Lesson Module Checklist
- Slides
- WB

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

- Dog script examples ready

- Materials uploaded
- Backup slides, CCC info, handouts on flash drive
- Check that backup room headset is charged
- Spare 9v battery for mic

Introductions and Credits

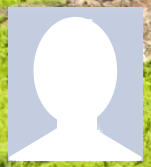
Jim Griffin
- Created this Linux course
- Created Opus and the CIS VLab
- Jim's site: http://cabrillo.edu/~jgriffin/

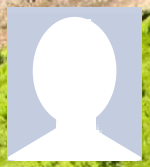
Rich Simms
- HP Alumnus
- Started teaching this course in 2008 when Jim went on sabbatical
- Rich's site: http://simms-teach.com

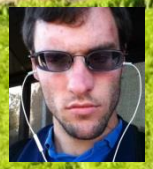
And thanks to:
- John Govsky for many teaching best practices: e.g. the First Minute quizzes, the online forum, and the point grading system (http://teacherjohn.com/)

CIS 90 - Lesson 14

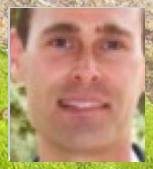
Instructor: **Rich Simms**
Dial-in: **888-450-4821**
Passcode: **761867**


Aaron


Andrew B.


Andrew C.


Arthur

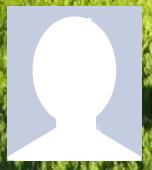

Brian


Cory


Daniel


David G.


Dave L.

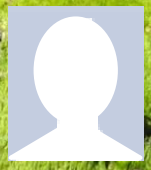

David P.


Debbie

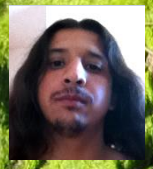

Edtson


Fidel

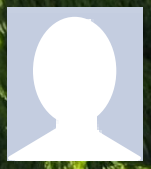

Humberto


Hunter


Imara


Ismael


Jessica


Joseph


Juliana

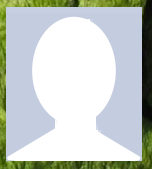

Lucie

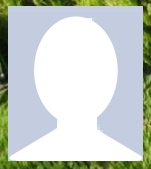

Marc

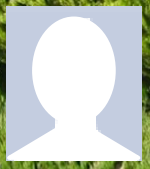

Marty

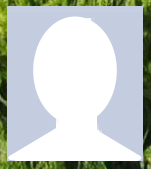

Matt


Michael

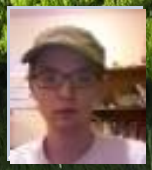

Rochelle

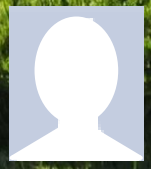

Shawn

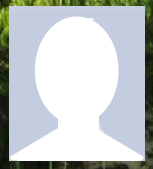

Tabitha


Taylor


Tyler

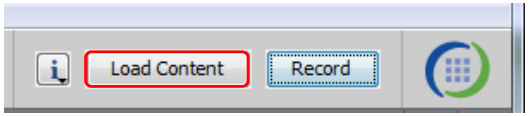

Will


Zachary


Zsolt

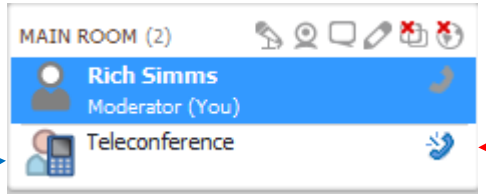
Email me (risimms@cabrillo.edu) a relatively current photo of your face for 3 points extra credit

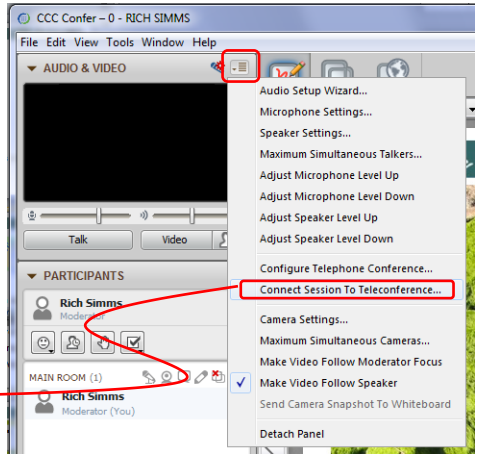
[ ] Preload White Board with *cis*lesson??*-WB*



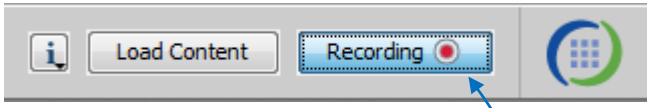
[ ] Connect session to Teleconference

Session now connected to teleconference





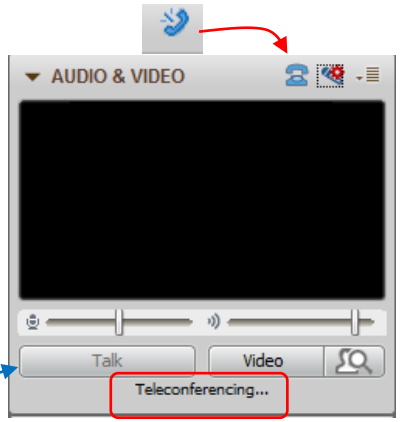
[ ] Is recording on?



