Rich's CCC Confer checklist - setup

Last updated 11/28/2018

- ❑ Zoom recording named and published for previous lesson

- ❑ Slides, Project, Lab X1 and Lab X2 posted
- ❑ Print out agenda slide and annotate page numbers

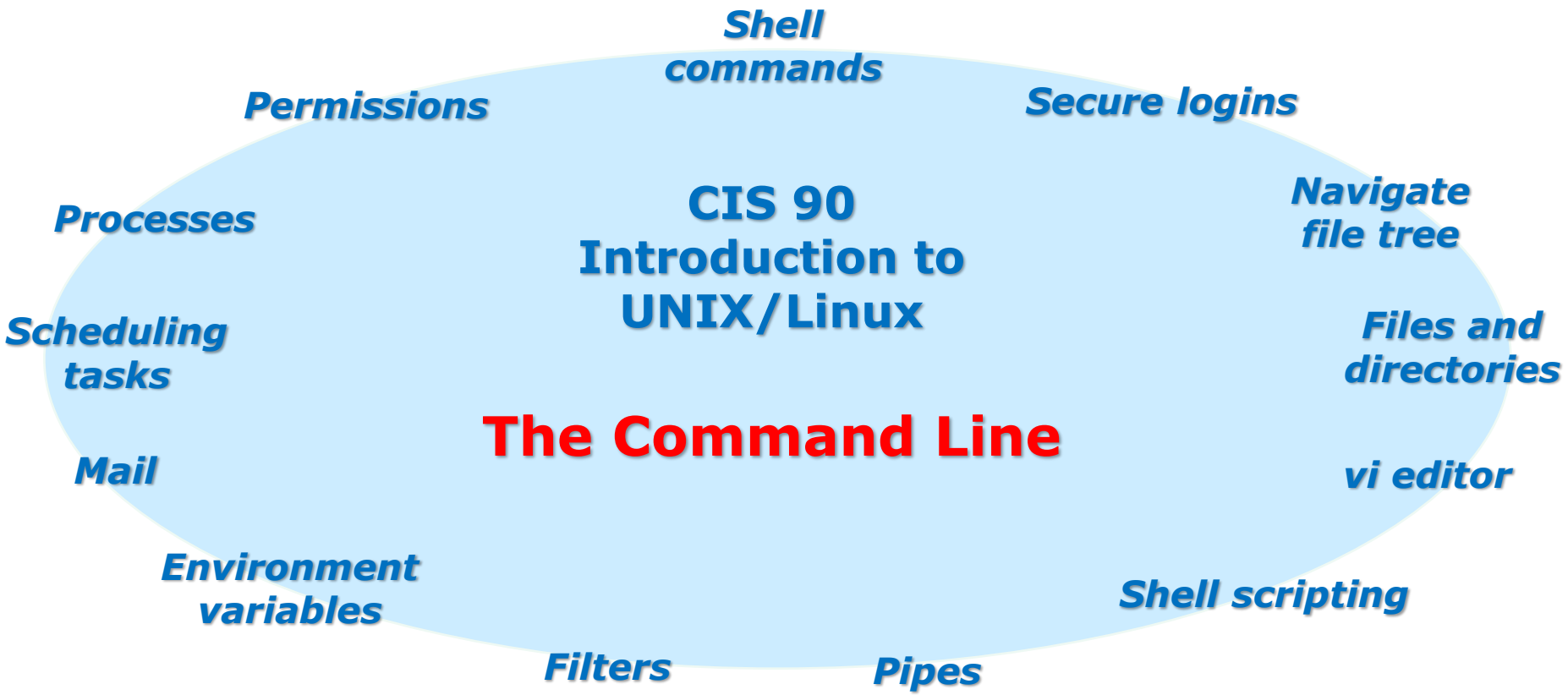
- ❑ Flash cards
- ❑ 1st minute quiz
- ❑ Web Calendar updated

- ❑ Dog and starter script examples ready

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

- ❑ https://zoom.us

- ❑ Putty + Slides + Chrome
- ❑ Enable/Disable attendee sharing
  ^ > Advanced Sharing Options > Only Host
- ❑ Enable/Disable attended annotations
  Share > More > Disable Attendee Sharing



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.

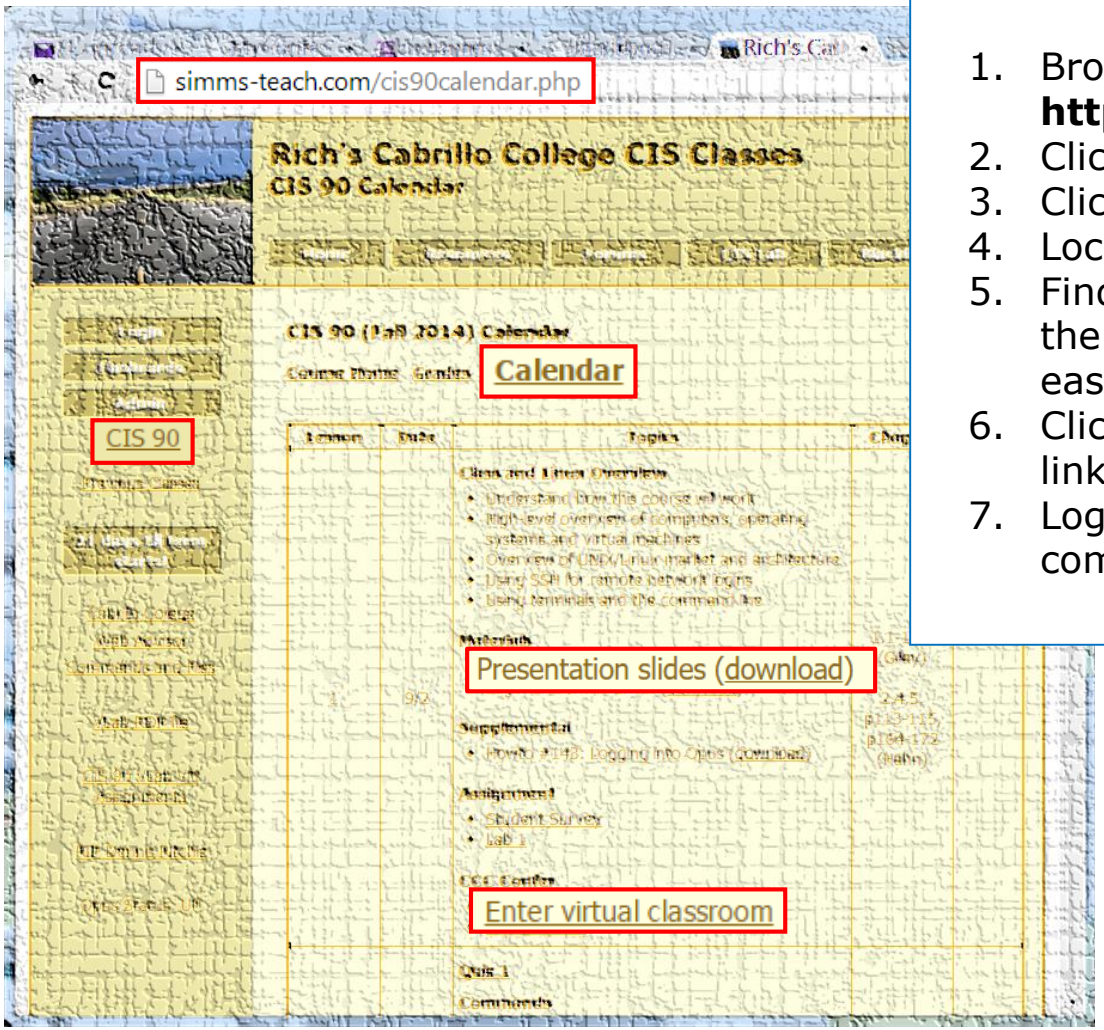
Introductions and Credits

Jim Griffin
- Created this Linux course
- Created Opus and the CIS VLab
- Jim's site: https://web.archive.org/web/20140209023942/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. John's site: http://teacherjohn.com/
- Jaclyn Kostner for many webinar best practices: e.g. mug shot page.

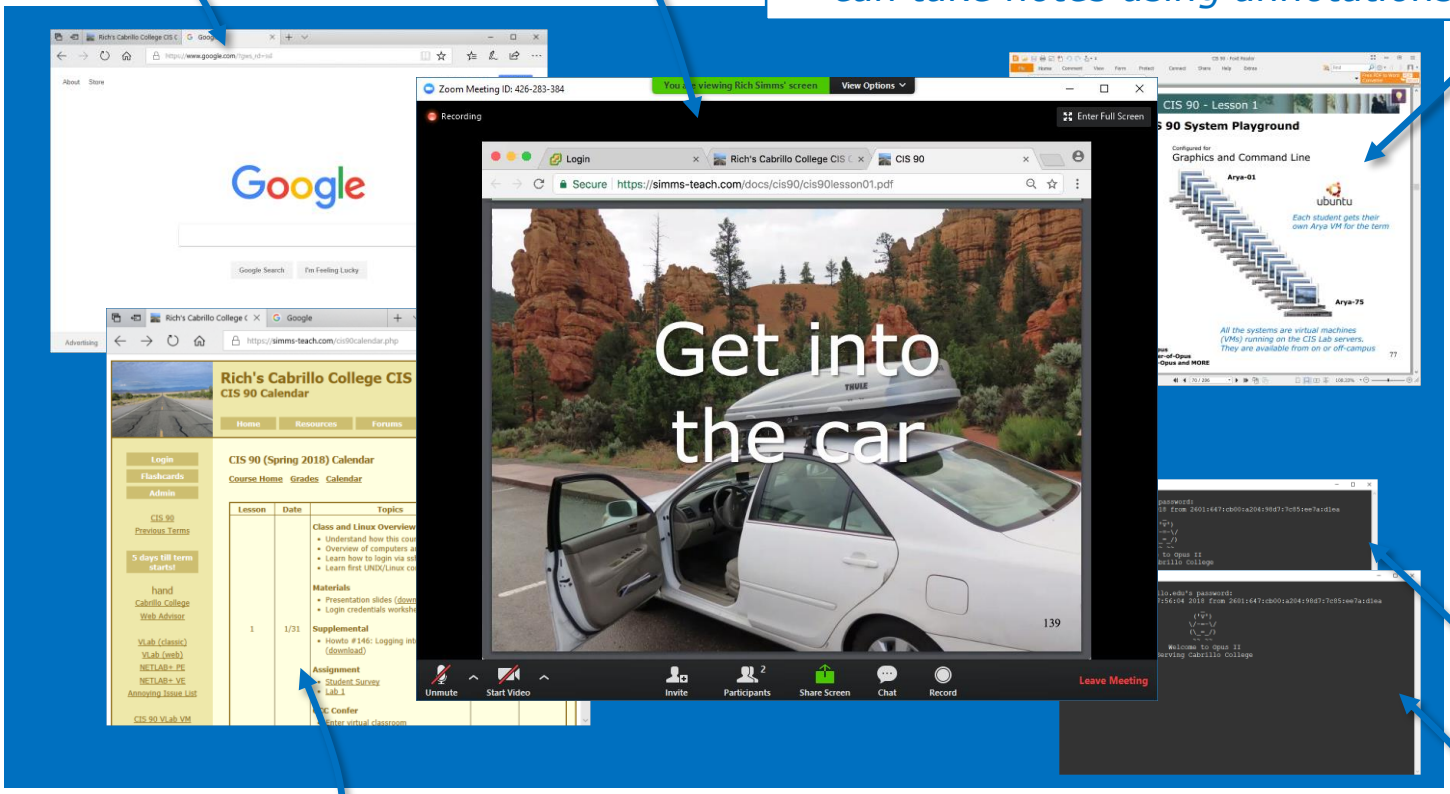
Student checklist - Before class starts



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 ConferZoom.
7. Log into Opus-II with Putty or ssh command.

