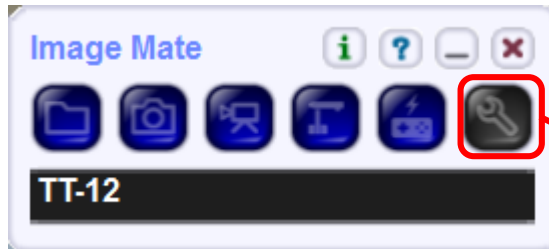


[] Video (webcam)

[] Make Video Follow Moderator Focus



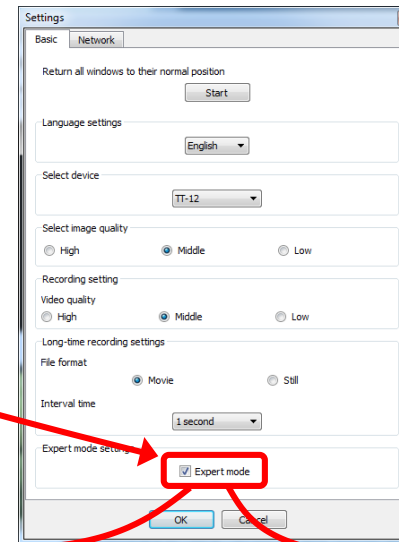
Rich's CCC Confer checklist - Elmo



Elmo rotated down to view side table



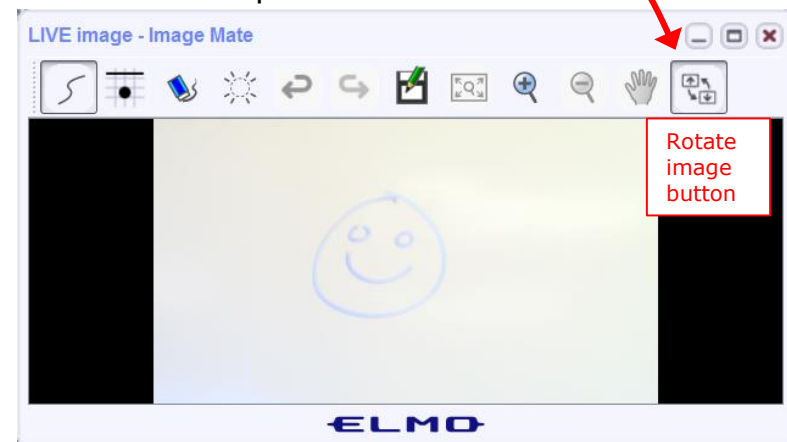
Run and share the Image Mate program just as you would any other app with CCC Confer



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 up to view white board



Rich's CCC Confer checklist - universal fixes

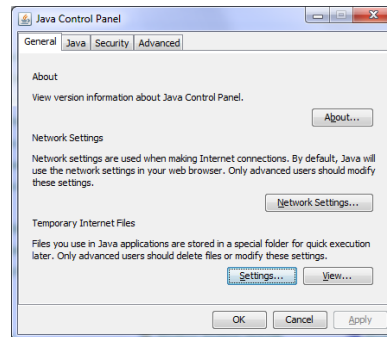
Universal Fix for CCC Confer:

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

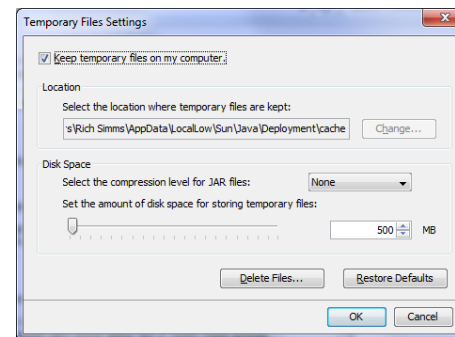
Control Panel (small icons)



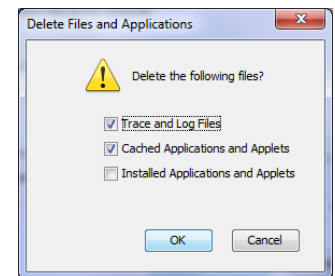
General Tab > Settings...