Red dot means recording

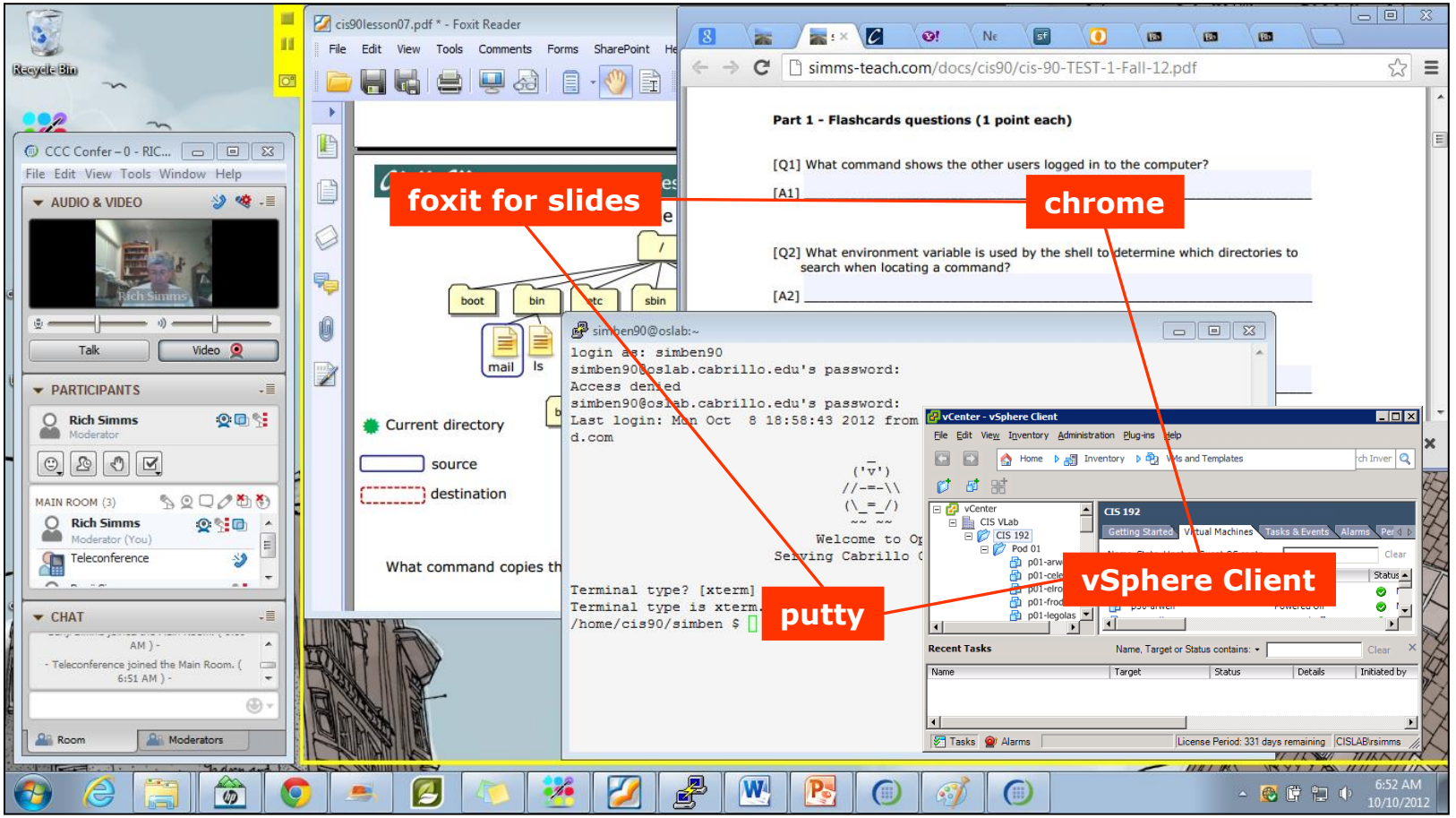
[ ] Use teleconferencing, not mic

Should be greyed out



CCC Confer

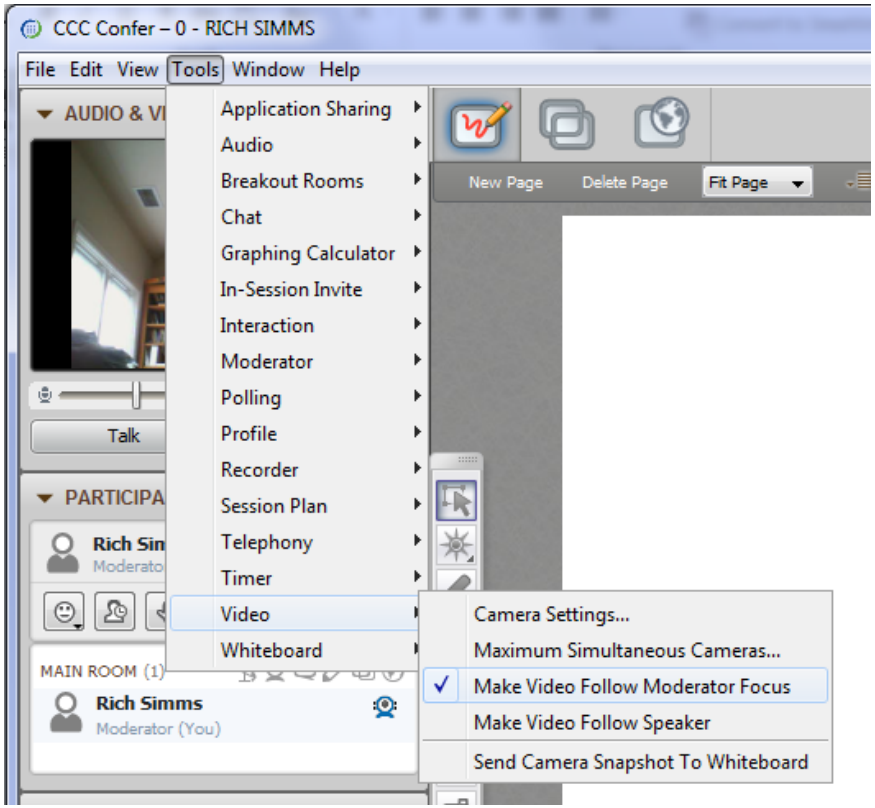
- [ ] **Video (webcam) optional**
- [ ] **layout and share apps**



[ ] Video (webcam) optional

[ ] Follow moderator

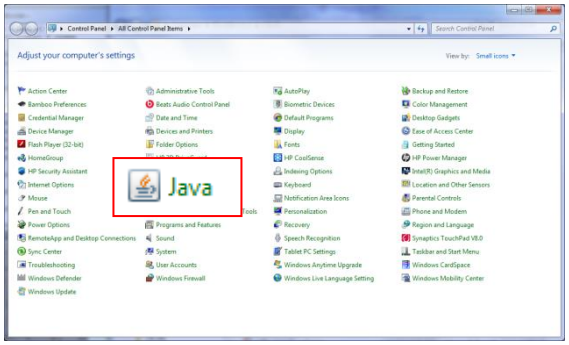
[ ] Double-click on postages stamps



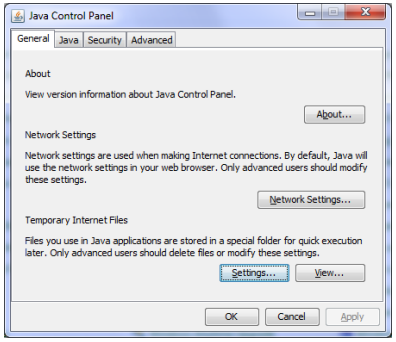
Universal Fix for CCC Confer:

1) Shrink (500 MB) and delete Java cache
2) Uninstall and reinstall latest Java runtime

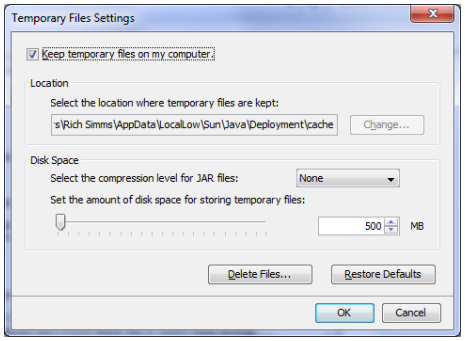
Control Panel (small icons)



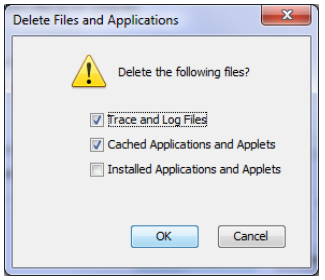
General Tab > Settings…



500MB cache size



Delete these



Google Java download

Quiz

No Quiz Today !

More Shell Scripting

Objectives	Agenda
• Use conditionals in scripts • Transfer files between computers • Archive directories using tar	• No Quiz • Questions from last week • scp • Tarballs • Getting started (if you haven't already) • Scripting tips • Wrap up

Questions

Questions?

Lesson material?

Labs? Tests?

How this course works?

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

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

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

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

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 copies 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>*

Copying files from a remote system

scp command syntax:

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

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

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

*When copying files from a remote system it is necessary to specify non-standard **ports** and **login credentials** for the remote system*

Copying files from remote system

```
/home/cis90/simben/bin $ hostname
son-of-opus.cishawks.net
/home/cis90/simben/bin $ ls -l myscript*
-rwxr-xr-x. 1 simben90 cis90 10511 Nov 25 17:31 myscript
-rwxr-xr-x. 1 simben90 cis90 10561 Dec  2 07:38 myscript.v1
-rwxr-xr-x. 1 simben90 cis90 11109 Dec  2 07:44 myscript.v2
-rwxr-xr-x. 1 simben90 cis90 11807 Dec  2 07:50 myscript.v3
```

```
/home/cis90/simben/bin $ hostname
oslab.cishawks.net
/home/cis90/simben/bin $ ls -l myscript*
-rwxr-xr-x. 1 simben90 cis90 10511 Nov 25 17:31 myscript
```

We want to copy the new files we made on Son-of-Opus over to Opus

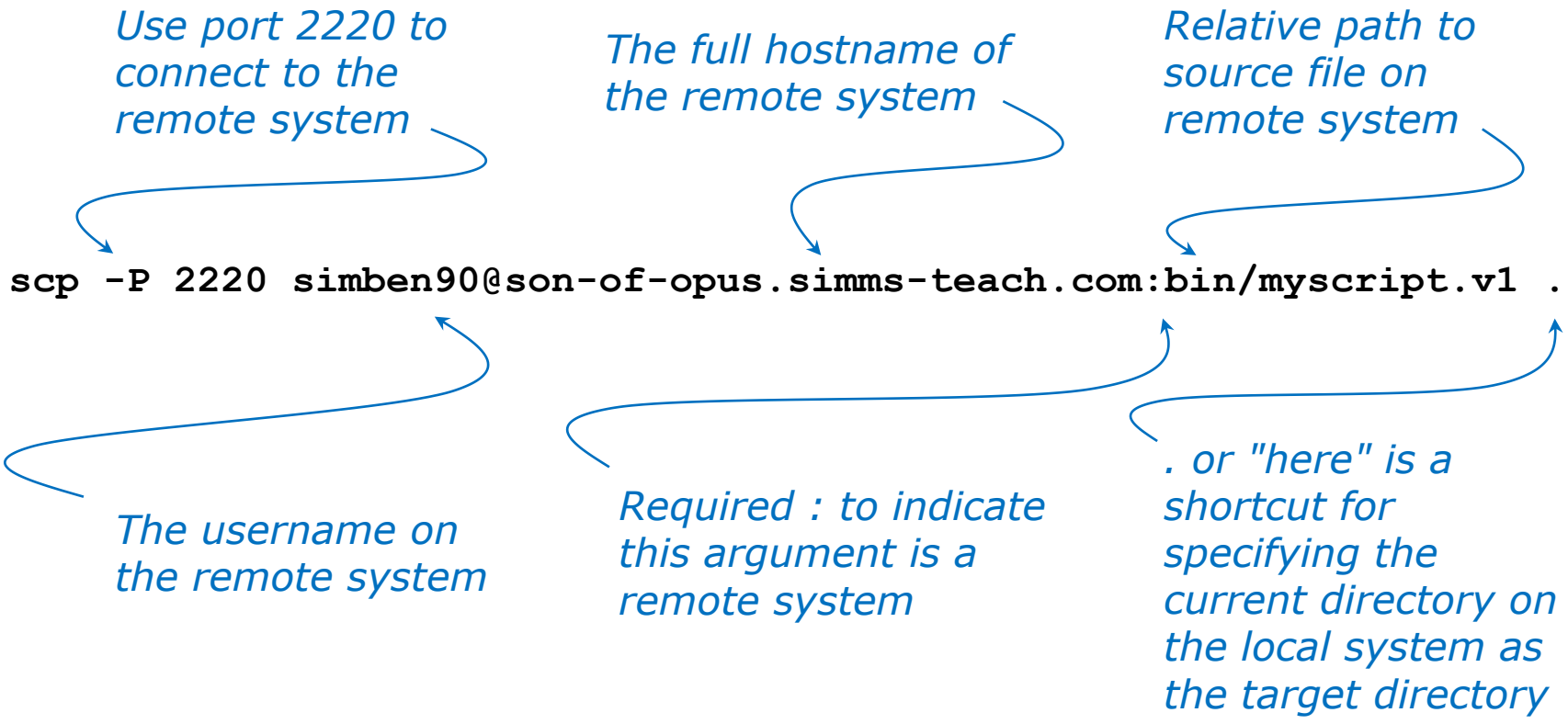
Copying a single file from a remote system