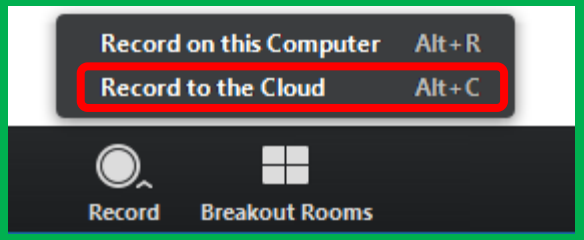
Student checklist - Before class starts

- ❑ *Google*
- ❑ *ConferZoom*
- ❑ *Downloaded PDF of Lesson Slides. I like Foxit Reader so I can take notes using annotations.*



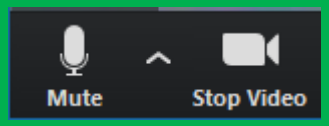
- ❑ *CIS 90 website Calendar page*
- ❑ *One or more login sessions to Opus-II*

Start



Start Recording

Audio Check



Start Recording

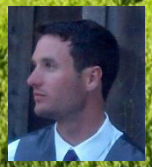
Audio & video Check

CIS 90 - Lesson 14

Instructor: **Rich Simms**
Dial-in: **408-638-0968 (toll)**
Meeting ID: **426 283 384**


Mikey


Jona


Joseph


Tara Marie


Fredi


Carina

Isaac


Matthew

Erik


Tony

Branden

Dominic

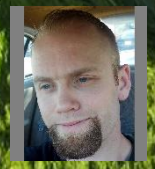
Ryan L.


Alejandra

Blair


Zari

Victor


Danny

Gabriel

Janelly


Austin


Aaron

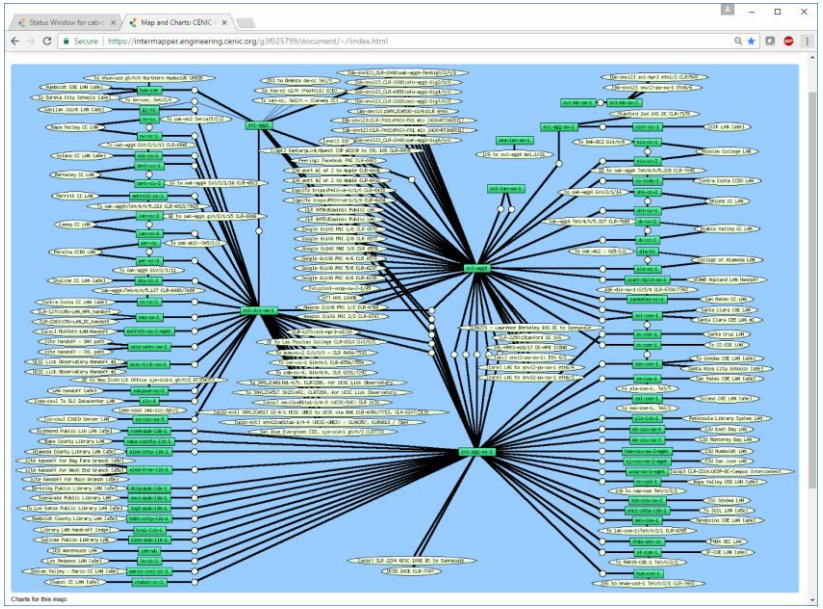

Ryan M.

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

Quiz

No Quiz Today !

Network Check

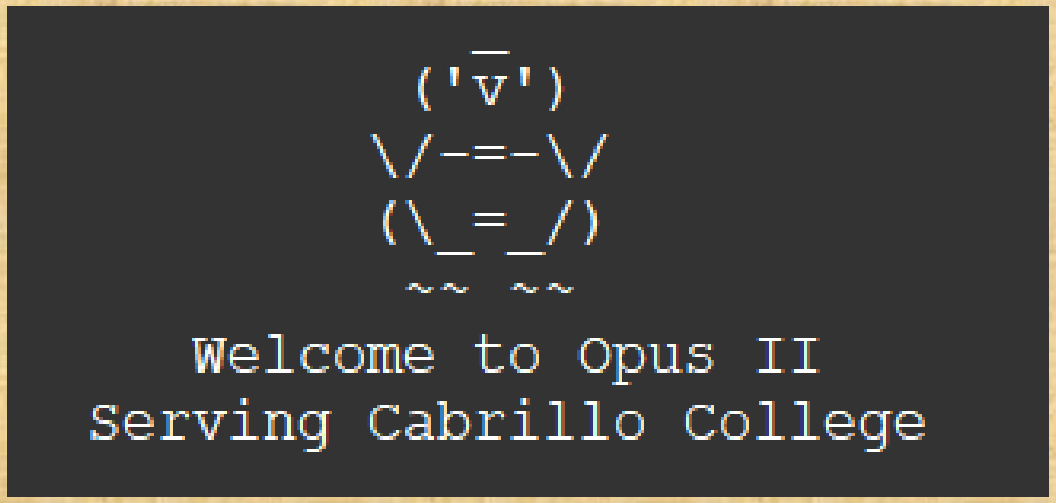


https://intermapper.engineering.cenic.org/g3f025799/document/~/!index.html

More Shell Scripting

Objectives	Agenda
• Transfer files between computers • Archive files using tar • Learn some scripting techniques	• 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

Class Activity



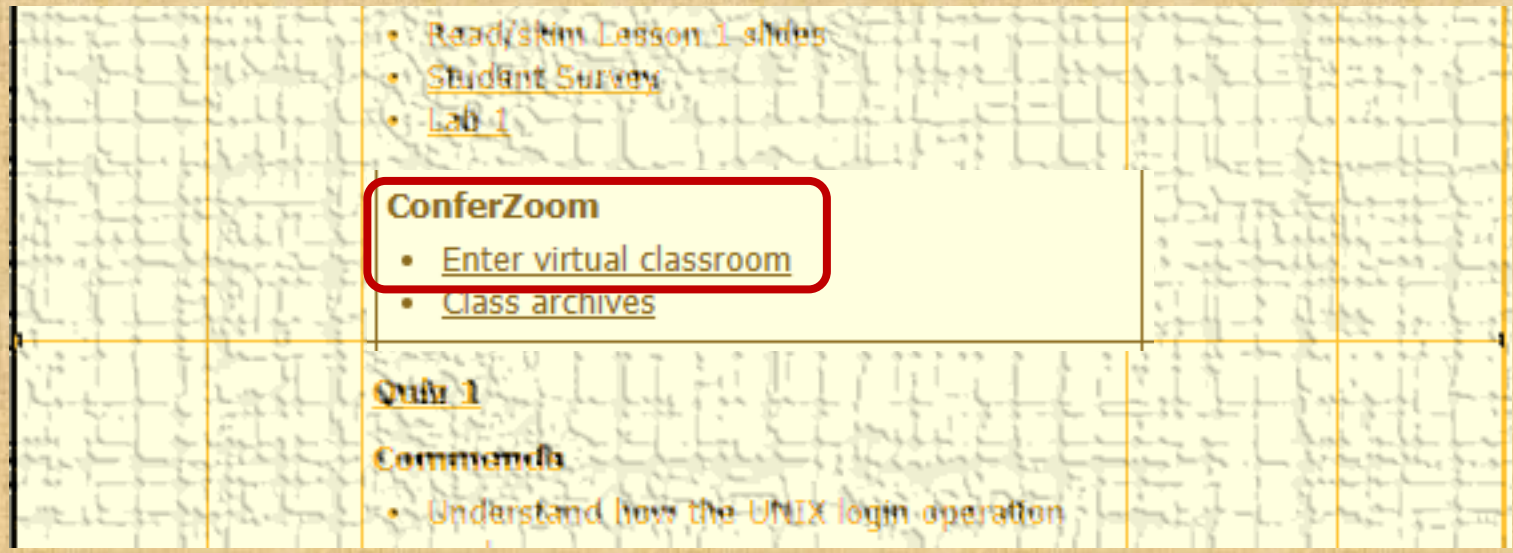
If you haven't already, log into Opus-II

Class Activity

https://simms-teach.com/cis90calendar.php

If you haven't already, download the lesson slides

Class Activity



https://simms-teach.com/cis90calendar.php

If you haven't already, join ConferZoom classroom

Questions

Questions?

Lesson material?

Labs? Tests?

How this course works?

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

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

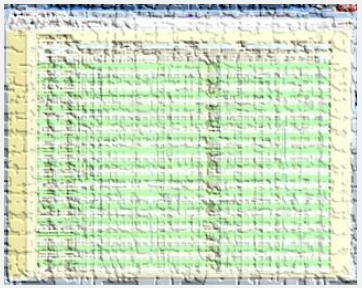
If you don't ask, you don't get.
- Mahatma Gandhi

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.*

Review your progress in the course

Check the website Grades page

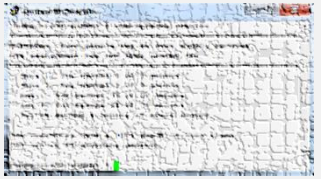
http://simms-teach.com/cis90grades.php



Or check on Opus-II

checkgrades *codename*

(where codename is your LOR codename)



Written by Jesse Warren a past CIS 90 Alumnus

- **Send me your survey to get your LOR codename.**
- **Graded labs and tests are in your home directories.**

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**

Extra Credit

On the forum

Be sure to monitor the forum as I may post extra credit opportunities without any other notice!

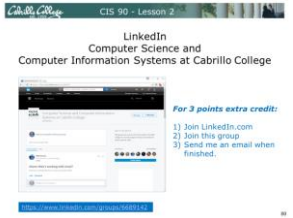
On some labs

Extra credit (2 points)

For a small taste of what you would learn in CIS 191 let's add a new user to your Arya VM. Once added we will see how the new account is represented in */etc/passwd* and */etc/shadow*.

1. Log into your Arya VM as the cis90 user. Make sure it's your VM and not someone else's.
2. Install the latest updates:
   ```
   sudo apt-get update
   sudo apt-get upgrade
   ```
3. Add a new user account for yourself. You may make whatever username you wish. The example below shows how Benji would make the same username he uses on Opus:
   ```
   sudo useradd -G sudo -c "Benji Simms" -m -s /bin/bash simben90
   ```

In lesson slides (search for extra credit)

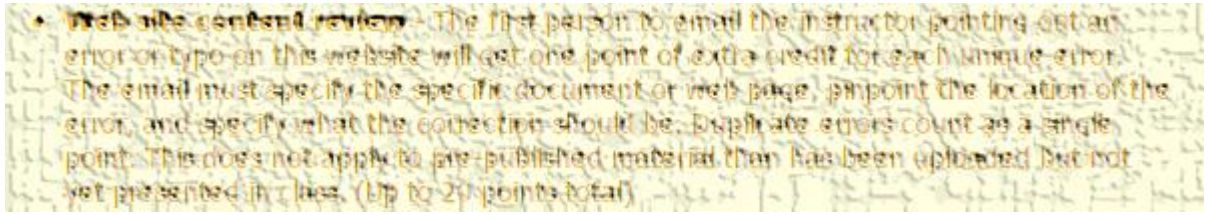


On the website

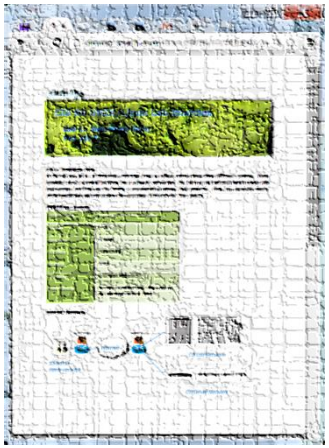
http://simms-teach.com/cis90grades.php

For some flexibility, personal preferences or family emergencies there is an additional 90 points available of **extra credit** activities.

http://simms-teach.com/cis90extracredit.php



Lab Assignments -- Pearls of Wisdom



- Don't wait till the last minute to start.
- Plan for things to go wrong and give yourself time to ask questions and get answers.
- The *slower* you go the *sooner* you will be finished.
- A few minutes reading the forum can save you hour(s).
- Line up materials, references, equipment and software ahead of time.
- It's best if you fully understand each step as you do it. Use Google or refer back to lesson slides to understand the commands you are using.
- Keep a growing cheat sheet of commands and examples.
- Study groups are very productive and beneficial.
- Use the forum to collaborate, ask questions, get clarifications and share tips you learned while doing a lab.
- **Late work is not accepted** so submit what you have for partial credit.