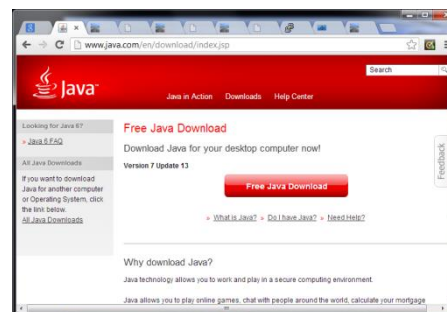
500MB cache size



Delete these

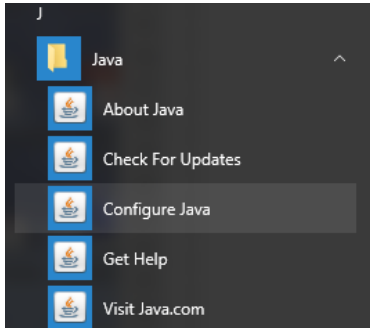


Google Java download

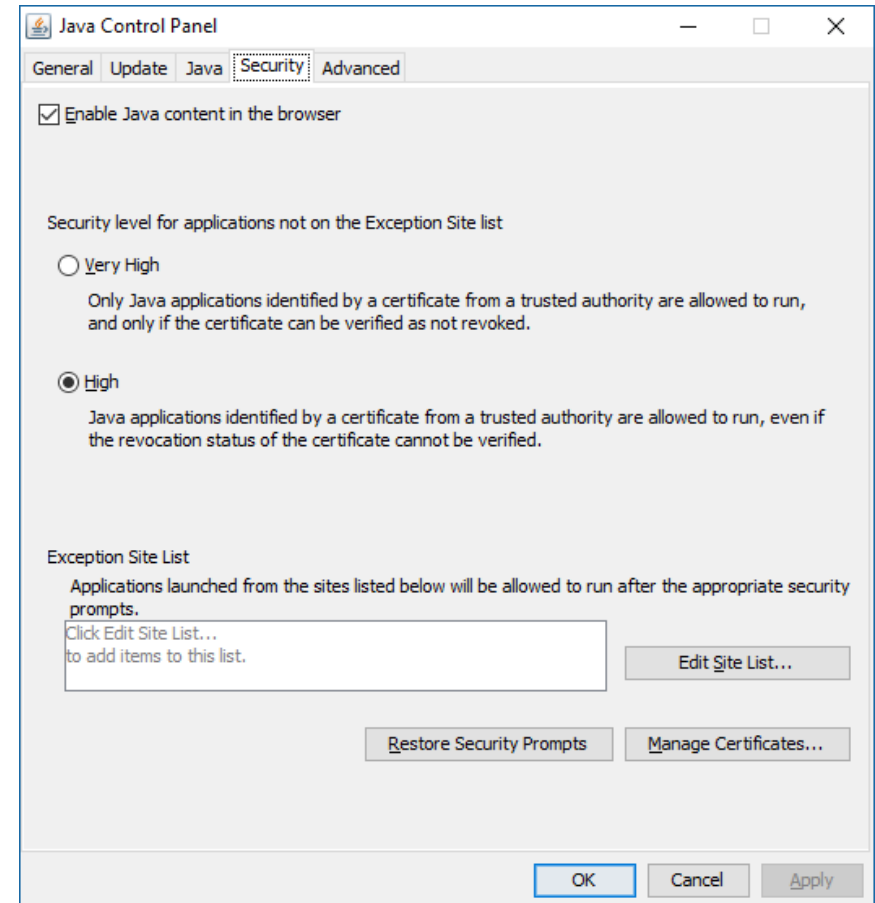




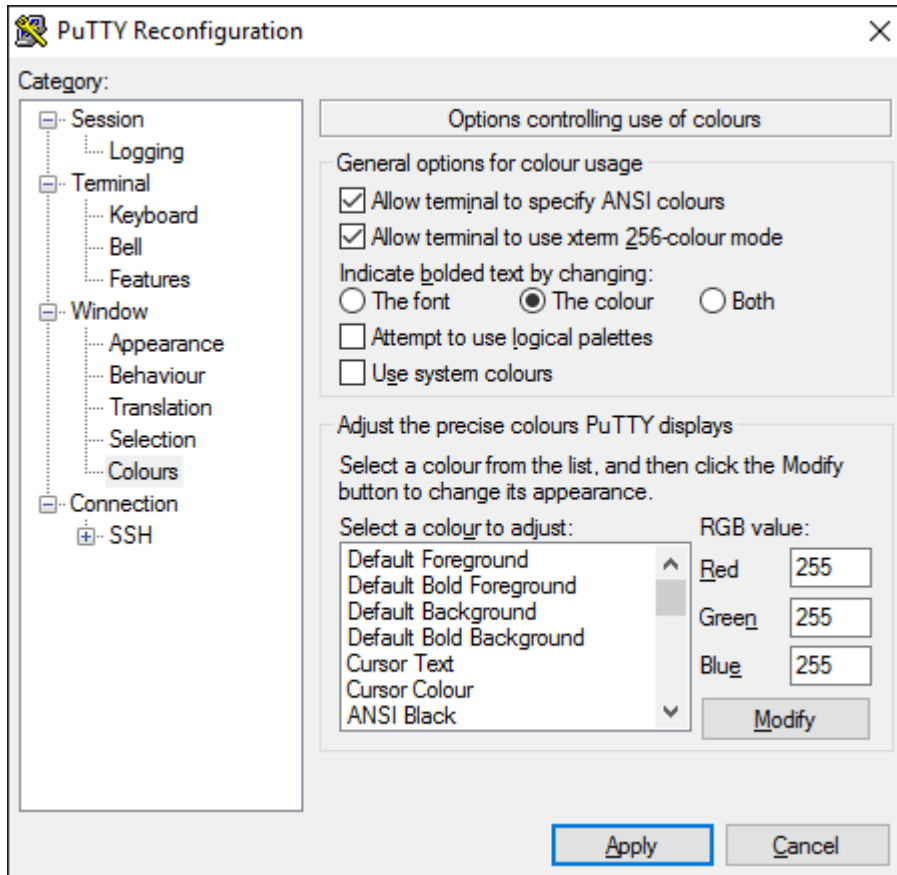
Rich's CCC Confer checklist - digital certificate work around



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.illuminate.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



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

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



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**



Melissa



Tess



Philip



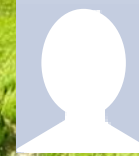
Daniel



Ian J.



Harold



Steven P.



Jasen



Sam



Ryan



Roberto



Hans



Cristian



Nicholas



Nigel



Luis



Stephen L.



Alison



Gracie



Cameron



Julian



Ken



Ian C.



Samantha



Justin

Quiz

**No Quiz
Today !**

More Shell Scripting

Objectives

- Transfer files between computers
- Archive files using tar
- Learn some scripting techniques

Agenda

- No Quiz
- Questions
- Raspberry Pi demos
- ssh and scp
- tar
- tar + scp
- Housekeeping
- Refresh on shell scripts
- Project
- Scripting tips - vi
- Scripting tips - sleep
- Scripting tips \$(cmd) and `cmd`
- Scripting tips - field extraction
- Scripting tips - simple if
- Scripting tips - or logic
- Scripting tips - and logic
- Scripting tips - file types
- Scripting tips - if-then-else
- Scripting tips - set command
- Scripting tips - color
- Scripting tips - username <-> home directory
- Scripting tips - simple for loop
- Assignment
- Wrap up



Questions



Questions?

Lesson material?

Labs? Tests?

How this course works?

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

Who questions much, shall learn much, and retain much.

- Francis Bacon

If you don't ask, you don't get.

- Mahatma Gandhi

Chinese
Proverb

他問一個問題，五分鐘是個傻子，他不問一個問題仍然是一個傻瓜永遠。

He who asks a question is a fool for five minutes; he who does not ask a question remains a fool forever.



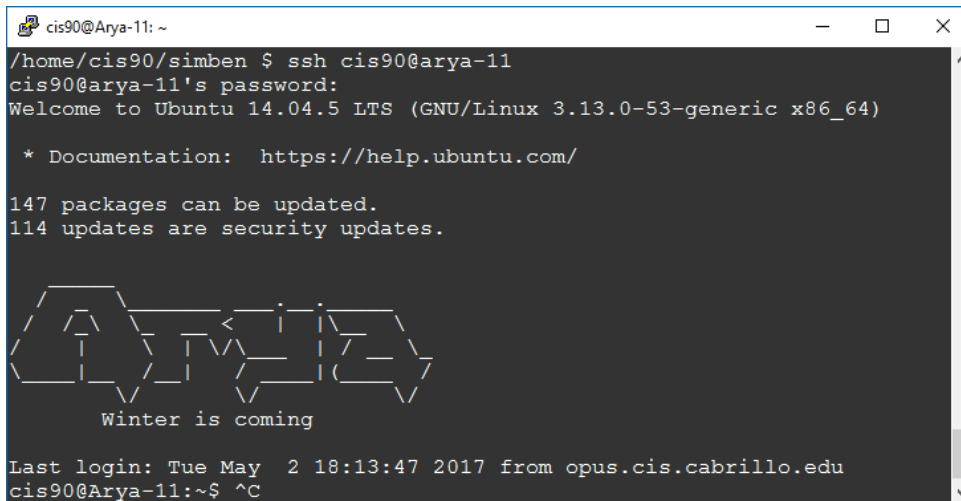
More on ssh

Running a command on a
remote system

Did you know?