scp -P 2220 simben90@son-of-opus.simms-teach.com:bin/myscript.v1 .

```
/home/cis90/simben/bin $ hostname
oslab.cishawks.net
/home/cis90/simben/bin $ scp -P 2220 simben90@son-of-opus.simms-teach.com:bin/myscript.v1 .
The authenticity of host '[son-of-opus.simms-teach.com]:2220 ([54.215.232.67]:2220)' can't be
established.
RSA key fingerprint is 77:4c:65:4c:c8:70:c6:3f:00:cf:22:99:b6:a1:9a:a3.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[son-of-opus.simms-teach.com]:2220' (RSA) to the list of known
hosts.
simben90@son-of-opus.simms-teach.com's password:  <enter remote password>
myscript.v1                                   100%   10KB  10.3KB/s   00:00
/home/cis90/simben/bin $
/home/cis90/simben/bin $ ls -l myscript*
-rwxr-xr-x. 1 simben90 cis90 10511 Nov 25 17:31 myscript
-rwxr-xr-x. 1 simben90 cis90 10561 Dec  2 08:04 myscript.v1
/home/cis90/simben/bin $
```

From your bin directory on Opus, this command copies the file myscript.v1 on Son-of-Opus to Opus

Copying a single file from a remote system



Copying multiple files from a remote system

scp -P 2220 simben90@son-of-opus.simms-teach.com:bin/myscript.v* .

```
/home/cis90/simben/bin $ scp -P 2220 simben90@son-of-opus.simms-teach.com:bin/myscript.v* .
simben90@son-of-opus.simms-teach.com's password: <enter remote password>
myscript.v1                                   100%   10KB  10.3KB/s   00:00
myscript.v2                                   100%   11KB  10.9KB/s   00:00
myscript.v3                                   100%   12KB  11.5KB/s   00:00
/home/cis90/simben/bin $
/home/cis90/simben/bin $ ls -l myscript*
-rwxr-xr-x. 1 simben90 cis90 10511 Nov 25 17:31 myscript
-rwxr-xr-x. 1 simben90 cis90 10561 Dec  2 08:32 myscript.v1
-rwxr-xr-x. 1 simben90 cis90 11109 Dec  2 08:32 myscript.v2
-rwxr-xr-x. 1 simben90 cis90 11807 Dec  2 08:32 myscript.v3
/home/cis90/simben/bin $
```

From your bin directory on Opus, this command copies myscript.v1, myscript.v2 and myscript.v3 on Son-of-Opus to Opus

Copying multiple files from a remote system

Use port 2220 to connect to the remote system

The full hostname of the remote system

*Relative path using filename expansion character * to specify multiple source files on remote system*

```
scp -P 2220 simben90@son-of-opus.simms-teach.com:bin/myscript.v* .
```

The username on the remote system

Required : to indicate this argument is a remote system

. or "here" is a shortcut for specifying the current directory on the local system as the target directory

Class Activity

On Son-of-Opus:
If you haven't already, make a backup snapshot of your latest work:

```
hostname
cd bin
cp myscript myscript.vn
```

where n is your latest version number

On Opus:
Copy your latest version from Son-of-Opus:

```
hostname
cd bin
scp $LOGNAME@son-of-opus.simms-teach.com:bin/myscript.vn .
```

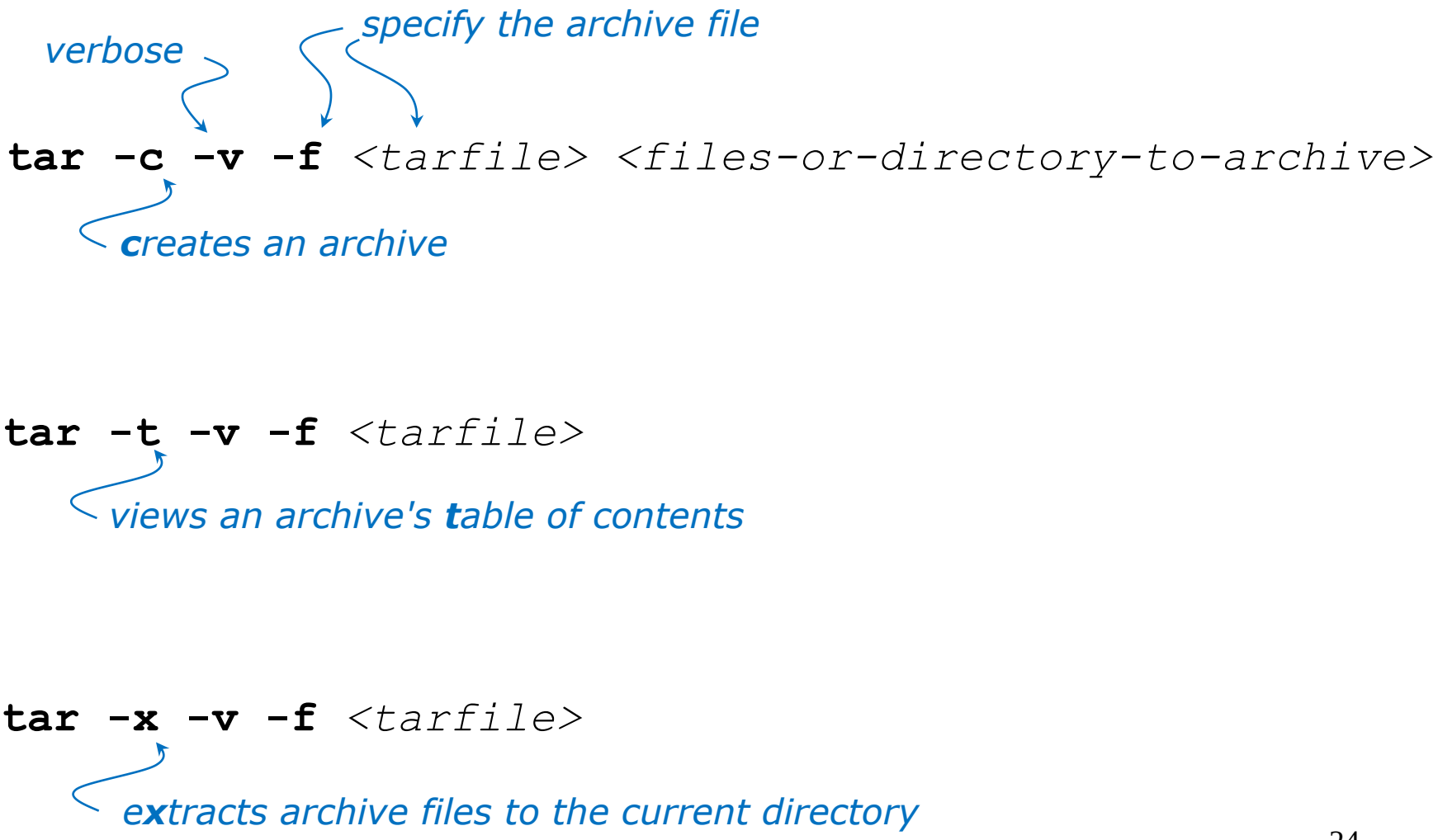
Note: This will overwrite any file named myscript.v*n* on Opus