Getting Help When Stuck on an Assignment

- Google the topic/error message.
- Search the Lesson Slides (they are PDFs) for a relevant example on how to do something.
- Check the forum. Someone else may have run into the same issue and found a way past it. If not start a new topic, explain what you are trying to do and what you have tried so far.
- Talk to a STEM center tutor/assistant.
- Come see me during my office or lab hours:
  https://www.cabrillo.edu/salsa/listing.php?staffId=1426
  I'm in the CTC (room 1403) every Tuesday from 3:30-5:00 pm.
- Make use of the Open Questions time at the start of every class.
- Make a cheat sheet of commands and examples so you never again get stuck on the same thing!

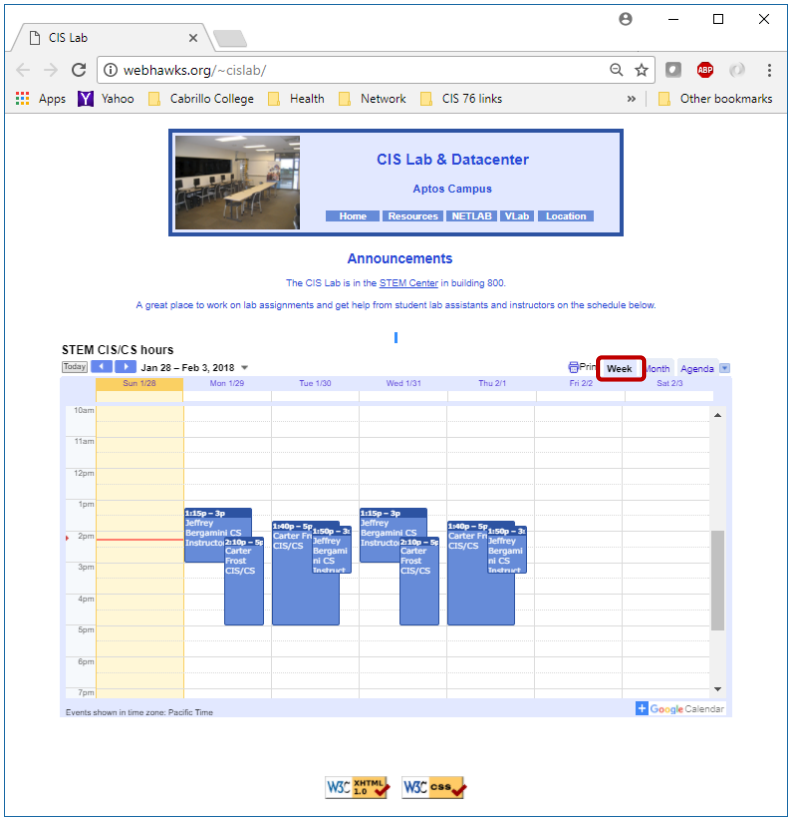
CIS Labs always involve some troubleshooting!

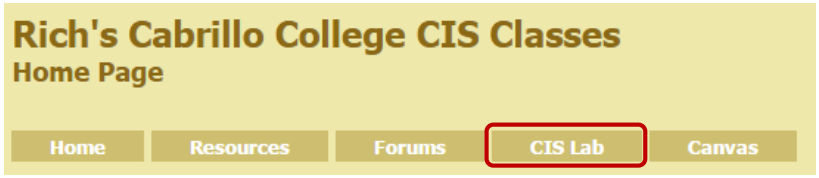
CTC - Building 1400
On lower campus

I will be in the CTC (room 1403) every Tuesday afternoon from 3:30-5:00

Help Available in the CIS Lab

Instructors, lab assistants and equipment are available for CIS students to work on assignments.





To see schedule, click the CIS Lab link on the website and use the "Week" calendar view

The slippery slope

1) If you didn't submit the last lab ...

2) If you were in class and didn't submit the last quiz ...

3) If you didn't send me the student survey assigned in Lesson 1 ...

4) If you haven't made a forum post in the last quarter of the course ...

5) If you had trouble doing the last test ...

Please contact me by email, see me during my office hours or when I'm in the CTC

Email: risimms@cabrillo.edu

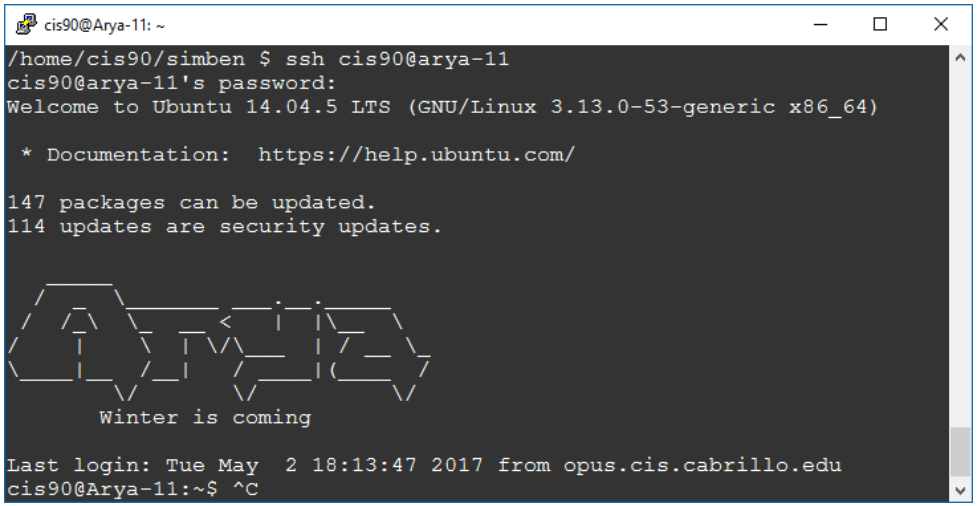
More on ssh

Running a command on a remote system

Did you know?

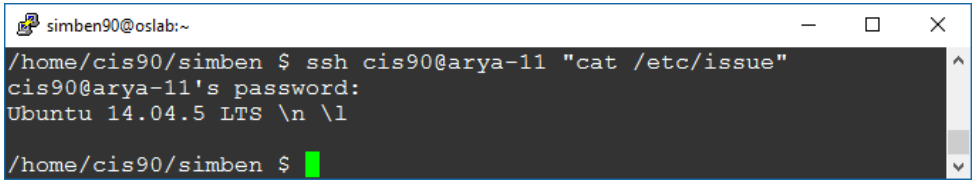
You can add a command to the end of an ssh command

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

Did you make it? Let me know in the chat window.

From your Arya run a remote command on Opus-II

Example 1

Use your own username

This command will be run on Opus-II

```
cis90@Arya-11:~$ ssh xxxxxx90@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=$(ssh xxxxxx90@opus-ii "who -s | cut -f1 -d' '")
simben90@opus's password:
cis90@Arya-11:~$ echo $opusUsers
rsimms rsimms jordan90 rsimms farsha154 pajste90 soramr90 soramr90
cis90@Arya-11:~$
```

Did it work? Let me know in the chat window.

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
```

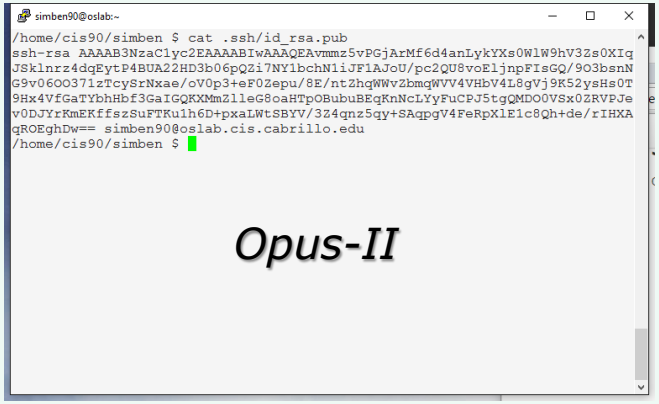
Your public key (can share with anyone)

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 your 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 .

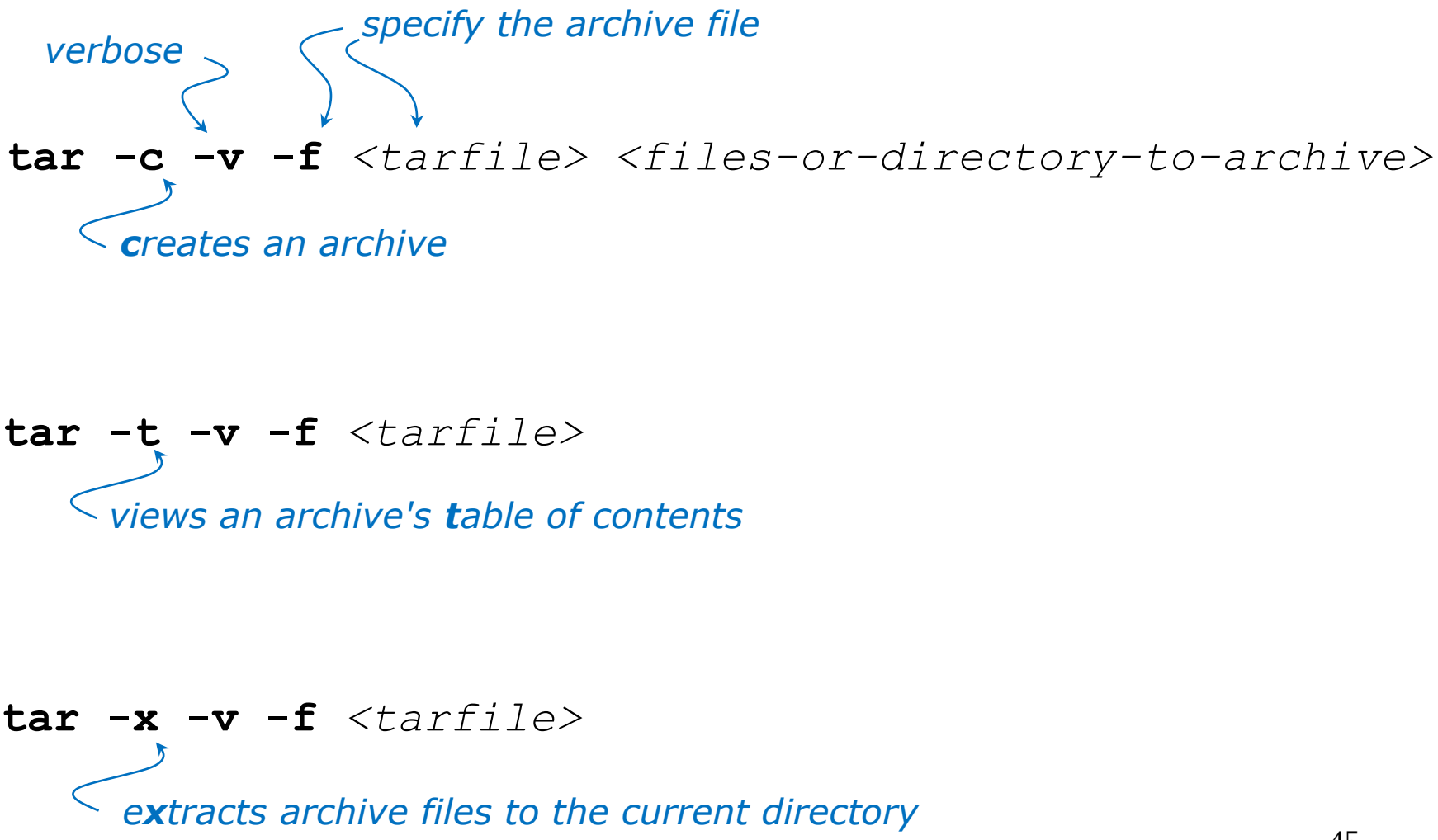
Use your own Opus-II username and password when trying this

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



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 $
```