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

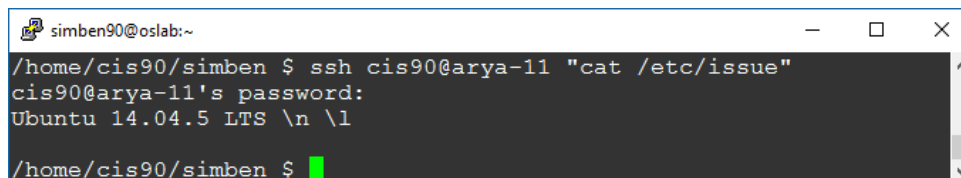
ssh cis90@arya-11



```
cis90@Arya-11: ~  
/home/cis90/simben $ ssh cis90@arya-11  
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/  
  
147 packages can be updated.  
114 updates are security updates.  
  
Winter is coming  
  
Last login: Tue May  2 18:13:47 2017 from opus.cis.cabrillo.edu  
cis90@Arya-11:~$ ^C
```

*This ssh command
logs you into arya-11*

ssh cis90@arya-11 "cat /etc/issue"



```
simben90@oslab:~  
/home/cis90/simben $ ssh cis90@arya-11 "cat /etc/issue"  
cis90@arya-11's password:  
Ubuntu 14.04.5 LTS \n \l  
  
/home/cis90/simben $ █
```

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

Log into your Arya VM using ssh

All these work from Opus:

```
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-11
```

Log into your own Arya VM

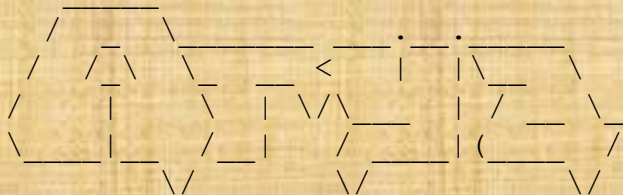
```
cis90@arya-11's password:
```

```
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
```

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

```
Last login: Sun Mar 12 18:01:01 2017 from opus.cis.cabrillo.edu
```

```
cis90@Arya-11:~$
```

From Arya run a remote command on Opus

Example 1

*This who command
will be run on Opus*

```
cis90@Arya-11:~$ ssh simben90@opus "who -Hu"
```

```
simben90@opus'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*

```
cis90@Arya-11:~$ opusUsers=$(ssh simben90@opus "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

```
/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 private key (NEVER EVER share with anyone)
 Your public key (can share with anyone)

Look Ma, no password

Opus

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

```
simben90@oslab:~$ cat .ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAvmnz5vPGjArMf6d4anLykYXs0WlW9hV3Zs0XIq
JSklmrz4dqEytP4BUA22HD3b06pQzi7NYlbnN1iJF1AJoU/pc2QU8voEljnpFisGQ/903bsnN
G9v0600371zTcySrNxae/oV0p3+eF0Zepu/8E/ntZhqWWvZbmQWVV4VHbV4L8gVj9K52ysHs0T
9Hx4VfGaTYbhHbf3GaIGQKXMMz1leG8oaHTpOBubuBEqKnNcLYyFuCPJ5tgQMD00VSx0ZRPVJe
v0DJYrKmEKffszSuFTKulh6D+pxaLWtSBYV/3Z4qnz5qy+SAqpgV4FeRpXlE1c8Qh+de/rIHXA
qROEghDw== simben90@oslab.cis.cabrillo.edu
/home/cis90/simben $
```

Arya

```
cis90@Arya-11:~$ mkdir .ssh
cis90@Arya-11:~$ chmod 700 .ssh
cis90@Arya-11:~$ vi .ssh/authorized_keys
```

```
cis90@Arya-03: ~$ cat .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAvmnz5vPGjArMf6d4anLykYXs0WlW9hV3Zs0XIq
JSklmrz4dqEytP4BUA22HD3b06pQzi7NYlbnN1iJF1AJoU/pc2QU8voEljnpFisGQ/903bsnN
G9v0600371zTcySrNxae/oV0p3+eF0Zepu/8E/ntZhqWWvZbmQWVV4VHbV4L8gVj9K52ysHs0T
9Hx4VfGaTYbhHbf3GaIGQKXMMz1leG8oaHTpOBubuBEqKnNcLYyFuCPJ5tgQMD00VSx0ZRPVJe
v0DJYrKmEKffszSuFTKulh6D+pxaLWtSBYV/3Z4qnz5qy+SAqpgV4FeRpXlE1c8Qh+de/rIHXA
qROEghDw== simben90@oslab.cis.cabrillo.edu
-- INSERT -- 1,413 All
```

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

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-11
```

Log into your own Arya VM

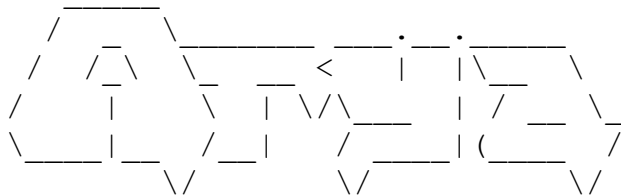
```
cis90@arya-11's password:
```

```
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

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

```
Last login: Sat Feb 21 18:23:19 2015 from opus.cis.cabrillo.edu
```

```
cis90@Arya-11:~$
```

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

```
ssh -p 22 cis90@arya-03
```

```
ssh -p 22 cis90@arya-03.cis.cabrillo.edu
```

Copy one file from Opus

Syntax:

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

```
cis90@Arya-11:~$ scp simben90@opus:letter .  
simben90@opus'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@oslab.cis.cabrillo.edu:letter .  
scp -P 2220 simben90@oslab.cis.cabrillo.edu:letter letter
```

Use your own Opus username and password when trying this

Copy several files from Opus

Syntax:

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

```
cis90@Arya-11:~$ scp simben90@opus:poems/Shakespeare/sonnet* .
simben90@opus's password:
sonnet1                100% 614      0.6KB/s   00:00
sonnet10               100% 620      0.6KB/s   00:00
sonnet11               100% 689      0.7KB/s   00:00
sonnet15               100% 618      0.6KB/s   00:00
sonnet17               100% 647      0.6KB/s   00:00
sonnet2                100% 631      0.6KB/s   00:00
sonnet26               100% 601      0.6KB/s   00:00
sonnet3                100% 615      0.6KB/s   00:00
sonnet35               100% 598      0.6KB/s   00:00
sonnet4                100% 588      0.6KB/s   00:00
sonnet5                100% 622      0.6KB/s   00:00
sonnet7                100% 581      0.6KB/s   00:00
sonnet9                100% 620      0.6KB/s   00:00
cis90@Arya-11:~$
```

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

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

Use your own Opus username and password when trying this

Copy (recursively) an entire file tree branch from Opus

Syntax:

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

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