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 $
```

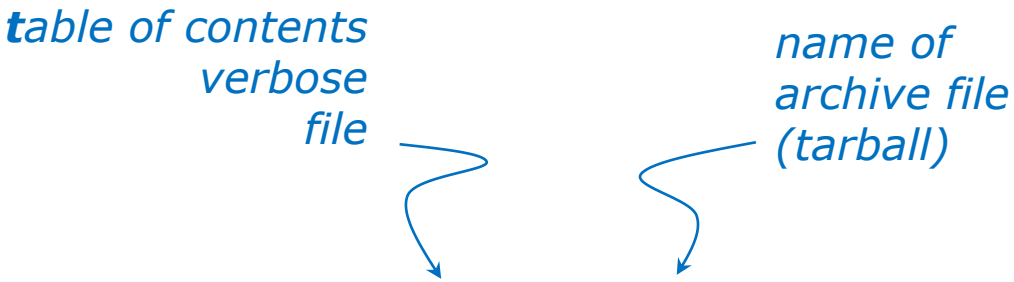
pathname to directory to archive

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  myscript.v3  tryme
banner  enlightenment   home  myscript   myscript.v2  treed        zoom
/home/cis90/simben $ tar cvf bin.tar bin/
bin/
bin/enlightenment
bin/treed
bin/zoom
bin/myscript.v1
bin/myscript.v3
bin/app
bin/home
bin/hi
bin/myscript
bin/myscript.v2
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-r--. 1 simben90 cis90 71680 Dec  2 10:41 bin.tar
/home/cis90/simben $ tar tvf bin.tar
drwxr-x--- simben90/cis90    0 2013-12-02 10:40 bin/
-r-xr-xr-- simben90/cis90 3442 2013-09-02 07:43 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
-rwxr-xr-x simben90/cis90 10561 2013-12-02 08:32 bin/myscript.v1
-rwxr-xr-x simben90/cis90 11807 2013-12-02 08:32 bin/myscript.v3
-r-xr-x--- simben90/cis90   220 2004-04-22 18:51 bin/app
-rwxr-xr-x simben90/cis90   109 2013-11-16 18:53 bin/home
-r-xr-x--- simben90/cis90   107 2001-07-20 21:06 bin/hi
-rwxr-xr-x simben90/cis90 10511 2013-11-25 17:31 bin/myscript
-rwxr-xr-x simben90/cis90 11109 2013-12-02 08:32 bin/myscript.v2
-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 2013-09-22 16:11 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

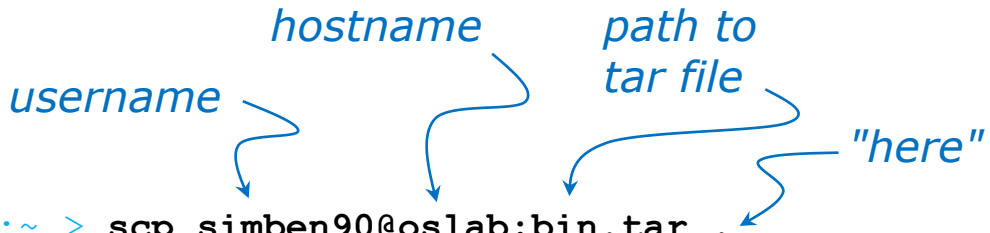
username *hostname*

```
/home/cis90/simben $ ssh cis90@p06-arwen
cis90@p06-arwen's password:
Welcome to Linux Mint 15 Olivia (GNU/Linux 3.8.0-26-generic x86_64)

Welcome to Linux Mint
 * Documentation:  http://www.linuxmint.com
Last login: Sun Sep  1 19:48:05 2013
cis90@p06-arwen:~ >
```

Login to your Arwen VM from Opus

Example

Copy archived directory to another system




```
cis90@p06-arwen:~ > scp simben90@oslab:bin.tar .
The authenticity of host 'oslab (2607:f380:80f:f425::230)' can't be
established.
RSA key fingerprint is 7d:32:80:b9:52:32:c8:dc:3b:16:0e:ba:8c:fd:79:ef.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'oslab,2607:f380:80f:f425::230' (RSA) to the
list of known hosts.
simben90@oslab's password:
bin.tar                                       100%   70KB  70.0KB/s
00:00
cis90@p06-arwen:~ > ls -l bin.tar
-rw-rw-r-- 1 cis90 cis90 71680 Dec  2 10:47 bin.tar
cis90@p06-arwen:~ >
```

Note how archive files are shown in red

Copy your bin archive from Opus to Arwen

Example

Copy archived directory to another system

Extract your bin directory to your Arwen home directory

```
cis90@p06-arwen:~ > ls
bin.tar  Desktop  Documents  Downloads  Music  Pictures  Projects  Videos
cis90@p06-arwen:~ > tar xvf bin.tar
bin/
bin/enlightenment
bin/treed
bin/zoom
bin/myscript.v1
bin/myscript.v3
bin/app
bin/home
bin/hi
bin/myscript
bin/myscript.v2
bin/I
bin/tryme
bin/datecal
bin/banner
cis90@p06-arwen:~ >
```

extract verbose file

name of archive file (tarball)

Example

Copy archived directory to another system

Run myscript file in the bin directory

```
cis90@p06-arwen:~ > myscript
No command 'myscript' found, did you mean:
 Command 'pyscript' from package 'python-pyscript' (universe)
myscript: command not found
cis90@p06-arwen:~ >
```

Oops … since the bin directory is not on our path we must specify the path manually

```
cis90@p06-arwen:~ > bin/myscript
bin/myscript: line 44: finger: command not found
What is your first name? ^C
cis90@p06-arwen:~ >
```

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

Example

Copy archived directory to another system

Install the finger command on Arwen

```
cis90@p06-arwen:~ > su -
Password:
p06-arwen ~ # 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 368 not upgraded.
Need to get 17.3 kB of archives.
After this operation, 68.6 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu/ raring/universe finger amd64
0.17-15 [17.3 kB]
Fetched 17.3 kB in 0s (31.3 kB/s)
Selecting previously unselected package finger.
(Reading database ... 143228 files and directories currently installed.)
Unpacking finger (from .../finger_0.17-15_amd64.deb) ...
Processing triggers for man-db ...
Setting up finger (0.17-15) ...
p06-arwen ~ # exit
```

Become root and install finger command

Exit root session back to cis90 user

Example

Copy archived directory to another system

Run myscript file in the bin directory