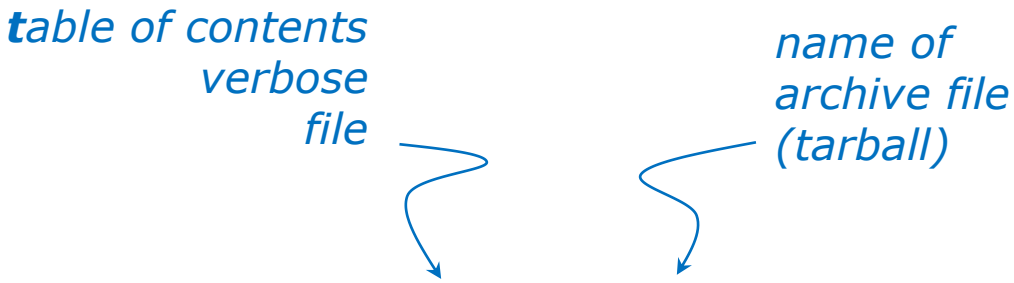
pathname to directory to archive

name of archive file (tarball)

***c**reate verbose file*

Example

Backup and restore a directory




```
/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

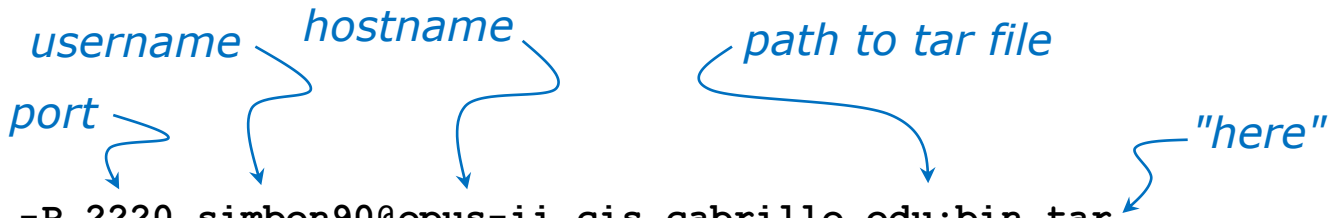
username *hostname*

```
/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

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 tbd1090 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 10th 1-3:50PM**

Mon	12/10	**Test #3 (the final exam)** **Time** • MONDAY 1:00PM - 3:50PM in Room 828 or online **Materials** • Presentation slides (download) • Test (canvas) **ConferZoom** • Enter virtual classroom • Class archives		5 posts Lab X1 Lab X2

Extra credit Labs X1/X2 and final posts ***due by 11:59PM***

Final grades *available by the end of the next day*

- 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 ConferZoom 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 2018 FINAL EXAMINATIONS SCHEDULE
DECEMBER 10 TO DECEMBER 15

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 10
9:00 am and 10:15 am, MW/Daily	7:00 am-9:50 am	Wednesday, December 12
10:20 am and 11:35 am, MW/Daily	10:00 am-12:50 pm	Monday, December 10
11:40 am and 12:55 pm, MW/Daily	10:00 am-12:50 pm	Wednesday, December 12
1:00 pm and 2:15 pm, MW/Daily	1:00 pm-3:50 pm	Monday, December 10
2:20 pm and 3:35 pm, MW/Daily	1:00 pm-3:50 pm	Wednesday, December 12
3:40 pm and 5:30 pm, MW/Daily	4:00 pm-6:50 pm	Wednesday, December 12

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

Section 1 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. This course has zero cost for textbooks.

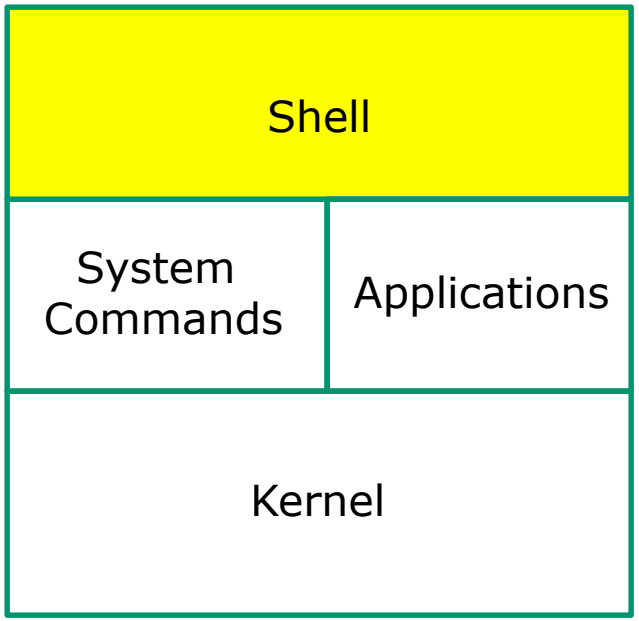
2	W	1:00PM-4:05PM	3.00	R.Simms	828
&	Arr.	Arr.		R.Simms	OL

Section 2 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. This course has zero cost for textbooks.

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:

/sbin

How many are scripts?

Write your answer in the chat window

Class Activity
Scripting

Of all the files in:

/etc

How many are shell scripts?

Hint: Use find command with -exec option

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 12 "original" lines of bash script – Has one or more non-generic comments to explain what it is doing – Has user interaction
24	At least six bash constructs from this list: • Redirecting stdin (4 points) • Redirecting stdout (4 points) • Redirecting stderr (4 points) • Use of permissions (4 points) • Use of filename expansion characters (4 points) • Use of absolute path (4 points) • Use of relative path (4 points) • Use of a PID (4 points) • Use of inodes (4 points) • Use of links (4 points) • Use of color (4 points) • Use of scheduling (4 points) • Use of a GID or group (4 points) • Use of a UID or user (4 points) • Use of a /dev/tty device (4 points) • Use of a signal (4 points) • Use of piping (4 points) • Use of an environment variable (4 points) • Use of /bin/mail (4 points) • Use of a conditional (4 points) • Use of $(*command*) The maximum for this section is 24 points.
6	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-II.
Extra credit	
30	Up to three additional tasks (10 points each)

Grading Rubric for Final Project

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 2018 CIS 90 Online Projects              *
**********************************************************
1) Aaron
2) Alejandra
3) Austin
4) Benji
5) Blair
6) Branden
7) Carina
8) Danny
9) Dominic
10) Duke
11) Erik
12) Fredi
13) Gabriel
14) Homer
15) Isaac
16) Janelly
17) Jona
18) Joseph
19) Matthew
20) Mikey
21) Ryan L.
22) Ryan M.
23) Tara
24) Tony
25) Victor
26) Zari

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:
```

Project Status

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

rsimms@opus-ii:~

```
[rsimms@opus-ii ~]$ ls -lt /home/cis90/*/bin/myscript
-rwxr-x---. 1 lewaus90 cis90   549 Nov 25 20:50 /home/cis90/lewaus/bin/myscript
-rwxr-x---. 1 climat90 cis90  8224 Nov 25 14:33 /home/cis90/climat/bin/myscript
-rwxr-x---. 1 farton90 cis90   774 Nov 25 08:42 /home/cis90/farton/bin/myscript
-rwxrwx--x. 1 mocrya90 cis90  2916 Nov 24 15:52 /home/cis90/mocrya/bin/myscript
-rwxrwxr-x. 1 navvic90 cis90   703 Nov 21 16:00 /home/cis90/navvic/bin/myscript
-rwxr-x---. 1 watshe90 cis90   706 Nov 21 14:55 /home/cis90/watshe/bin/myscript
-rwxr-x---. 1 caljos90 cis90   549 Nov 21 14:32 /home/cis90/caljos/bin/myscript
-rwxr-x---. 1 simben90 cis90 10513 Nov 21 12:26 /home/cis90/simben/bin/myscript
-rwxr-x---. 1 pindan90 cis90   714 Nov 16 21:44 /home/cis90/pindan/bin/myscript
-rwxrwxr-x. 1 berale90 cis90   705 Nov 14 14:09 /home/cis90/berale/bin/myscript
-rwxr-x---. 1 learya90 cis90   700 Nov 14 14:08 /home/cis90/learya/bin/myscript
-rwxrwxr-x. 1 alvjon90 cis90  1035 Nov 14 14:06 /home/cis90/alvjon/bin/myscript
-rwxr-x---. 1 galaar90 cis90   701 Nov 14 14:05 /home/cis90/galaar/bin/myscript
-rwxrwxr-x. 1 espdom90 cis90   713 Nov 14 14:03 /home/cis90/espdom/bin/myscript
-rwxr-x---. 1 gongab90 cis90   705 Nov 14 14:02 /home/cis90/gongab/bin/myscript
-rwxr-x---. 1 serjan90 cis90   703 Nov 14 14:01 /home/cis90/serjan/bin/myscript
-rwxr-x---. 1 rojfre90 cis90   697 Nov 14 13:58 /home/cis90/rojfre/bin/myscript
-rwxr-x---. 1 milhom90 cis90  4543 Nov 14 10:16 /home/cis90/milhom/bin/myscript
-rwxr-x---. 1 rodduk90 cis90  1686 Nov 11 12:59 /home/cis90/rodduk/bin/myscript
[rsimms@opus-ii ~]$
```

Is your script "hackable" by other classmates?

Don't forget to do this!

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

chmod 750 ~
chmod 750 ~/bin
chmod 750 ~/bin/myscript

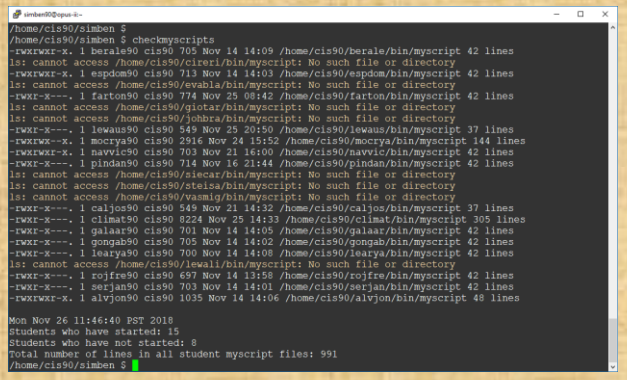
Points lost	
-15	Fails to run from **allscripts**
-15	Other students in the class are unable to read and execute your script.
-15	Error messages are displayed when running one or more tasks
-up to 90	No credit for any task which contains unoriginal script code that: • Doesn't give full credit to the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

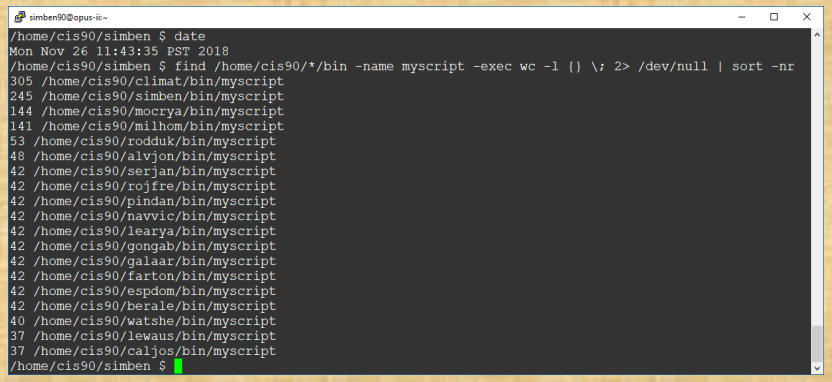
If you are not sure, log into Opus-II as the cis90 user and confirm

Final Project Progress

Run the following:

```
checkmyscripts

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





Don't wait till the last minute!