```
simben90@opus's password:
```

```
sonnet10      100% 620    0.6KB/s  00:00
sonnet15      100% 618    0.6KB/s  00:00
sonnet26      100% 601    0.6KB/s  00:00
sonnet3       100% 615    0.6KB/s  00:00
sonnet35      100% 598    0.6KB/s  00:00
sonnet2       100% 631    0.6KB/s  00:00
sonnet4       100% 588    0.6KB/s  00:00
sonnet1       100% 614    0.6KB/s  00:00
.1979.egg     100% 733    0.7KB/s  00:00
sonnet11      100% 689    0.7KB/s  00:00
sonnet7       100% 581    0.6KB/s  00:00
sonnet5       100% 622    0.6KB/s  00:00
sonnet9       100% 620    0.6KB/s  00:00
sonnet17      100% 647    0.6KB/s  00:00
mooncat       100% 856    0.8KB/s  00:00
1982.egg      100% 134    0.1KB/s  00:00
whitebirds    100% 863    0.8KB/s  00:00
old           100% 520    0.5KB/s  00:00
1978.egg      100% 734    0.7KB/s  00:00
nursery       100% 779    0.8KB/s  00:00
ant           100% 237    0.2KB/s  00:00
twilight      100% 654    0.6KB/s  00:00
artichoke     100% 1436   1.4KB/s  00:00
dog           100% 1842   1.8KB/s  00:00
.1983.egg     100% 734    0.7KB/s  00:00
twister       100% 151    0.2KB/s  00:00
bird          100% 975    1.0KB/s  00:00
woman         100% 1273   1.2KB/s  00:00
1984.egg      100% 404    0.4KB/s  00:00
you           100% 236    0.2KB/s  00:00
diner         100% 741    0.7KB/s  00:00
eden          100% 189    0.2KB/s  00:00
hope          100% 343    0.3KB/s  00:00
charm         100% 203    0.2KB/s  00:00
forget        100% 228    0.2KB/s  00:00
.1988.egg     100% 405    0.4KB/s  00:00
tiger         100% 115    0.1KB/s  00:00
1991.egg      100% 725    0.7KB/s  00:00
jerusalem     100% 582    0.6KB/s  00:00
cis90@Arya-03:
```

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

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

Use your own Opus username and password when trying this

tar

tar command

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

Basic tar command syntax

verbose
specify the archive file

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

creates an archive

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

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

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

extracts 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 $
```

*create
verbose
file*

*name of
archive file
(tarball)*

*pathname
to directory
to archive*

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

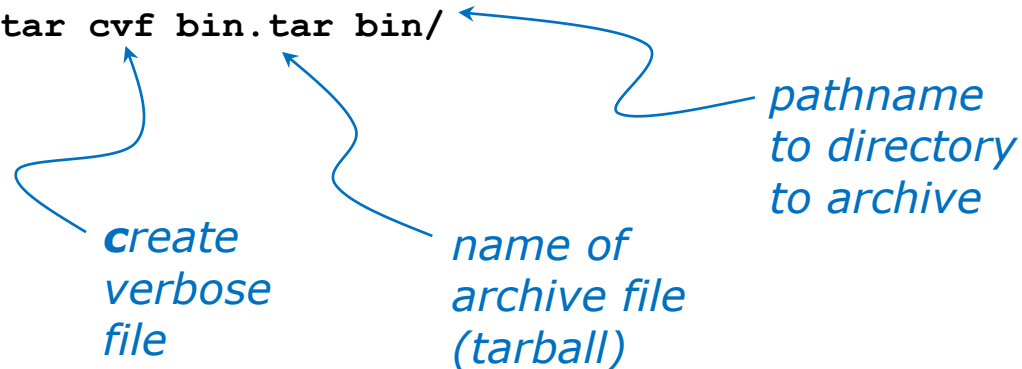
```
/home/cis90/simben $ ls bin
```

```
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 $
```



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

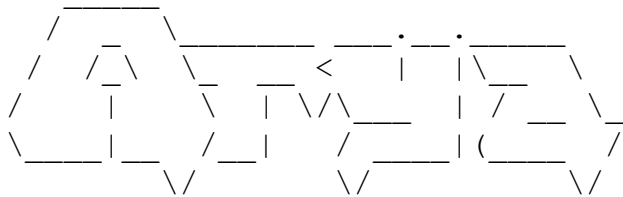
username → *hostname*

```
/home/cis90/simben $ ssh cis90@arya-11
cis90@arya-11'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-11:~$
```

*Login to your
own Arya VM
from Opus*

Example

Copy archived directory to another system

username *hostname* *path to tar file*
port

```
cis90@arya-11:~$ scp -P 2220 simben90@oslab.cis.cabrillo.edu:bin.tar .
simben90@oslab.cis.cabrillo.edu's password:
bin.tar                                100%   40KB   40.0KB/s
00:00
```

"here"

```
cis90@Arya-11:~$ ls -l bin.tar
-rw-rw---- 1 cis90 cis90 40960 Dec  2 07:52 bin.tar
cis90@Arya-11:~$
```

*Note how
archive files are
shown in red*

Copy your bin archive from Opus to Arya

Example

Copy archived directory to another system

```
cis90@Arya-11:~$ 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-03:~$
```

*extract
verbose
file*

*name of
archive file
(tarball)*

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

```
cis90@Arya-11:~$ ls bin
```

```
app      datecal      hi      I      myscript.v1  tryme  
banner  enlightenment  home  myscript  treed      zoom  
cis90@Arya-11:~$
```

Example

Copy archived directory to another system

```
cis90@Arya-11:~$ myscript
No command 'myscript' found, did you mean:
  Command 'pyscript' from package 'python-pyscript' (universe)
myscript: command not found
cis90@Arya-11:~$

cis90@Arya-11:~$ 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-11:~$ cd bin
cis90@Arya-11:~/bin$ ./myscript
/home/cis90/bin/myscript: line 44: finger: command not found
What is your first name? ^C
cis90@Arya-11:~$
```

 *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

```
cis90@Arya-03:~$ 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-03:~$
```