```
cis90@p06-arwen:~ > bin/myscript

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

    1) What is today?
    2) The users on p06-arwen
    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: Q
cis90@p06-arwen:~ >
```

Note: Linux Mint uses .profile instead of .bash_profile. It checks to see if you have a local bin directory and adds it to your path if there is one. Next time you log in just type myscript as it will be on your path!

Housekeeping

Next Class

Project is due next week!

1. No labs due today

2. There is a check script for Lab X2

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

4. Two weeks from now (see calendar)
   - Final Exam (Test #3) 1-3:50PM
   - Extra credit labs are due by 11:59PM .

Make backup copies of your script

change, change, change, change, … rest

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

change, change, change, change, … rest

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

change, change, change, change, … rest

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

Spring 2014 Linux Classes

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: CS 1L or CIS 172.

Transfer Credit: CSU.

Section	Days	Times	Units	Instructor	Room
82750	W	09:00AM-12:05PM	3.00	R.Simms	OL

Section 82750 is an ONLINE course. Meets weekly throughout the semester online at the scheduled times by remote technology using CCC Confer. For details, see instructor's web page at go.cabrillo.edu/online.

CIS 98 UNIX/Linux Shell Programming

Presents an introduction to shell programming in a UNIX/Linux environment, and is designed for system administrators or technical users with little or no programming background. Prerequisite: CIS 90.

Transfer Credit: CSU.

Section	Days	Times	Units	Instructor	Room
83707	TH	11:00AM-02:05PM	4.00	M.Matera	829
&	Arr.	Arr.		M.Matera	OL

Section 83707 is a Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 5 hr 5 min online lab per week. Students will be required to show that they meet the course prerequisites. For details, see instructor's web page at go.cabrillo.edu/online.

CIS 192AB UNIX/Linux Network Administration

Teaches the building of network infrastructures, and the installation, configuration, and protection services on Linux TCP/IP networks. Prerequisites: CIS 81 and CIS 90 or equivalent. Recommended Preparation: CIS 191AB.

Section	Days	Times	Units	Instructor	Room
82744	TH	05:30PM-09:35PM	4.00	M.Matera	OL
&	Arr.	Arr.		M.Matera	OL

Section 82744 is an ONLINE course. Meets weekly throughout the semester online at the scheduled times by remote technology using CCC Confer with an additional 2 hr 5 min online lab per week. Students will be required to show that they meet the course prerequisites. For details, see instructor's web page at go.cabrillo.edu/online.

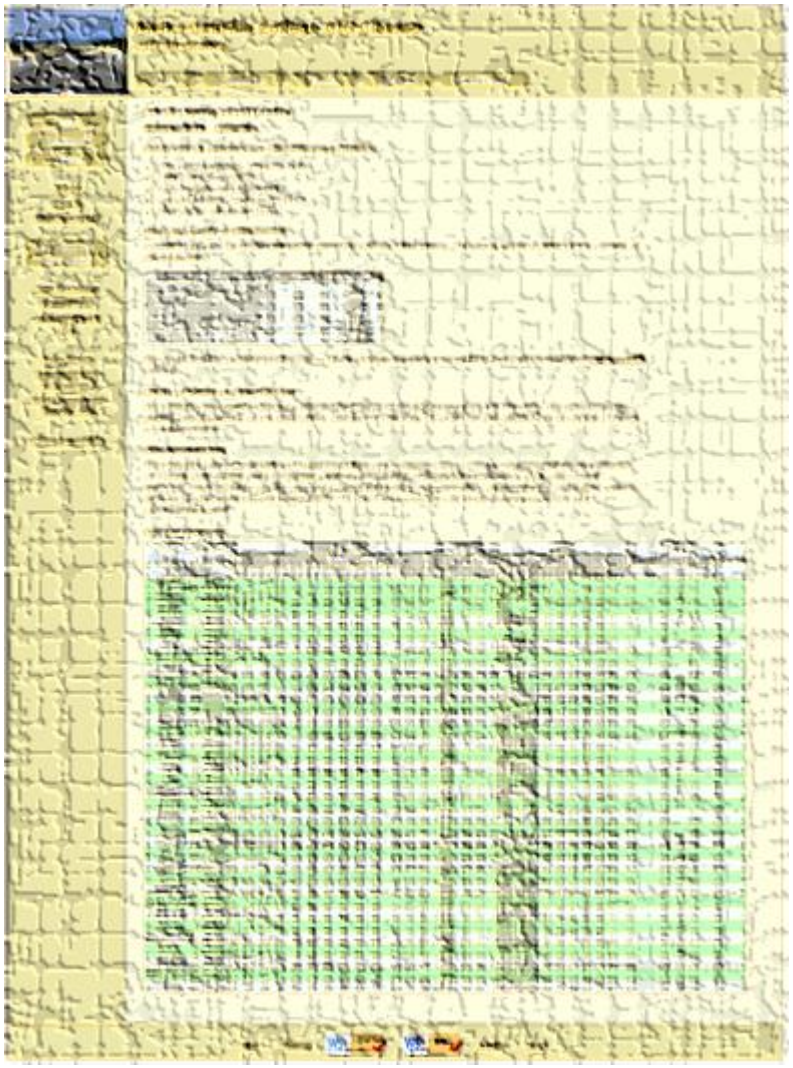
Final Exam

Test #3 (final exam)

- Must be face-to-face or proctored (not online using CCC Confer).
- We will be in room 828 on campus.
- Timed test (no 11:59PM grace period)
- Practice test will be available one week before

	12/17	**Test #3 (the final exam)** **Time** • 1:00PM - 3:50PM in Room 828 **Materials** • Presentation slides (download) • Test (download)		5 posts Lab X1 Lab X2

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



GRADES

- Check your progress on the Grades page
- If you haven't already, send me a student survey to get your LOR secret code name
- Graded labs & tests are placed in your home directories on Opus
- Answers to labs, tests and quizzes are in the */home/cis90/answers* directory on Opus

Current Point Tally

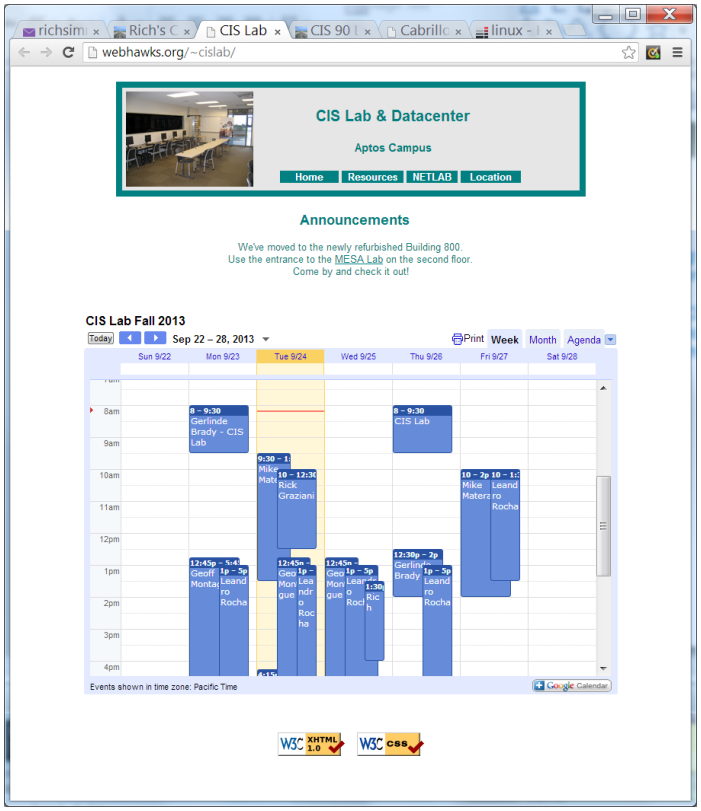
As of 12/01/2013

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**

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

adaldrida: 95% (430 of 450 points)
anborn: 0% (0 of 450 points)
aragorn: 95% (428 of 450 points)
arwen: 79% (357 of 450 points)
balrog: 33% (150 of 450 points)
barliman: 0% (4 of 450 points)
beregond: 74% (336 of 450 points)
boromir: 5% (25 of 450 points)
celebrian: 79% (359 of 450 points)
dori: 32% (146 of 450 points)
dwalin: 71% (320 of 450 points)
elrond: 92% (415 of 450 points)
eomer: 66% (299 of 450 points)
faramir: 95% (429 of 450 points)
frodo: 97% (438 of 450 points)
gimli: 93% (419 of 450 points)
goldberry: 103% (465 of 450 points)
ingold: 98% (442 of 450 points)
ioreth: 65% (295 of 450 points)
legolas: 72% (328 of 450 points)
marhari: 100% (450 of 450 points)
pallando: 101% (456 of 450 points)
pippen: 91% (413 of 450 points)
quickbeam: 26% (121 of 450 points)
samwise: 84% (379 of 450 points)
sauron: 101% (457 of 450 points)
shadowfax: 62% (280 of 450 points)
strider: 87% (394 of 450 points)
theoden: 99% (446 of 450 points)
treebeard: 88% (399 of 450 points)
tulkas: 82% (373 of 450 points)
ulmo: 69% (313 of 450 points)

CIS Lab Schedule
http://webhawks.org/~cislab/



Work on assignments together with other classmates

Get help from instructors and student lab assistants

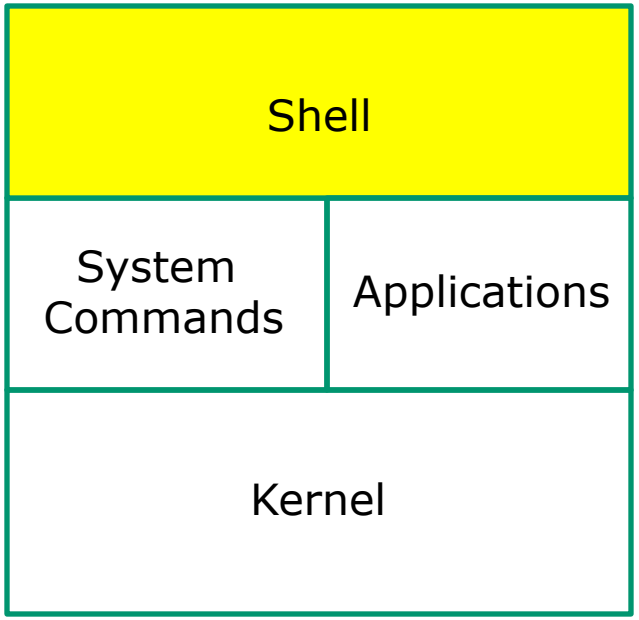
MESA grants requires logging help sessions with MESA funded student assistants

Michael, CIS 90 alumnus just hired, is now working mornings, Wednesday to Friday

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 (born again shell), csh (C shell), ksh (Korn shell).
- **A user interface and a programming language (scripts).**
- GNOME and KDE desktops could be called graphical shells

Shell Scripts

Some scripts on opus

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

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

Many commands are scripts

Which commands in /bin are really scripts?

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

How many commands in /bin are scripts?

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

Class Activity
Scripting

Of all the UNIX/Linux commands in:

/bin

/usr/bin

/sbin

/usr/sbin

How many are scripts?

Write your answer in the chat window

Project

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

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

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

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

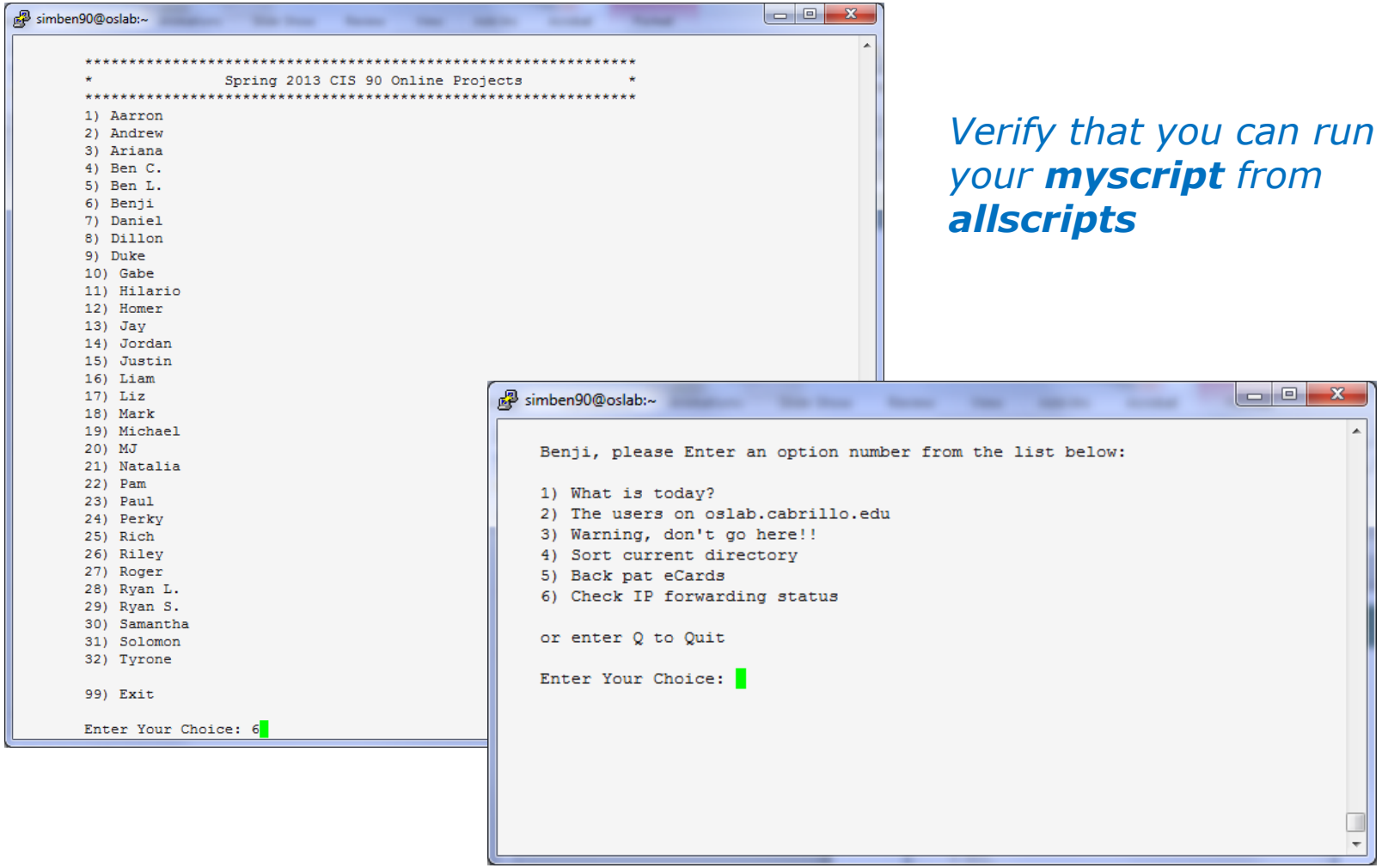
4. Do the example grep task shown in Lesson 13

Grading rubric (60 points maximum)

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

This applies to each individual task

This applies to the project as a whole



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

Don't forget to do this!

chmod 750 ~/bin/myscript

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

Project Status

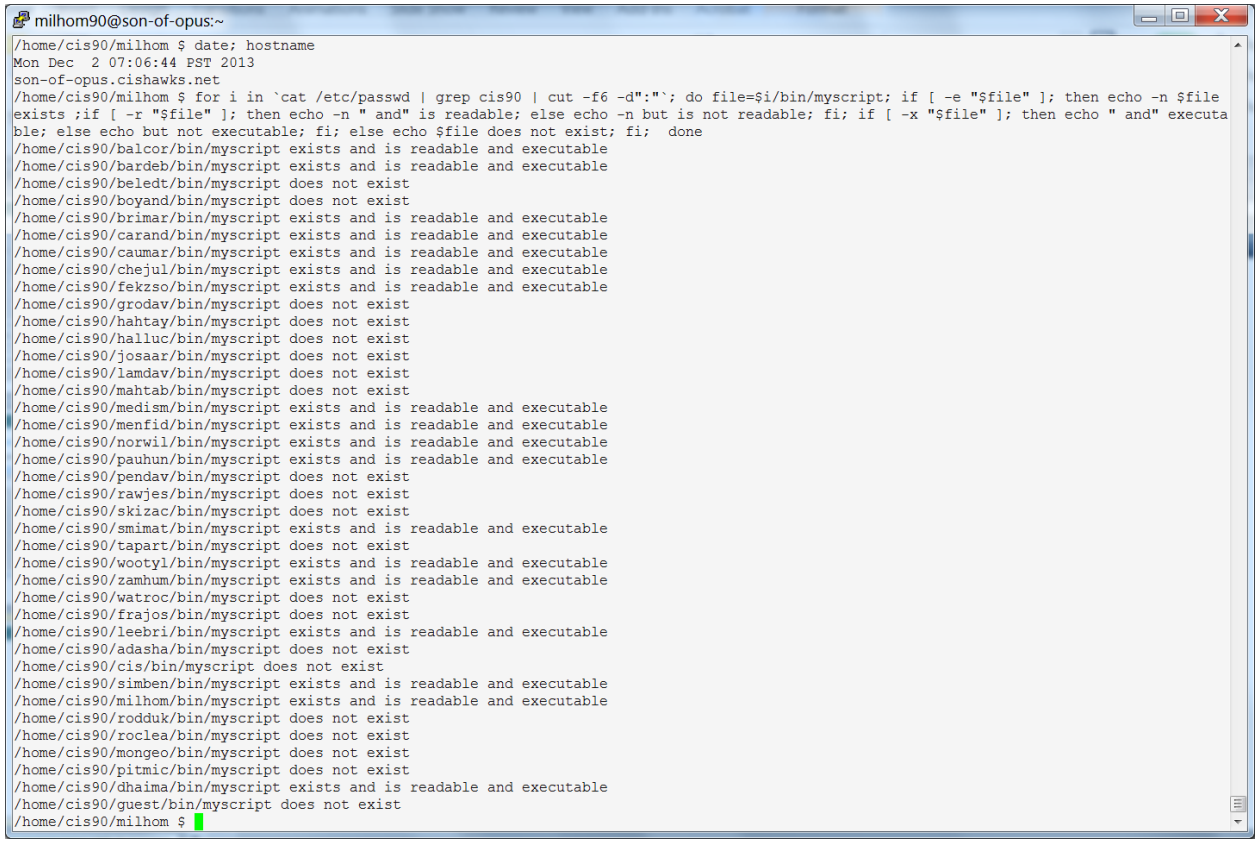
`ls -l /home/cis90/*/bin/myscript`