Shell Scripting 101

Example Scripts

starter-00:# Description: Hello World

starter-01:# Description: Just a bunch of commands

starter-02:# Description: Sh-bang, comments and exit

starter-03:# Description: Using variables and $(command) construct

starter-04:# Description: Clearing and pausing

starter-05:# Description: Reading user input

starter-06:# Description: Arguments and exit codes

starter-07:# Description: Using color

starter-08:# Description: Simple loop through list

starter-09:# Description: Simple loop through records in a file

starter-10:# Description: Simple loop through range of integers

starter-11:# Description: Simple loop for counting and parsing words in random poem lines

starter-12:# Description: Demonstrate simple if statement

starter-13:# Description: Scrape a web page for data

starter-14:# Description: Remotely control via ssh a Hue smart light

starter-15:# Description: Remotely turn off via ssh a Hue smart light

starter-16:# Description: Random numbers, terminal text placement and color

starter-17:# Description: Display a message on the STEM center LEDs

Shell Scripting 102

Example Scripts

starter-20:# Description: scraping log files (multiple versions v1-v5)

starter-21:# Description: tcp port probe

starter-30:# Description: Google Maps API

starter-31:# Description: Google Translation API

Shell Scripts

- In its simplest form a shell script can just be a list of commands in a file.
- Read "r" and execute "x" permissions must be enabled on the script file.
- The script must be on your path or you must use an absolute or relative pathname to run it.
- Putting #!/bin/bash on line 1 specifies which program should be used to execute the script. The default, if not specified, is /bin/bash. Note this enables vi to use color syntax.
- Putting the exit command at the end triggers a system call to the kernel to terminate the process and release all resources. Note a numerical status can be specified as an argument (e.g. exit 20) which will be communicated back to the parent process.

lastb and btmp log

```
rsimms@sun-hwa:~$ sudo lastb -awi | tail -n5
root     ssh:notty    Fri May 11 02:13 - 02:13  (00:00)     222.101.130.159
root     ssh:notty    Fri May 11 02:13 - 02:13  (00:00)     222.101.130.159
root     ssh:notty    Fri May 11 02:13 - 02:13  (00:00)     222.101.130.159
btmp begins Fri May 11 02:13:26 2018


rsimms@sun-hwa:~$ ls /var/log/btmp*
/var/log/btmp    /var/log/btmp.2  /var/log/btmp.4  /var/log/btmp.6  /var/log/btmp.8
/var/log/btmp.1  /var/log/btmp.3  /var/log/btmp.5  /var/log/btmp.7


rsimms@sun-hwa:~$ sudo lastb -awi -f /var/log/btmp.7 | tail -n5
test     ssh:notty    Sun Oct  1 14:29 - 14:29  (00:00)     111.53.3.160
test     ssh:notty    Sun Oct  1 14:29 - 14:29  (00:00)     111.53.3.160
test     ssh:notty    Sun Oct  1 14:29 - 14:29  (00:00)     111.53.3.160
btmp.7 begins Sun Oct  1 14:29:38 2017
```

Failed logins, such as is seen with brute force attacks, are logged in /var/log/btmp

Scraping Log Files v1

```
/home/cis90/simben/bin $ cd ~/bin
/home/cis90/simben/bin $ cp ~/../depot/scripts/starter-20.v1 .
/home/cis90/simben/bin $ vi starter-20.v1
```

simben90@opus-ii:~/bin

```
#!/bin/bash
# Scripting 101 sample script for CIS 90
# Description: log scraping

logfile=/home/cis90/depot/logs/sun-hwa-btmp-log

echo "FAILED LOGIN REPORT"
echo
echo "Log file: $logfile"
echo "Date: $(date +%m/%d/%Y)"
echo
echo "Long listing:"
ls -l $logfile
echo
echo "Number of lines:"
wc -l $logfile

exit
~
"./starter-20.v1" 18L, 315C                                    1,1           All
```

Use Esc **:wq** *to save file and quit vi*

This practice log file was created by concatenating ***lastb -awi*** *output for all the wtmp files on Sun-Hwa.*

Headers and and local failed login attempts were removed.

```
/home/cis90/simben/bin $ chmod +x starter-20.v1
/home/cis90/simben/bin $ starter-20.v1
```

How many lines are in the sun-hwa-btmp-log file?

Write your answer in the chat window

Scraping Log Files v2

```
/home/cis90/simben/bin $ cd ~/bin
/home/cis90/simben/bin $ cp ~/../depot/scripts/starter-20.v2 .
/home/cis90/simben/bin $ vi starter-20.v2
```

simben90@opus-ii:~/bin

```
#!/bin/bash
# Scripting 101 sample script for CIS 90
# Description: log scraping

logfile=/home/cis90/depot/logs/sun-hwa-btmp-log

echo "FAILED LOGIN REPORT"
echo
echo "Log file: $logfile"
echo "Date: $(date +%m/%d/%Y)"
echo

echo Top 15 usernames used:
cat $logfile | cut -f1 -d" " | sort | uniq -c | sort -rn | head -n15

exit
~
~
~
"./starter-20.v2" 16L, 332C                                   7,1           All
```

*The **-c** option on **uniq** does a count of matching records.*

*The **-rn** options on **sort** does a reverse numeric sort.*

Use Esc **:wq** *to save file and quit vi*

```
/home/cis90/simben/bin $ chmod +x starter-20.v2
/home/cis90/simben/bin $ starter-20.v2
```

What username did the attackers use the most?

Write your answer in the chat window

Scraping Log Files v3

```
/home/cis90/simben/bin $ cd ~/bin
/home/cis90/simben/bin $ cp ~/../depot/scripts/starter-20.v3 .
/home/cis90/simben/bin $ vi starter-20.v3
```

simben90@opus-ii:~/bin

```
#!/bin/bash
# Scripting 101 sample script for CIS 90
# Description: log scraping

logfile=/home/cis90/depot/logs/sun-hwa-btmp-log

echo "FAILED LOGIN REPORT"
echo
echo "Log file: $logfile"
echo "Date: $(date +%m/%d/%Y)"
echo

echo Top 15 usernames used:
cat $logfile | cut -f1 -d" " | sort | uniq -c | sort -rn | head -n15
echo
echo Attackers and attempts:
cat $logfile | tr -s ' ' | cut -f 15 -d" " | sort | uniq -c | sort -rn
exit
~
"./starter-20.v3" 18L, 436C                                   7,1           All
```

*The **tr -s ' '** command squeezes multiple blanks to a single blank.*

The 15th field (using blank as field separator) is the IP address.

Use Esc **:wq** *to save file and quit vi*

```
/home/cis90/simben/bin $ chmod +x starter-20.v3
/home/cis90/simben/bin $ starter-20.v3
```

Which IP address was responsible for the most attacks?

Write your answer in the chat window

Scraping Log Files v4

```
/home/cis90/simben/bin $ cd ~/bin
/home/cis90/simben/bin $ cp ~/../depot/scripts/starter-20.v4 .
/home/cis90/simben/bin $ vi starter-20.v4
```

simben90@opus-ii:~/bin

```
#!/bin/bash
# Scripting 101 sample script for CIS 90
# Description: log scraping

logfile=/home/cis90/depot/logs/sun-hwa-btmp-log

echo "FAILED LOGIN REPORT"
echo
echo "Log file: $logfile"
echo "Date: $(date +%m/%d/%Y)"
echo

echo Top 15 usernames used:
cat $logfile | cut -f1 -d" " | sort | uniq -c | sort -rn | head -n15
echo
echo Attackers and attempts:
cat $logfile | tr -s ' ' | cut -f 15 -d" " | sort | uniq -c | sort -rn | tee blacklist
echo
echo Reverse lookup of attacker IP addresses:
while read line; do
  ip=$(echo $line | cut -f2 -d" ")
  name=$(host $ip)
  if [ "$?" != "0" ]; then name="Name not found"; fi
  echo -e "$ip:  ($name)"
done < blacklist

exit
~
"./starter-20.v4" 27L, 675C                                   7,1           All
```

The attacker IP addresses are saved in a file named blacklist.

This new file is then read line by line to do a reverse DNS lookup to discover if the IP address has a name.

Use Esc **:wq** *to save file and quit vi*

```
/home/cis90/simben/bin $ chmod +x starter-20.v4
/home/cis90/simben/bin $ starter-20.v4
```

Which of the attacker IP addresses resolves to mx1.coteccons.vn?

Write your answer in the chat window

Scraping Log Files v5

```
/home/cis90/simben/bin $ cd ~/bin
/home/cis90/simben/bin $ cp ~/../depot/scripts/starter-20.v5 .
/home/cis90/simben/bin $ vi starter-20.v5
```

simben90@opus-ii:~/bin

```
#!/bin/bash
# Scripting 101 sample script for CIS 90
# Description: log scraping

logfile=/home/cis90/depot/logs/sun-hwa-btmp-log

echo "FAILED LOGIN REPORT"
echo
echo "Log file: $logfile"
echo "Date: $(date +%m/%d/%Y)"
echo

echo Top 15 usernames used:
cat $logfile | cut -f1 -d" " | sort | uniq -c | sort -rn | head -n15
echo
echo Attackers and attempts:
cat $logfile | tr -s ' ' | cut -f 15 -d" " | sort | uniq -c | sort -rn | tee blacklist
echo
echo Reverse lookup of attacker IP addresses:
while read line; do
  ip=$(echo $line | cut -f2 -d" ")
  name=$(host $ip)
  if [ "$?" != "0" ]; then name="Name not found"; fi
  echo -e "$ip:  ($name)"
  whois $ip | grep -i "^city:"
  whois $ip | grep -i "^country:"
  whois $ip | grep -i "^address:"
  whois $ip | grep -i "^address:"
  whois $ip | grep -i "^person:"
  echo
done < blacklist

exit
~
"./starter-20.v5" 33L, 848C                                   1,1           All
```

The attacker IP addresses are saved in a file named blacklist.

This new file is then read line by line to do a reverse DNS lookup and gather selected whois data fields.

Use Esc **:wq** *to save file and quit vi*

```
/home/cis90/simben/bin $ chmod +x starter-20.v5
/home/cis90/simben/bin $ starter-20.v5
```

From which country did most of the attacks come from?