*Use sudo to install
finger as the root
superuser*

Example

Copy archived directory to another system

Run myscript file in the bin directory

```
cis90@Arya-03:~/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

phpBB®
creating communities

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

Search...  

 Quick links  FAQ

 Register  Login

 Board index < Cabrillo College Spring 2017 Courses < CIS 90 - Spring 2017

CIS Internship Interest Form

  Search this topic...  

1 post • Page 1 of 1

CIS Internship Interest Form

by **Rich Simms** » Tue May 02, 2017 11:26 am

Just got this from Gerlinde and Matt. They work with local businesses and organizations to arrange internships for Cabrillo students. If you are interested in an intern position please use this form:

<https://docs.google.com/a/cabrillo.edu/...rm?c=0&w=1>

- Rich



Rich Simms
Posts: 2364
Joined: Sat Jan 16, 2010 5:47 pm
Contact: 

1 post • Page 1 of 1

 Return to "CIS 90 - Spring 2017"

Jump to 

WHO IS ONLINE
Users browsing this forum: harold grieco and 2 guests

 Board index

 Contact us  The team  Delete all board cookies • All times are UTC-07:00

Powered by phpBB® Forum Software © phpBB Limited

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

Next Class

**Project is due
next week!**



1. No labs due today.
2. There is a check script for Lab X2.
3. There is no check script for Lab X1. To test permissions copy it to the /tmp directory and run it using the cis90 user account.
4. 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 May 15th 7-9:50AM**

Mon	5/15	Test #3 (the final exam) Time - MONDAY 7:00AM - 9:50AM in Room 828 Materials - Test (canvas) CCC Confer Enter virtual classroom Class archives		5 posts Lab X1 Lab X2
------------	------	---	--	--

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

- All students will take the test at the same time. The test must be completed by **9:50AM**.
- Working and long distance students can take the test online via CCC Confer and 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)

FINAL EXAMINATIONS SCHEDULE: SPRING 2017, May 15 to May 20

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 their 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	Wednesday, May 17
9:00 am and 10:15 am, MW/Daily	7:00 am-9:50 am	Monday, May 15
10:20 am and 11:35 am, MW/Daily	10:00 am-12:50 pm	Wednesday, May 17
11:40 am and 12:55 pm, MW/Daily	10:00 am-12:50 pm	Monday, May 15
1:00 pm and 2:15 pm, MW/Daily	1:00 pm-3:50 pm	Wednesday, May 17
2:20 pm and 3:35 pm, MW/Daily	1:00 pm-3:50 pm	Monday, May 15
3:40 pm and 5:30 pm, MW/Daily	4:00 pm-6:50 pm	Monday, May 15
6:30 am and 8:55 am, TTh	7:00 am-9:50 am	Monday, May 15
9:00 am and 10:15 am, TTh	7:00 am-9:50 am	Monday, May 15
10:20 am and 11:35 am, TTh	10:00 am-12:50 pm	Monday, May 15
11:40 am and 12:55 pm, TTh	10:00 am-12:50 pm	Monday, May 15
1:00 pm and 2:15 pm, TTh	1:00 pm-3:50 pm	Monday, May 15
2:20 pm and 3:35 pm, TTh	1:00 pm-3:50 pm	Monday, May 15
3:40 pm and 5:30 pm, TTh	4:00 pm-6:50 pm	Monday, May 15
Friday am	9:00 am-11:50 am	Monday, May 15
Friday pm	1:00 pm-3:50 pm	Monday, May 15
Saturday am	9:00 am-11:50 am	Monday, May 15
Saturday pm	1:00 pm-3:50 pm	Monday, May 15

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
95746	W	9:00AM-12:05PM	3.00	R.Simms	OL
&	Arr.	Arr.		R.Simms	OL

Section 95746 is an ONLINE course. Meets weekly throughout the semester online during the scheduled times by remote technology with an additional 50 min arranged online lab per week. For details, see instructor's web page at go.cabrillo.edu/online.

95747	W	9:00AM-12:05PM	3.00	R.Simms	828
&	Arr.	Arr.		R.Simms	OL

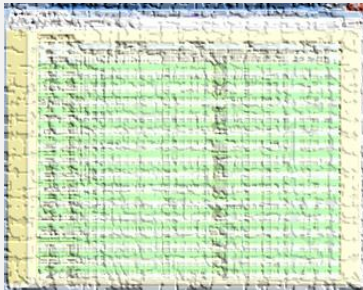
Section 95747 is a Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 50 min online lab per week. For details, see instructor's web page at go.cabrillo.edu/online.

Where to find your grades

Send me your survey to get your LOR code name.

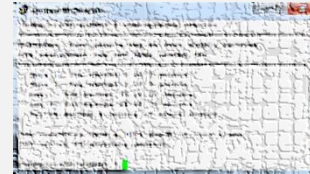
The CIS 90 website Grades page

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



Or check on Opus

checkgrades *codename*
(where codename is your LOR codename)



Written by Jesse Warren a past CIS 90 Alumnus

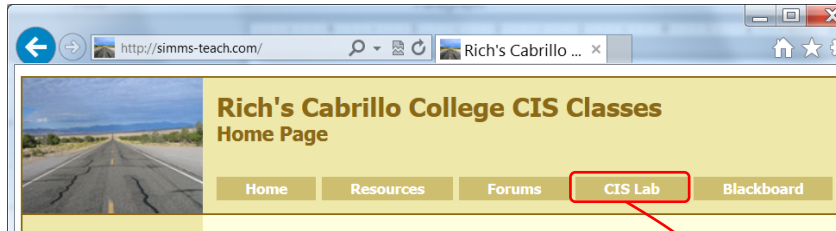
Percentage	Total Points	Letter Grade	Pass/No Pass
90% or higher	504 or higher	A	Pass
80% to 89.9%	448 to 503	B	Pass
70% to 79.9%	392 to 447	C	Pass
60% to 69.9%	336 to 391	D	No pass
0% to 59.9%	0 to 335	F	No pass

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

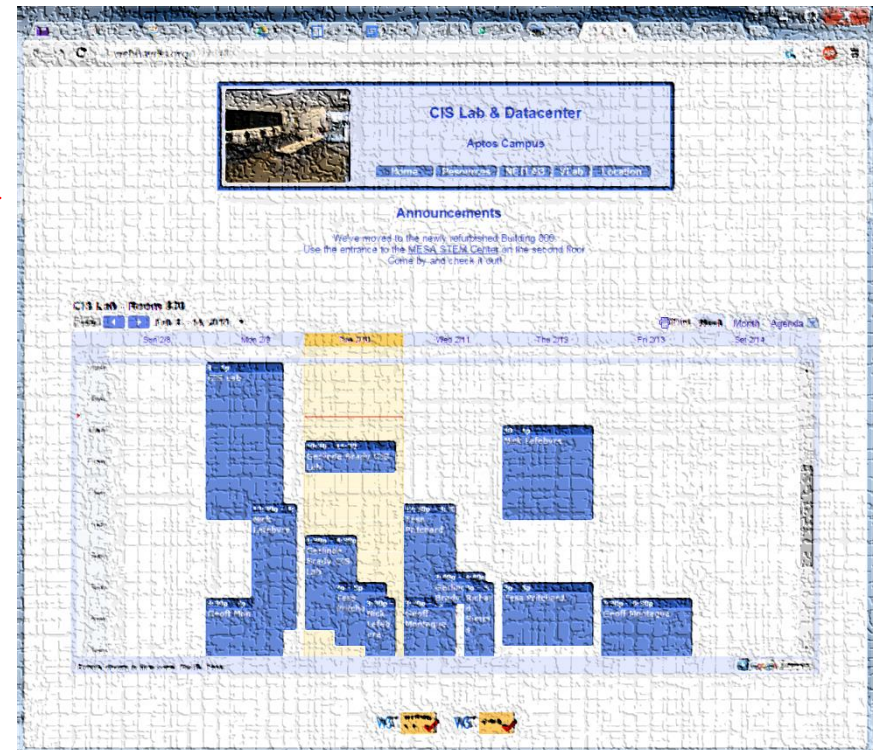
Points that could have been earned:

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

Stuck on something or just like some extra help?



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

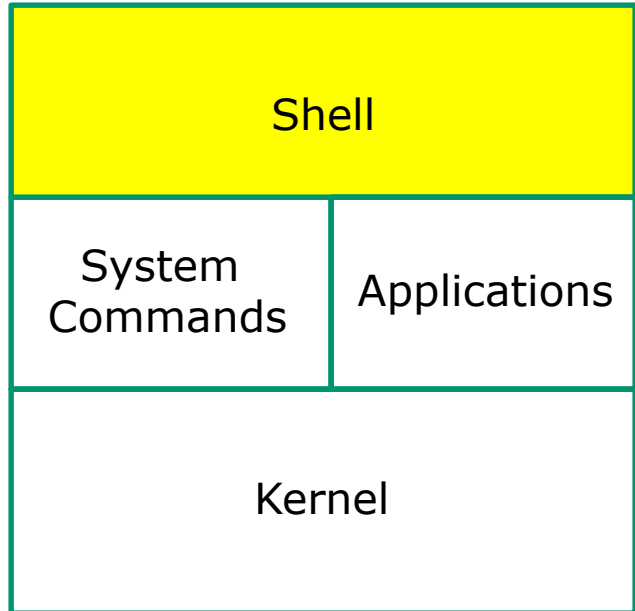




Refresh

UNIX/Linux Architecture

The Shell



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



Shell Scripts

Some scripts on opus

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

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

Many commands are scripts

Which commands in /bin are really scripts?

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

How many commands in /bin are scripts?

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

Class Activity

Scripting

Of all the UNIX/Linux commands in:

/bin

/usr/bin

/sbin

/usr/sbin

How many are scripts?

Write your answer in the chat window

Project

Get started on the project!

(If you haven't already)

1. Create a file in your bin directory named *myscript*:
 - Copy from `/home/cis90/depot/myscript`
 - or copy and paste template code from:
<http://simms-teach.com/docs/cis90/cis90final-project.pdf>
2. Give yourself full permissions and give CIS 90 group read and execute permissions
 - **chmod 750 myscript**
3. Run **allscripts** and verify your script will run without any errors
4. Do the example grep task shown in Lesson 13

Possible Points	Requirements
30	Implementing all five tasks (6 points each): - Requirements for each task: - Minimum of 10 "original" script command lines - Has one or more non-generic comments to explain what it is doing - Has user interaction
25	You don't have to do all of these but do at least five: - Redirecting stdin (5 points) - Redirecting stdout (5 points) - Redirecting stderr (5 points) - Use of permissions (5 points) - Use of filename expansion characters (5 points) - Use of absolute path (5 points) - Use of relative path (5 points) - Use of a PID (5 points) - Use of inodes (5 points) - Use of links (5 points) - Use of color (5 points) - Use of scheduling (5 points) - Use of a GID or group (5 points) - Use of a UID or user (5 points) - Use of a /dev/tty device (5 points) - Use of a signal (5 points) - Use of piping (5 points) - Use of an environment variable (5 points) - Use of /bin/mail (5 points) - Use of a conditional (5 points) The maximum for this section is 25 points.
5	Present your script to the class
Points lost	
-15	Fails to run from allscripts
-15	Other students in the class are unable to read and execute your script.
-15	Error messages are displayed when running one or more tasks
-up to 90	No credit for any task which contains unoriginal script code that: - Doesn't give full credit to the original author. - Doesn't indicate where the code was obtained from. - Doesn't include licensing terms. - Violates copyright or licensing terms.
-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@oslab:~  
*****  
*           Spring 2017 CIS 90 Online Projects           *  
*****  
1) Alison  
2) Benji  
3) Cameron  
4) Cristian  
5) Daniel H.  
6) Dillon  
7) Duke  
8) Gracie  
9) Hans  
10) Harold  
11) Homer  
12) Ian C.  
13) Ian J.  
14) Jasen  
15) Joshua  
16) Julian  
17) Justin  
18) Ken  
19) Luis  
20) Melissa  
21) Nicholas  
22) Nigel  
23) Philip  
24) Roberto  
25) Ryan  
26) Samantha  
27) Sam  
28) Stephen  
29) Steven  
30) Tess  
  
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: • Doesn't give full credit to the original author. • Doesn't indicate where the code was obtained from. • Doesn't include licensing terms. • Violates copyright or licensing terms.
-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 as the cis90 user and confirm

Project Status

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