```
milhom90@son-of-opus:~
/home/cis90/milhom $ date; hostname
Mon Dec  2 06:55:11 PST 2013
son-of-opus.cishawks.net
/home/cis90/milhom $ ls -l /home/cis90/*/bin/myscript
-rwxr-x---. 1 balcor90 cis90   546 Nov 26 14:47 /home/cis90/balcor/bin/myscript
-rwxr-x---. 1 bardeb90 cis90   734 Nov 26 15:25 /home/cis90/bardeb/bin/myscript
-rwxrwxr-x. 1 brimar90 cis90   778 Nov 25 00:20 /home/cis90/brimar/bin/myscript
-rwxr-x---. 1 carand90 cis90  2011 Nov 26 21:56 /home/cis90/carand/bin/myscript
-rwxr-x---. 1 caumar90 cis90   576 Nov 26 15:59 /home/cis90/caumar/bin/myscript
-rwxr-x---. 1 chejul90 cis90  5194 Dec  2 01:25 /home/cis90/chejul/bin/myscript
-rwxr-x---. 1 dhaima90 cis90  1517 Dec  1 11:17 /home/cis90/dhaima/bin/myscript
-rwxr-x---. 1 fekzso90 cis90   554 Nov 26 16:29 /home/cis90/fekzso/bin/myscript
-rwxr-x---. 1 leebri90 cis90   921 Nov 19 15:55 /home/cis90/leebri/bin/myscript
-rwxr-x---. 1 medism90 cis90   546 Nov 26 14:22 /home/cis90/medism/bin/myscript
-rwxr-x---. 1 menfid90 cis90   697 Nov 26 15:43 /home/cis90/menfid/bin/myscript
-rwxr-xr-x. 1 milhom90 cis90  4535 Dec  1 17:59 /home/cis90/milhom/bin/myscript
-rwxr-x---. 1 norwil90 cis90  4499 Nov 30 16:18 /home/cis90/norwil/bin/myscript
-rwxr-xr-x. 1 pauhun90 cis90  1358 Nov 26 17:03 /home/cis90/pauhun/bin/myscript
-rwxr-xr-x. 1 simben90 cis90 10511 Nov 25 17:31 /home/cis90/simben/bin/myscript
-rwxr-x---. 1 smimat90 cis90   546 Nov 26 13:15 /home/cis90/smimat/bin/myscript
-rwxrwxr-x. 1 wootyl90 cis90   734 Nov 19 14:49 /home/cis90/wootyl/bin/myscript
-rwxr-x---. 1 zamhum90 cis90   736 Nov 19 14:49 /home/cis90/zamhum/bin/myscript
/home/cis90/milhom $
```

Is your script "hackable" by others classmates?

Project Status

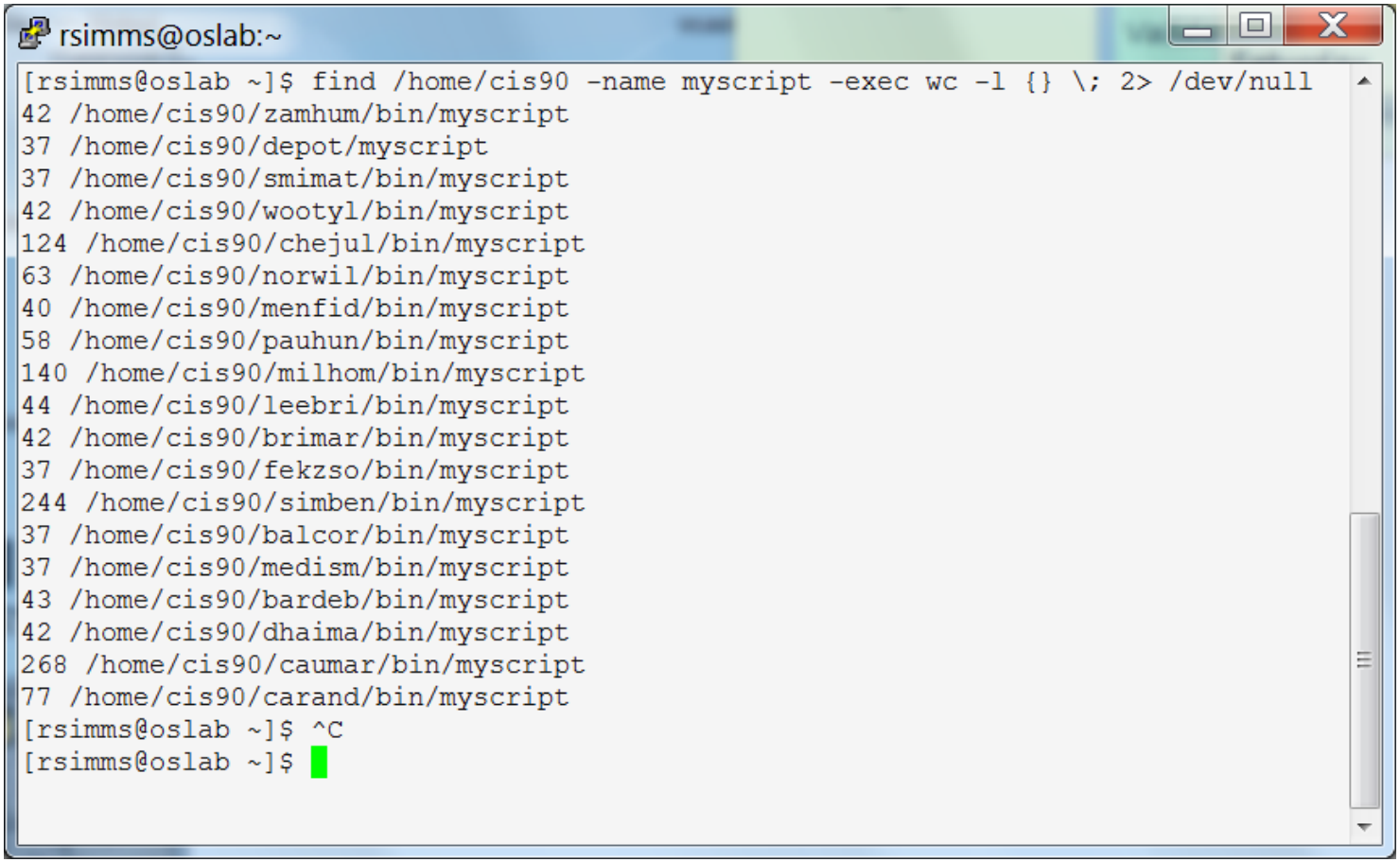
```
for i in `cat /etc/passwd | grep cis90 | cut -f6 -d":"`; do
file=$i/bin/myscript; if [ -e "$file" ]; then echo -n $file exists ;if [ -r
"$file" ]; then echo -n " and" is readable; else echo -n but is not
readable; fi; if [ -x "$file" ]; then echo " and" executable; else echo but
not executable; fi;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