Write your answer in the chat window

Scheduling a report to be emailed

```
/home/cis90/simben/bin $ at midnight
at> starter-20.v5 | mail -s "Failed Login Report" $LOGNAME
at> <EOT>
job 2413 at Mon Nov 26 00:00:00 2018
/home/cis90/simben/bin $

/home/cis90/simben/bin $ mail
Heirloom Mail version 12.5 7/5/10.  Type ? for help.
"/var/spool/mail/simben90": 1 message 1 unread
>U  1 Benji Simms           Mon Nov 26 00:00  67/1904  "Failed Login Report"
& 1
Message  1:
From simben90@opus-ii.cis.cabrillo.edu  Mon Nov 26 00:00:02 2018
<snipped>
Status: RO

FAILED LOGIN REPORT

Log file: /home/cis90/depot/sun-hwa-btmp-log
Date: 11/26/2018
<snipped>
```

Did it work?

Write your answer in the chat window

Apache and Postfix Logs

```
[root@opus-ii logs]# ls /var/log/httpd/
access_log           error_log-20181118           ssl_access_log-20181104  ssl_error_log-20181125
access_log-20181104  error_log-20181125           ssl_access_log-20181111  ssl_request_log
access_log-20181111  redirect.error.log          ssl_access_log-20181118  ssl_request_log-20181104
access_log-20181118  redirect.error.log-20181104 ssl_access_log-20181125  ssl_request_log-20181111
access_log-20181125  redirect.error.log-20181111 ssl_error_log            ssl_request_log-20181118
error_log            redirect.error.log-20181118 ssl_error_log-20181104   ssl_request_log-20181125
error_log-20181104   redirect.error.log-20181125 ssl_error_log-20181111
error_log-20181111   ssl_access_log              ssl_error_log-20181118
```

```
[root@opus-ii logs]# ls /var/log/maillog*
/var/log/maillog           /var/log/maillog-20181111  /var/log/maillog-20181125
/var/log/maillog-20181104  /var/log/maillog-20181118
```

See /home/cis90/depot/logs for more logs to play with or use in your final project.

TCP Port Probe

```
/home/cis90/simben/bin $ cd ~/bin
/home/cis90/simben/bin $ cp ~/../depot/scripts/starter-21 .
/home/cis90/simben/bin $ vi starter-21
```

rsimms@opus-ii:/home/cis90/depot/scripts

```
#!/bin/bash
# Scripting 101 sample script for CIS 90
# Description: TCP port probe

# See: http://xmodulo.com/tcp-udp-socket-bash-shell.html

host="arya-36"
port="80"

(echo >/dev/tcp/$host/$port)
if [ "$?" = "0" ]; then
  echo $host: port $port is up
else
  echo $host: port $port is down
fi

exit
~
"starter-21" 17L, 302C                                1,1           All
```

This does a three-way TCP handshake to open a connection to a port on a remote host.

This example tests port 90 (HTTP service).

Use Esc **:wq** *to save file and quit vi*

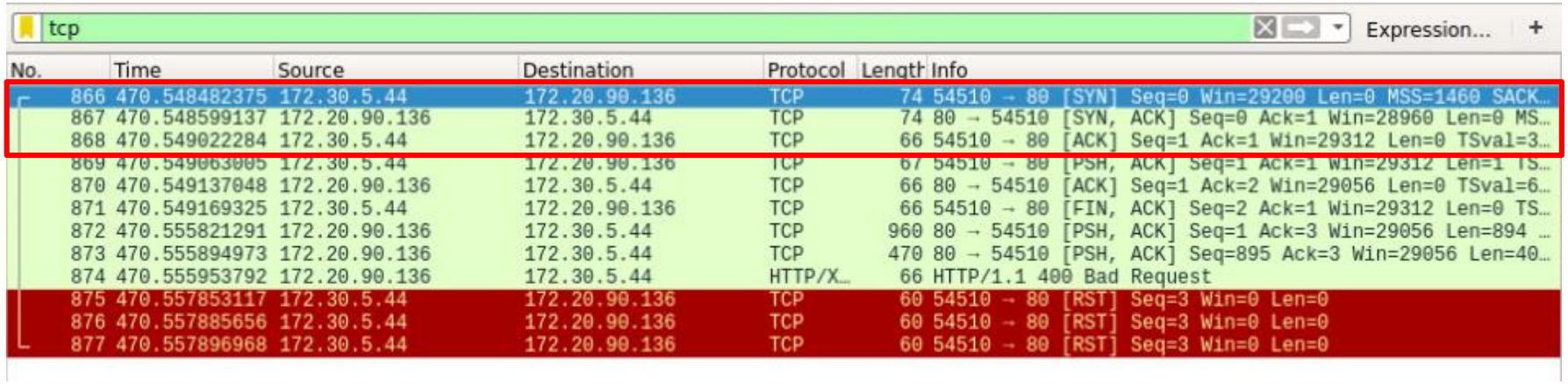
```
/home/cis90/simben/bin $ chmod +x starter-21
/home/cis90/simben/bin $ starter-21
```

Try it, is port 80 open on Arya-36?

Write your answer in the chat window

TCP Port Probe

```
/home/cis90/simben/bin $ ./starter-21
arya-36: port 80 is up
/home/cis90/simben/bin $
```



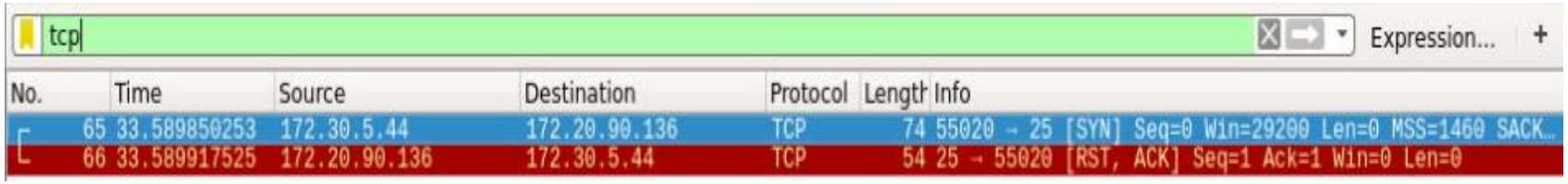
The TCP connection is established after the three way handshake: SYN, SYN-ACK, ACK

Note: To install Wireshark on your Arya VM use:
sudo apt-get update
sudo apt-get install wireshark

TCP Port Probe

Edit starter-21 and on line 8 change the port to 25 (SMTP): **port="25"**

```
/home/cis90/simben/bin $ ./starter-21
./starter-21: connect: Connection refused
./starter-21: line 10: /dev/tcp/arya-36/25: Connection refused
arya-36: port 25 is down
/home/cis90/simben/bin $
```



This Arya does not have a SMTP mail service so the connection is refused.

Note: To install Wireshark on your Arya VM use:
sudo apt-get update
sudo apt-get install wireshark

TCP Port Probe

```
/home/cis90/simben/bin $ cd ~/bin
/home/cis90/simben/bin $ cp ~/../depot/scripts/starter-21 .
/home/cis90/simben/bin $ vi starter-21
```

rsimms@opus-ii:/home/cis90/depot/scripts

```
#!/bin/bash
# Scripting 101 sample script for CIS 90
# Description: TCP port probe

# See: http://xmodulo.com/tcp-udp-socket-bash-shell.html

host="arya-36"
port="80"

(echo >/dev/tcp/$host/$port)
if [ "$?" = "0" ]; then
  echo $host: port $port is up
else
  echo $host: port $port is down
fi

exit
~
"starter-21" 17L, 302C                                1,1           All
```

This does a three-way TCP handshake to open a connection to a port on a remote host.

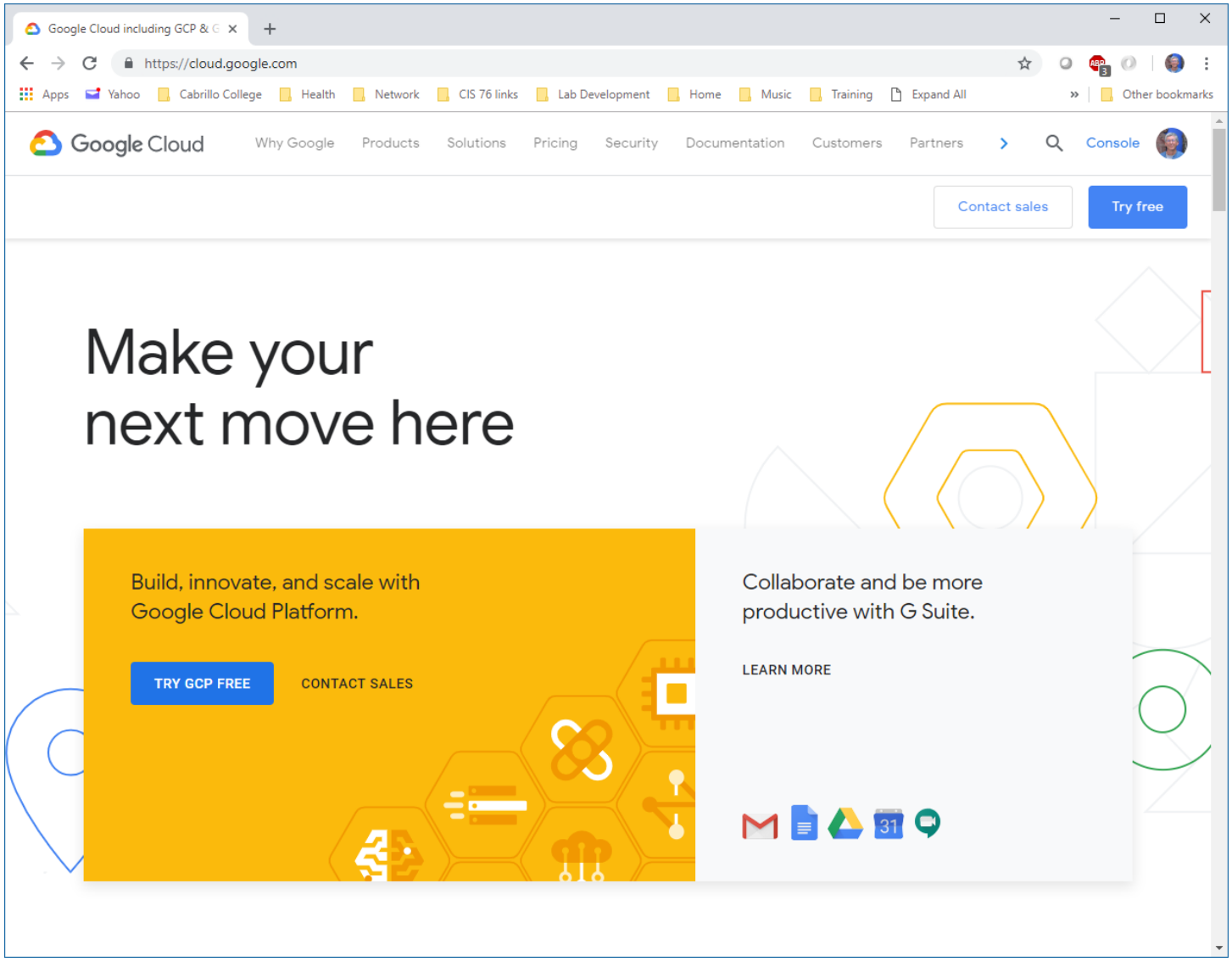
This example tests port 80 (HTTP service).

Use Esc **:wq** *to save file and quit vi*

```
/home/cis90/simben/bin $ chmod +x starter-21
/home/cis90/simben/bin $ starter-21
```

Change the port to 443 (SSL), is that port open on Arya-36?