```
rsimms@oslab:~$ ls -l /home/cis90/*/bin/myscript
-rwxr-x---. 1 atirob90 cis90 2252 May  2 14:56 /home/cis90/atirob/bin/myscript
-rwxrwxr-x. 1 blaali90 cis90  729 Apr 19 10:27 /home/cis90/blaali/bin/myscript
-rwxr-x---. 1 dixken90 cis90 1147 Apr 30 17:35 /home/cis90/dixken/bin/myscript
-rwxr-x---. 1 flyjul90 cis90  852 Apr 19 10:49 /home/cis90/flyjul/bin/myscript
-rwxr-x---. 1 hargra90 cis90  737 May  1 13:15 /home/cis90/hargra/bin/myscript
-rwxrwxr-x. 1 hugmel90 cis90  706 Apr 19 10:34 /home/cis90/hugmel/bin/myscript
-rwxrwxrwx. 1 johian90 cis90 2624 May  2 07:14 /home/cis90/johian/bin/myscript
-rwxr-x---. 1 lawphi90 cis90 2754 Apr 29 18:57 /home/cis90/lawphi/bin/myscript
-rwxr-xr-x. 1 lawste90 cis90  934 Apr 28 19:27 /home/cis90/lawste/bin/myscript
-rwxr-x---. 1 leuhan90 cis90 4881 May  1 21:20 /home/cis90/leuhan/bin/myscript
-rwxr-x---. 1 levjas90 cis90  702 Apr 26 11:52 /home/cis90/levjas/bin/myscript
-rwxr-x---. 1 milhom90 cis90 4946 Apr 23 17:55 /home/cis90/milhom/bin/myscript
-rwxr-x---. 1 padnic90 cis90 2545 Apr 28 23:00 /home/cis90/padnic/bin/myscript
-rwxrwxr-x. 1 rodduk90 cis90 1430 Apr 23 17:43 /home/cis90/rodduk/bin/myscript
-rwxr-x---. 1 simben90 cis90 10513 Apr 18 17:25 /home/cis90/simben/bin/myscript
-rwxr-x---. 1 strsam90 cis90 1508 Apr 30 16:24 /home/cis90/strsam/bin/myscript
-rwxr-x---. 1 wilnig90 cis90  933 Apr 26 17:15 /home/cis90/wilnig/bin/myscript
rsimms@oslab ~$
```

Is your script "hackable" by other classmates?

Project Status