a one line command using semi-colons!

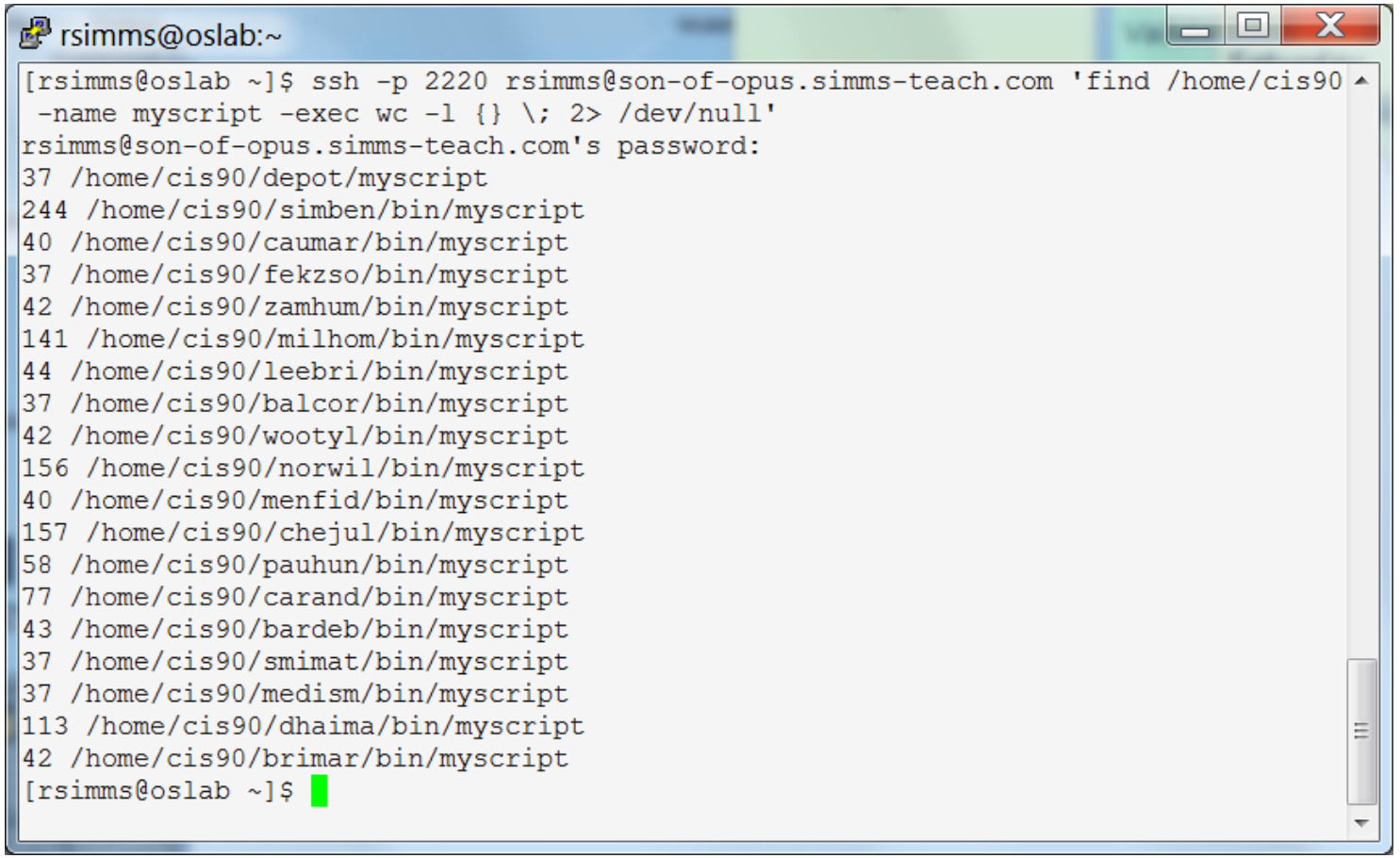
Project Status

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



Project Status

```
ssh -p 2220 rsimms@son-of-opus.simms-teach.com 'find /home/cis90 -
name myscript -exec wc -l {} \; 2> /dev/null'
```



Using ssh to run a command on a remote system

Review

```
function runningScript ()
{
```

The rules of the road for variables

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

Running a Script

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

Add this line to the last script we made

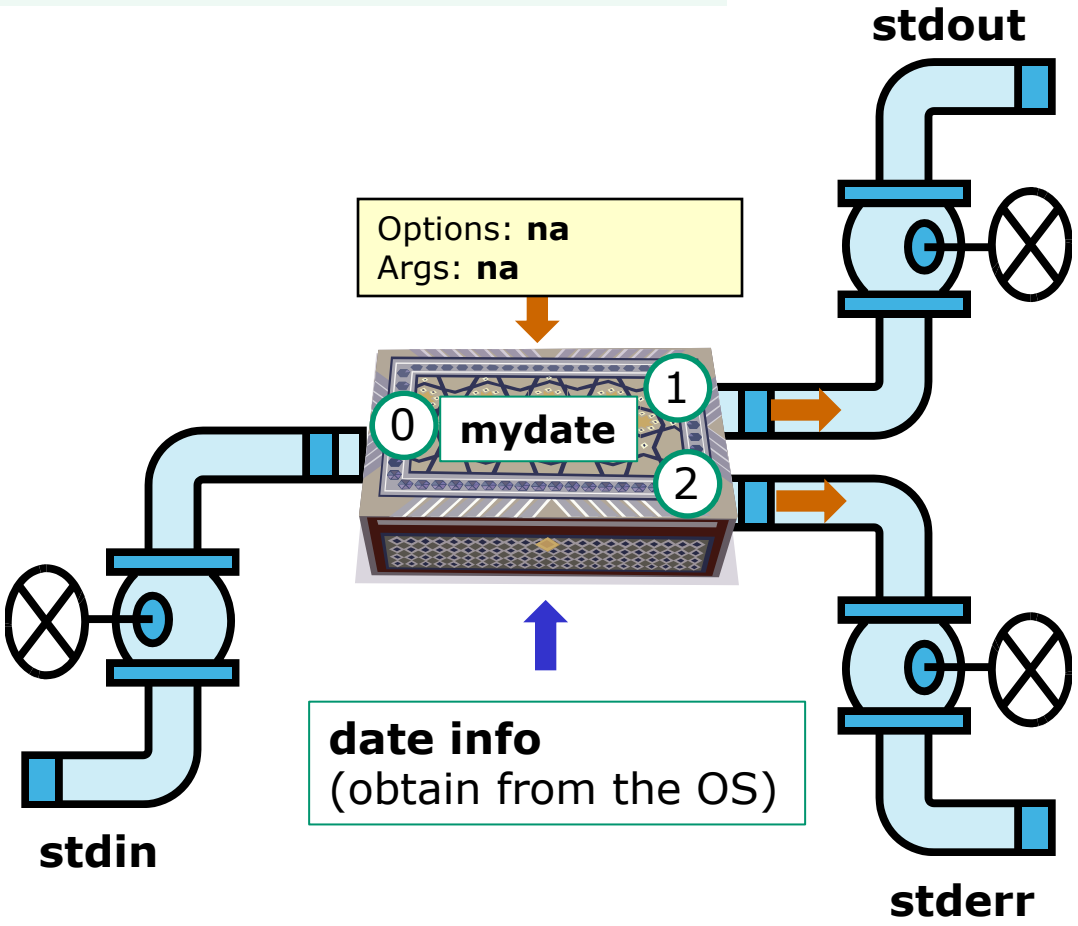
Don't initialize them yet

```
/home/cis90/simben $ mydate
Hola simben90
05/16/2013

/home/cis90/simben $
```