Write your answer in the chat window


Google Cloud including GCP &
https://cloud.google.com
Apps
Yahoo
Cabrillo College
Health
Network
CIS 76 links
Lab Development
Home
Music
Training
Expand All
Other bookmarks
Google Cloud
Why Google
Products
Solutions
Pricing
Security
Documentation
Customers
Partners
Console
Contact sales
Try free
Make your next move here
Build, innovate, and scale with Google Cloud Platform.
TRY GCP FREE
CONTACT SALES
Collaborate and be more productive with G Suite.
LEARN MORE

Geolocation using Google Maps API

```
/home/cis90/simben/bin $ cd ~/bin
/home/cis90/simben/bin $ cp ~/../depot/scripts/starter-30 .
/home/cis90/simben/bin $ vi starter-30
```

rsimms@opus-ii:/home/cis90/depot/scripts

```
#!/bin/bash
# Scripting 101 sample script for CIS 90

# To get your key see https://www.youtube.com/watch?v=9ImLCQBj9SE
key="paste your Google API key here"

read -p "Enter address: " address
if [ "$address" = "" ]; then
  address="Aptos CA"
fi
location=$(echo $address | tr " " "+")

googleMapsAPI="https://maps.googleapis.com/maps/api/geocode/xml"
curl "$googleMapsAPI?address=$location&key=$key" -o results 2> /dev/null

echo Location: $address
grep -A3 "<location>" results | head -n3 | tail -n2

exit

~
"starter-30" 20L, 507C                                1,1           All
```

Use Esc **:wq** *to save file and quit vi*

tr " " "+" *replaces blanks with pluses in the text entered by the user.*

The **-o** *option of* **curl** *outputs the downloaded web page to a file.*

The **-A3** *option on* **grep** *includes 3 lines after the match.*

```
/home/cis90/simben/bin $ chmod +x starter-30
/home/cis90/simben/bin $ starter-30
```

This script requires you to get a Google Cloud account and API key

Let me know in the chat window if you already have an account

```
/home/cis90/simben/bin $ ./starter-30
Enter address: Aptos CA
Location: Aptos CA
    <lat>36.9771729</lat>
    <lng>-121.8994016</lng>

/home/cis90/simben/bin $ ./starter-30
Enter address: Greenwich England
Location: Greenwich England
    <lat>51.4825766</lat>
    <lng>-0.0076589</lng>

/home/cis90/simben/bin $ ./starter-30
Enter address: maui usa
Location: maui usa
    <lat>20.7983626</lat>
    <lng>-156.3319253</lng>

/home/cis90/simben/bin $ ./starter-30
Enter address: 1600 Pennsylvania Ave Washington DC
Location: 1600 Pennsylvania Ave Washington DC
    <lat>38.8976633</lat>
    <lng>-77.0365739</lng>
```

Language translation using Google Translation API

```
/home/cis90/simben/bin $ cd ~/bin
/home/cis90/simben/bin $ cp ~/../depot/scripts/starter-31 .
/home/cis90/simben/bin $ vi starter-31
```

rsimms@opus-ii:/home/cis90/depot/scripts

```
#!/bin/bash
# Scripting 101 sample script for CIS 90
# Description: language translation using Google Translation API

# To get key see: https://cloud.google.com/translate/docs/quickstart?authuser=1
export GOOGLE_APPLICATION_CREDENTIALS=$HOME/bin/google-API-key.json

read -p "Enter phrase to translate: " phrase
if [ "$phrase" = "" ]; then
  phrase="Excuse me but this is my first time."
fi

curl -s -X POST -H "Content-Type: application/json" \
    -H "Authorization: Bearer "$(gcloud auth application-default print-access-token) \
    --data "{
  'q': '$phrase',
  'source': 'en',
  'target': 'fr',
  'format': 'text'
}" "https://translation.googleapis.com/language/translate/v2"

exit

"starter-31" 23L, 690C                                    1,1           All
```

You must get a private key to use the translation service and install the latest Google cloud SDK.

Use Esc **:wq** *to save file and quit vi*

```
/home/cis90/simben/bin $ chmod +x starter-31
/home/cis90/simben/bin $ starter-31
```

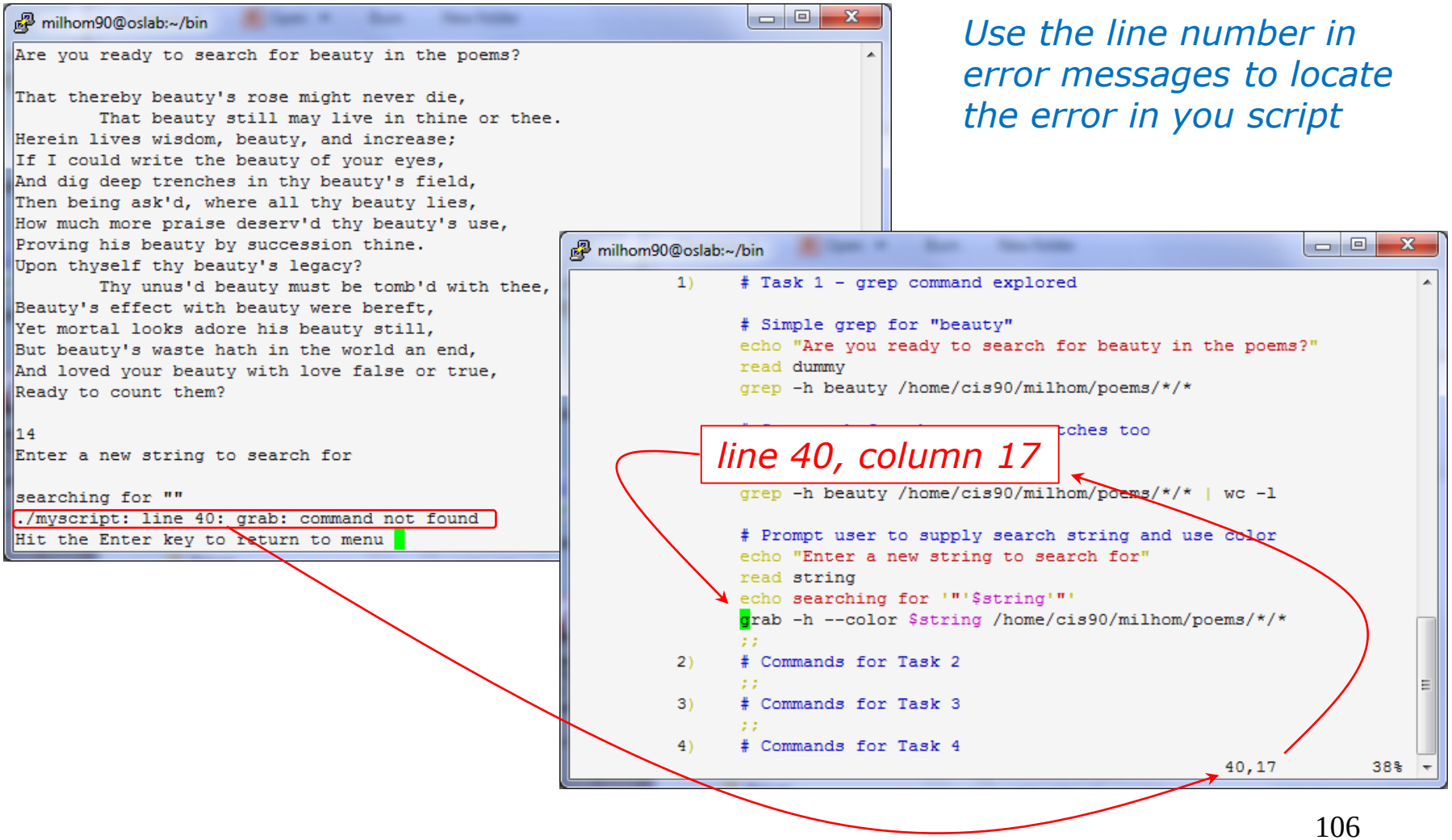
This script requires a Google Cloud service account and private API key

Let me know in the chat window if you already have an account

```
/home/cis90/simben/bin $ ./starter-31
Enter phrase to translate: Excuse me but this is my first time.
{
  "data": {
    "translations": [
      {
        "translatedText": "Excusez-moi mais c'est ma première fois."
      }
    ]
  }
}

/home/cis90/simben/bin $ ./starter-31
Enter phrase to translate: Where is a restroom?
{
  "data": {
    "translations": [
      {
        "translatedText": "Où sont les toilettes?"
      }
    ]
  }
}
```

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

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%
```

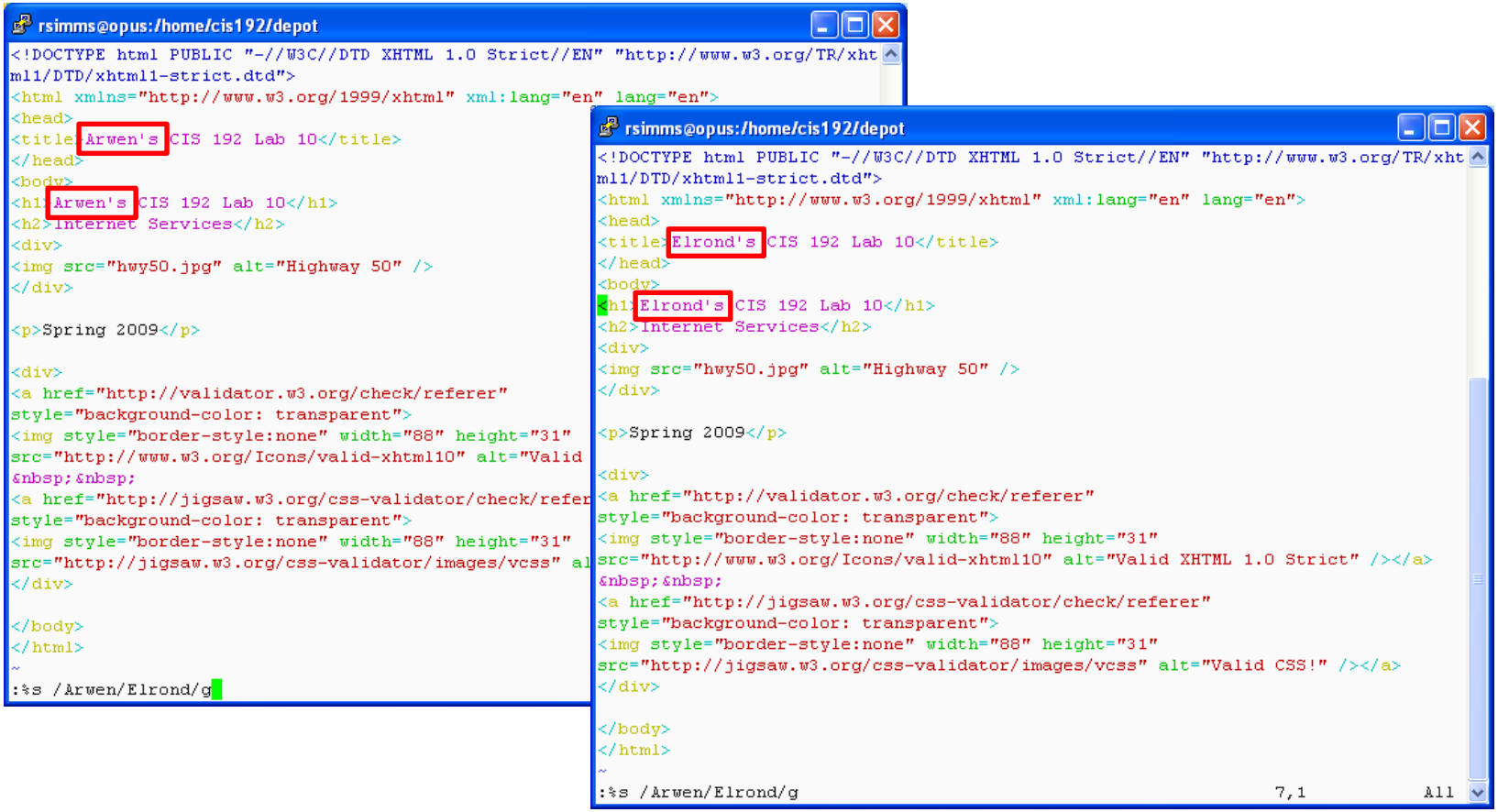
milhom90@oslab:~/bin