```
rsimms@oslab:~  
[rsimms@oslab ~]$ /home/cis90/bin/checkmyscripts  
-rwxr-x---. 1 simben90 cis90 10513 Apr 18 17:25 /home/cis90/simben/bin/myscript  
-rwxr-x---. 1 milhom90 cis90 4946 Apr 23 17:55 /home/cis90/milhom/bin/myscript  
-rwxrwxr-x. 1 rodduk90 cis90 1430 Apr 23 17:43 /home/cis90/rodduk/bin/myscript  
ls: cannot access /home/cis90/coltes/bin/myscript: No such file or directory  
-rwxr-x---. 1 dixken90 cis90 1147 Apr 30 17:35 /home/cis90/dixken/bin/myscript  
ls: cannot access /home/cis90/ezejus/bin/myscript: No such file or directory  
ls: cannot access /home/cis90/grihar/bin/myscript: No such file or directory  
-rwxrwxr-x. 1 hugmel90 cis90 706 Apr 19 10:34 /home/cis90/hugmel/bin/myscript  
-rwxrwxrwx. 1 johian90 cis90 2624 May  2 07:14 /home/cis90/johian/bin/myscript  
-rwxr-x---. 1 lawphi90 cis90 2754 Apr 29 18:57 /home/cis90/lawphi/bin/myscript  
-rwxr-x---. 1 leuhan90 cis90 4881 May  1 21:20 /home/cis90/leuhan/bin/myscript  
-rwxr-x---. 1 strsam90 cis90 1508 Apr 30 16:24 /home/cis90/strsam/bin/myscript  
ls: cannot access /home/cis90/yourya/bin/myscript: No such file or directory  
-rwxr-x---. 1 atirob90 cis90 2252 May  2 14:56 /home/cis90/atirob/bin/myscript  
ls: cannot access /home/cis90/bisjam/bin/myscript: No such file or directory  
-rwxr-x---. 1 flyjul90 cis90 852 Apr 19 10:49 /home/cis90/flyjul/bin/myscript  
ls: cannot access /home/cis90/gomcri/bin/myscript: No such file or directory  
ls: cannot access /home/cis90/gutlui/bin/myscript: No such file or directory  
-rwxr-x---. 1 hargra90 cis90 737 May  1 13:15 /home/cis90/hargra/bin/myscript  
ls: cannot access /home/cis90/howdan/bin/myscript: No such file or directory  
ls: cannot access /home/cis90/kensam/bin/myscript: No such file or directory  
-rwxr-x---. 1 levjas90 cis90 702 Apr 26 11:52 /home/cis90/levjas/bin/myscript  
-rwxr-x---. 1 padnic90 cis90 2545 Apr 28 23:00 /home/cis90/padnic/bin/myscript  
ls: cannot access /home/cis90/pizste/bin/myscript: No such file or directory  
-rwxr-x---. 1 wilnig90 cis90 933 Apr 26 17:15 /home/cis90/wilnig/bin/myscript  
ls: cannot access /home/cis90/croian/bin/myscript: No such file or directory  
-rwxrwxr-x. 1 blaali90 cis90 729 Apr 19 10:27 /home/cis90/blaali/bin/myscript  
-rwxr-xr-x. 1 lawste90 cis90 934 Apr 28 19:27 /home/cis90/lawste/bin/myscript  
[rsimms@oslab ~]$
```

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

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

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

14
Enter a new string to search for

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

```
milhom90@oslab:~/bin

1)  # Task 1 - grep command explored

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

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

    # Prompt user to supply search string and use color
    echo "Enter a new string to search for"
    read string
    echo searching for "'$string'"
    grab -h --color $string /home/cis90/milhom/poems/*/*
    ;;

2)  # Commands for Task 2
    ;;

3)  # Commands for Task 3
    ;;

4)  # Commands for Task 4
```

40,17 38%

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
;;
```

Use color syntax to spot unmatched quotes

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

Color Syntax

```
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
;;
```

```
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
;;
```

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

%s /oldstring/newstring/g

```
rsimms@opus:/home/cis192/depot<!--DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"><html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en"><head><title>Arwen's CIS 192 Lab 10</title></head><body><h1>Arwen's CIS 192 Lab 10</h1><h2>Internet Services</h2><div></div><p>Spring 2009</p><div><a href="http://validator.w3.org/check/referer" style="background-color: transparent"></a><br/><a href="http://jigsaw.w3.org/css-validator/check/referer" style="background-color: transparent"></a></div></body></html>~%s /Arwen/Elrond/g
```

```
rsimms@opus:/home/cis192/depot<!--DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"><html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en"><head><title>Elrond's CIS 192 Lab 10</title></head><body><h1>Elrond's CIS 192 Lab 10</h1><h2>Internet Services</h2><div></div><p>Spring 2009</p><div><a href="http://validator.w3.org/check/referer" style="background-color: transparent"></a><br/><a href="http://jigsaw.w3.org/css-validator/check/referer" style="background-color: transparent"></a></div></body></html>~%s /Arwen/Elrond/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 ticks 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
```

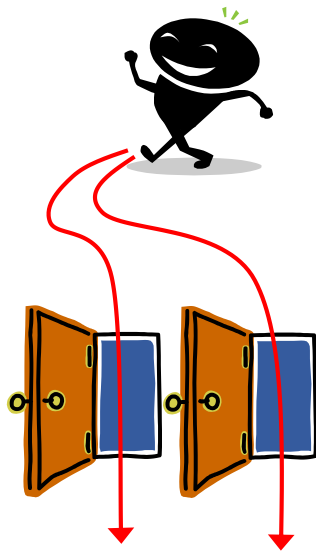


Scripting Tips

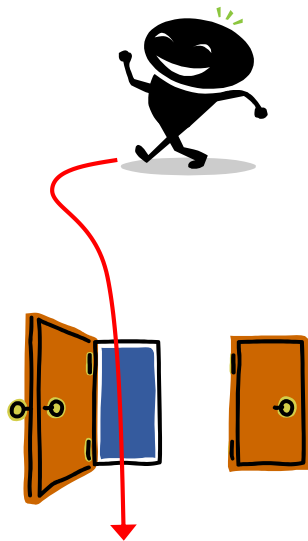
if statement with "or"

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

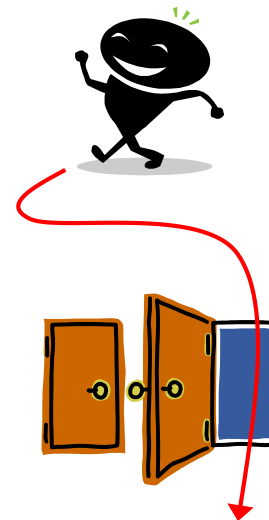
OR logic



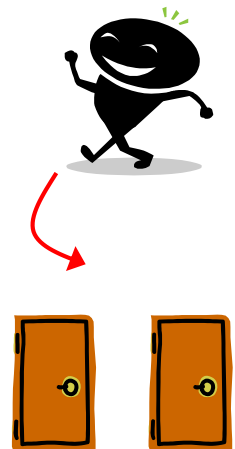
Yes



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
```

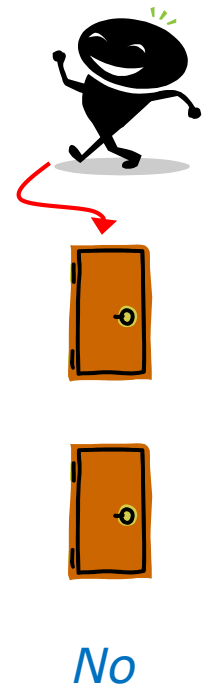
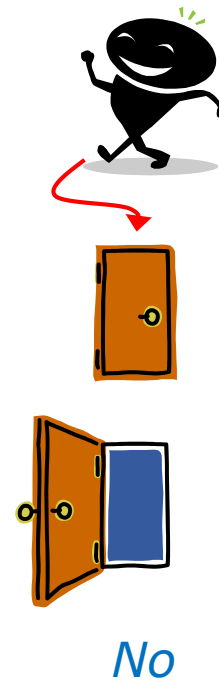
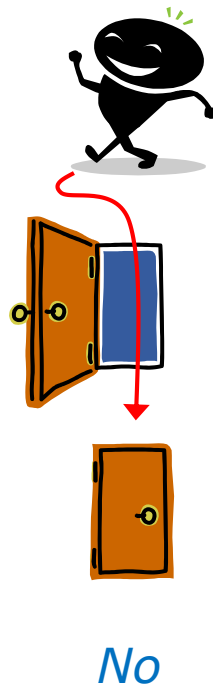
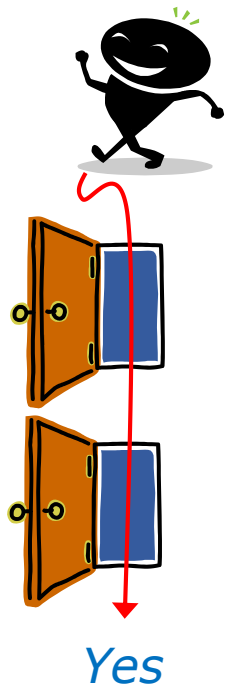


Scripting Tips

if statements with "and"

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

AND logic



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 -l
      echo -n "Which file are you interested in? : "
      read filename

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

*tests to see
if it's a
regular file*

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

*tests to see
if it's a
directory*

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

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.

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 moogleg.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

Using the set command

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

```
[rsimms@opus scripts]$ echo $1  
dogs
```

```
[rsimms@opus scripts]$ echo $2  
cats
```

```
[rsimms@opus scripts]$ echo $3  
birds
```

```
[rsimms@opus scripts]$ echo $4  
humans
```

```
[rsimms@opus scripts]$ echo $#  
4
```

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

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

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

*\$# is set to the total number
of arguments.*

```
[rsimms@opus bin]$ echo $(ls)  
1975.egg app banner datecal enlightenment hi I myscript myscript.milhom90  
myscript.v1 newsript 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 $5"  
The fifth file in this directory is enlightenment  
[rsimms@opus bin]$
```

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

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

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

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

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

```
[rsimms@opus scripts]$ echo $2
rsimms
```

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

```
[rsimms@opus scripts]$ echo $4
Rich
```

```
[rsimms@opus scripts]$ echo $5
Simms
```

```
[rsimms@opus scripts]$ firstname=$4
```

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

*Another way to
get a user's first
name*

Class Exercise

Make a new script in your bin directory

```
cd bin
```

```
vi example777
```

In vi add these lines to your script then save:

```
set $(finger $LOGNAME | head -1)
```

```
firstname=$4
```

```
echo My first name is $firstname
```

Prepare and run your script

```
chmod +x example777
```

```
example777
```



Scripting Tips

color

Using Color

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

```
/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[0n;nnm"** to turn on color and **\e[00m** to turn it off.*

(the -e option enables interpretation of backslash escapes)

Using Color

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

*Change to
color green*

```
/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'
```

*Revert color
back to normal*

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

Using Color

```
simben90@oslab:~/bin
/home/cis90/simben/bin $ off="\e[00m"
/home/cis90/simben/bin $ red="\e[00;31m"
/home/cis90/simben/bin $ white="\e[01;37m"
/home/cis90/simben/bin $ blue="\e[00;34m"
/home/cis90/simben/bin $ echo -e $red RED $white WHITE $blue BLUE $off
  RED  WHITE  BLUE
/home/cis90/simben/bin $ echo -e ${red}RED ${white}WHITE ${blue}BLUE $off
  RED  WHITE  BLUE
/home/cis90/simben/bin $
```

```
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 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 home directory name → username

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

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

Going from CIS 90 username → home directory name

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

*This variable holds your
username*

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

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

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

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

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

And this is how you could use it

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

Assignment



Next Class

**Project is due
next week!**

Finish your project!

Cabrillo College



CIS 90 Final Project
Developing a bash script
Fall 2015

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
at	env	ls	spell
banner	exit	mail	su
bash	export	man	tail
bc	file	msg	tee
cal	find	mkdir	touch
cancel	finger	more	type
cat	grep	mv	unmask
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	xxd

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

A full-page background image showing a sunset over a beach. The sky is filled with vibrant orange, pink, and purple clouds. The sun is low on the horizon, casting a warm glow. To the right, a dark, silhouetted cliff rises from the beach. The foreground shows the wet sand of the beach reflecting the colors of the sky, with some dark rocks scattered about.

Wrap up

Commands:

basename

scp

tar

if then else

[]

- extract filename from pathname

- secure copy command

- archive command

- 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):

- Requirements for each task:
 - Minimum of 10 “original” script command lines
 - Has one or more non-generic comments to explain what it is doing
 - Has user interaction

You don't have to do all of these but do at least five:

- Redirecting stdin (5 points)
- Redirecting stdout (5 points)
- Redirecting stderr (5 points)
- Use of permissions (5 points)
- Use of filename expansion characters (5 points)
- Use of absolute path (5 points)
- Use of relative path (5 points)
- Use of a PID (5 points)
- Use of inodes (5 points)
- Use of links (5 points)
- Use of scheduling (5 points)
- Use of a GID or group (5 points)
- Use of a UID or user (5 points)
- Use of a /dev/tty device (5 points)
- Use of a signal (5 points)
- Use of piping (5 points)
- Use of an environment variable (5 points)
- Use of /bin/mail (5 points)
- Use of a conditional (5 points)

The maximum for this section is 25 points.



Backup