```
        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
        ;;
                                                37,55         59%
```

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

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*

Activity

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
```

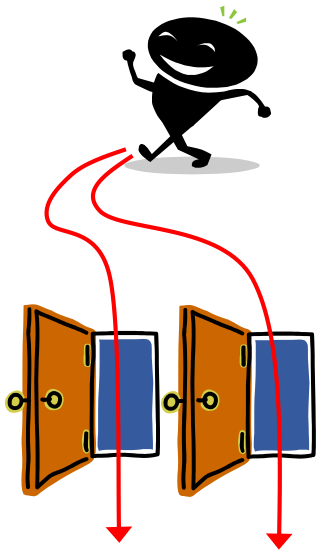
Let me know in the chat window when you have finished

Scripting Tips

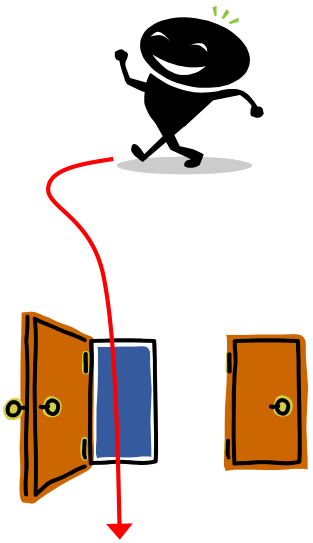
if statement with "or"

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

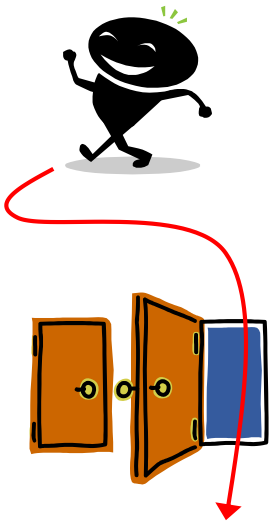
OR logic



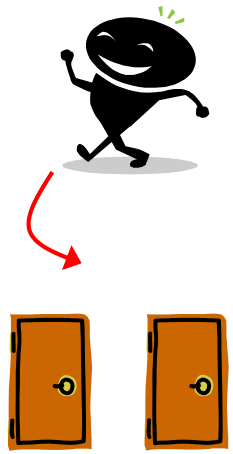
Yes



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"*

Activity

Copy the example script in the depot directory to your bin directory:

```
cd bin
cp ~/../depot/scripts/example654 .
```

View the script:

```
cat example654
```

```
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
```

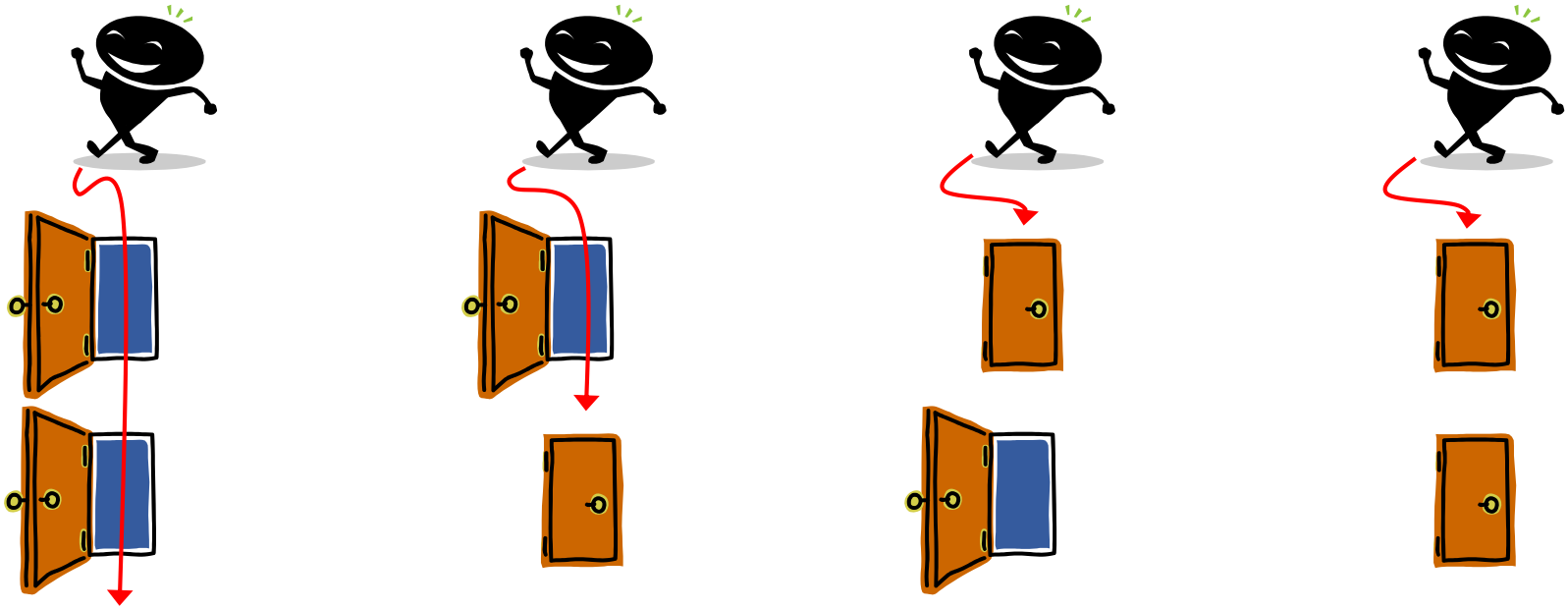
Let me know in the chat window when you have finished

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
```

Activity

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
```

Let me know in the chat window when you have finished

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

Activity

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
```

Let me know in the chat window when you have finished

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
```

Activity

Run the previous example task
- run **allscripts**
- select Homer's script
- select Task **3**

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

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

Let me know in the chat window when you have finished

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.

See: /home/cis90/depot/scripts/example101

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.

See: /home/cis90/depot/scripts/example102

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.

See: /home/cis90/depot/scripts/example103

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.

See: /home/cis90/depot/scripts/example104

Activity

Copy the example scripts in the depot directory to your bin directory:

```
cd bin
cp ~/../depot/scripts/example10* .
```

View the scripts:

```
head -n50 example10*
```

Prepare and run them

```
chmod -v +x example10*
example101
example102
example103
example104
```

Let me know in the chat window when you have finished

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

Activity

Copy the example script in the depot directory to your bin directory:

```
cd bin
cp ~/../depot/scripts/example777 .
```

View the script:

```
vi example777
```

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

Prepare and run your script

```
chmod +x example777
example777
```

Let me know in the chat window when you have finished

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

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

```
/home/cis90/simben $ echo The home of $LOGNAME is /home/cis90/$dir
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 $
```

Activity

Copy the example script in the depot directory to your bin directory:

```
cd bin
cp ~/../depot/scripts/example808 .
```

View the script:

```
vi example808
```

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

Prepare and run it

```
chmod +x example808
example808
```

Let me know in the chat window when you have finished

Scripting Tips

Pulling integers from files and adding them

Arithmetic with let command

```
/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

Arithmetic 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
```

Class Exercise

Copy the example script in the depot directory to your bin directory:

```
cd bin
cp ~/../depot/scripts/example9881 .
```

Modify the array with new names of your own choosing:

```
vi example9881
```

```
names[0]="make up your own name here"
names[1]="make up your own name here"
names[2]="make up your own name here"
names[3]="make up your own name here"
```

Prepare and run it:

```
chmod +x example9881
example9881
```

Let me know in the chat window when you have finished

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
```

Activity

Copy the example script in the depot directory to your bin directory:

```
cd bin
cp ~/../depot/scripts/example9872 .
```

View the script:

```
vi example9872
```

Prepare and run it:

```
chmod +x example9872
example9872
```

Let me know in the chat window when you have finished

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

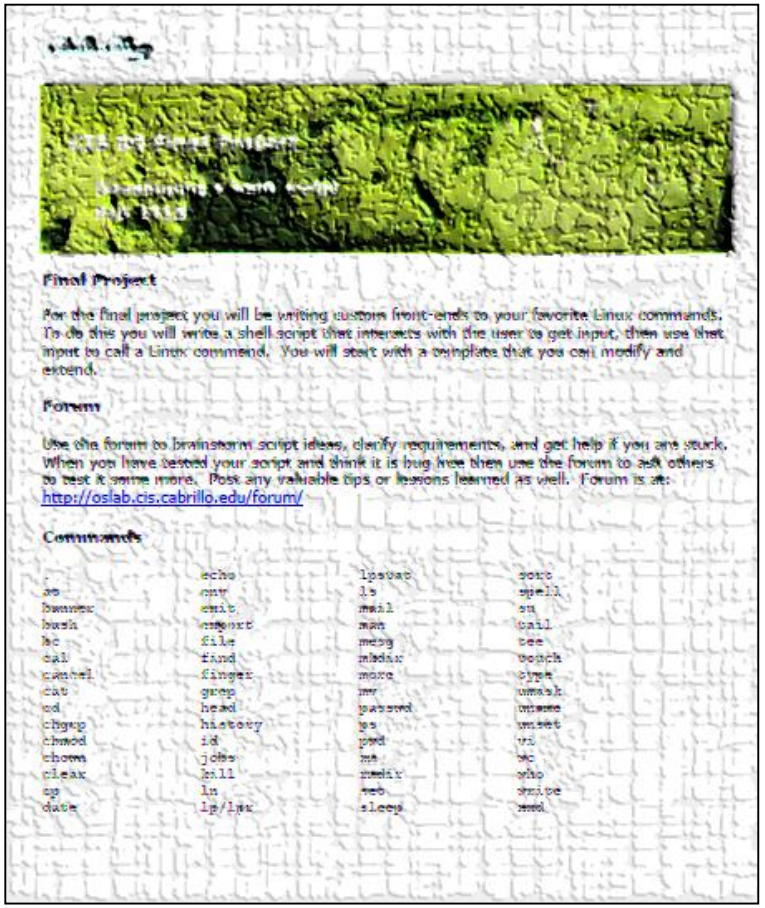
```
/home/cis90/simben/bin $ ./example6599
Enter username: milhom90
Username: milhom90
  Name = Homer Miller
  Shell = /bin/bash
```


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	seq	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 some more progress on your project today.
- Score your project using the Final Project rubric.

Implementing all five tasks (6 points each): • Requirements for each task: – Minimum of 12 "original" lines of bash script – Has one or more non-generic comments to explain what it is doing – Has user interaction
At least six bash constructs from this list: • Redirecting stdin (4 points) • Redirecting stdout (4 points) • Redirecting stderr (4 points) • Use of permissions (4 points) • Use of filename expansion characters (4 points) • Use of absolute path (4 points) • Use of relative path (4 points) • Use of a PID (4 points) • Use of inodes (4 points) • Use of links (4 points) • Use of color (4 points) • Use of scheduling (4 points) • Use of a GID or group (4 points) • Use of a UID or user (4 points) • Use of a /dev/tty device (4 points) • Use of a signal (4 points) • Use of piping (4 points) • Use of an environment variable (4 points) • Use of /bin/mail (4 points) • Use of a conditional (4 points) • Use of $(*command*) The maximum for this section is 24 points.

End Meeting

Backup