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

Running a Script

$ mydate



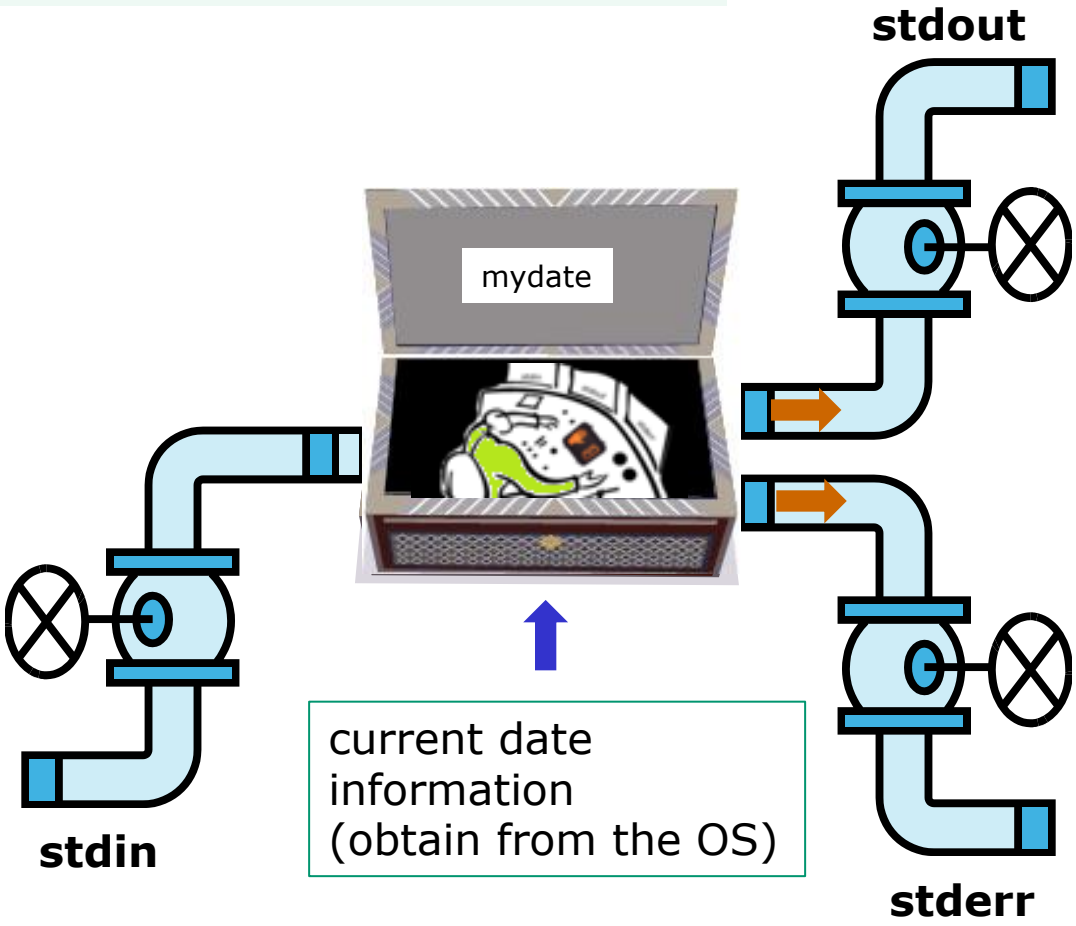

```
Hola simben90
05/16/2013
```

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

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

Running a Script

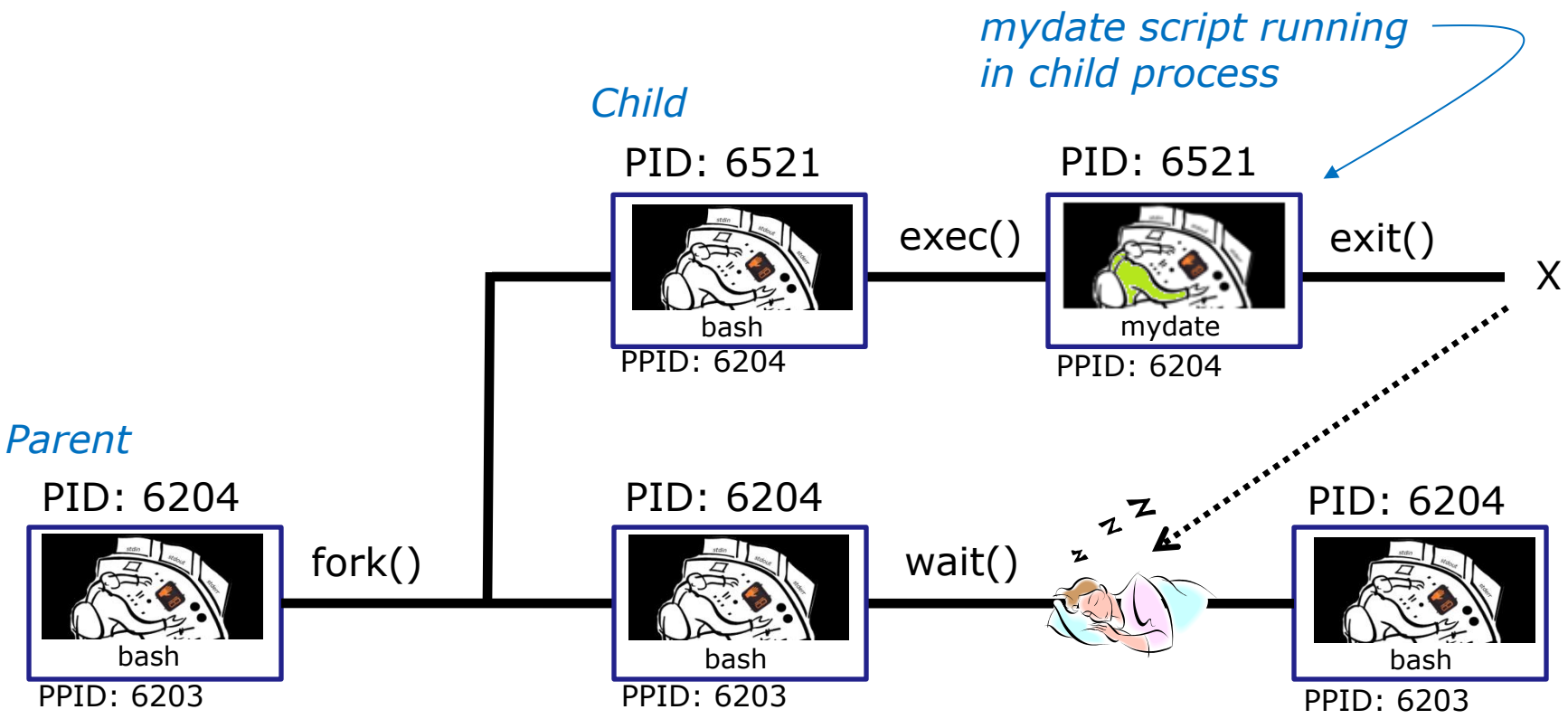
$ mydate




```
Hola simben90
05/16/2013
```

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

Running a Script



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

Running a Script

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

In the parent process, initialize the three variables

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

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

Running a Script

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

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

/home/cis90/simben $ mydate
Hola simben90
05/09/2012

/home/cis90/simben $
```

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

Why no Tic Tac Toe output?

Running a Script

```
/home/cis90/simben $ export myvar1
/home/cis90/simben $ mydate
Hola simben90
05/09/2012
Tic

/home/cis90/simben $ export myvar2
/home/cis90/simben $ mydate
Hola simben90
05/09/2012
Tic Tac

/home/cis90/simben $ export myvar3
/home/cis90/simben $ mydate
Hola simben90
05/09/2012
Tic Tac Toe
```

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

Running a Script

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

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

/home/cis90/simben $ mydate
Hola simben90
05/09/2012
Tic Tac Toe
red white blue

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

Add these new lines

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

Running a Script

Unless we want them to

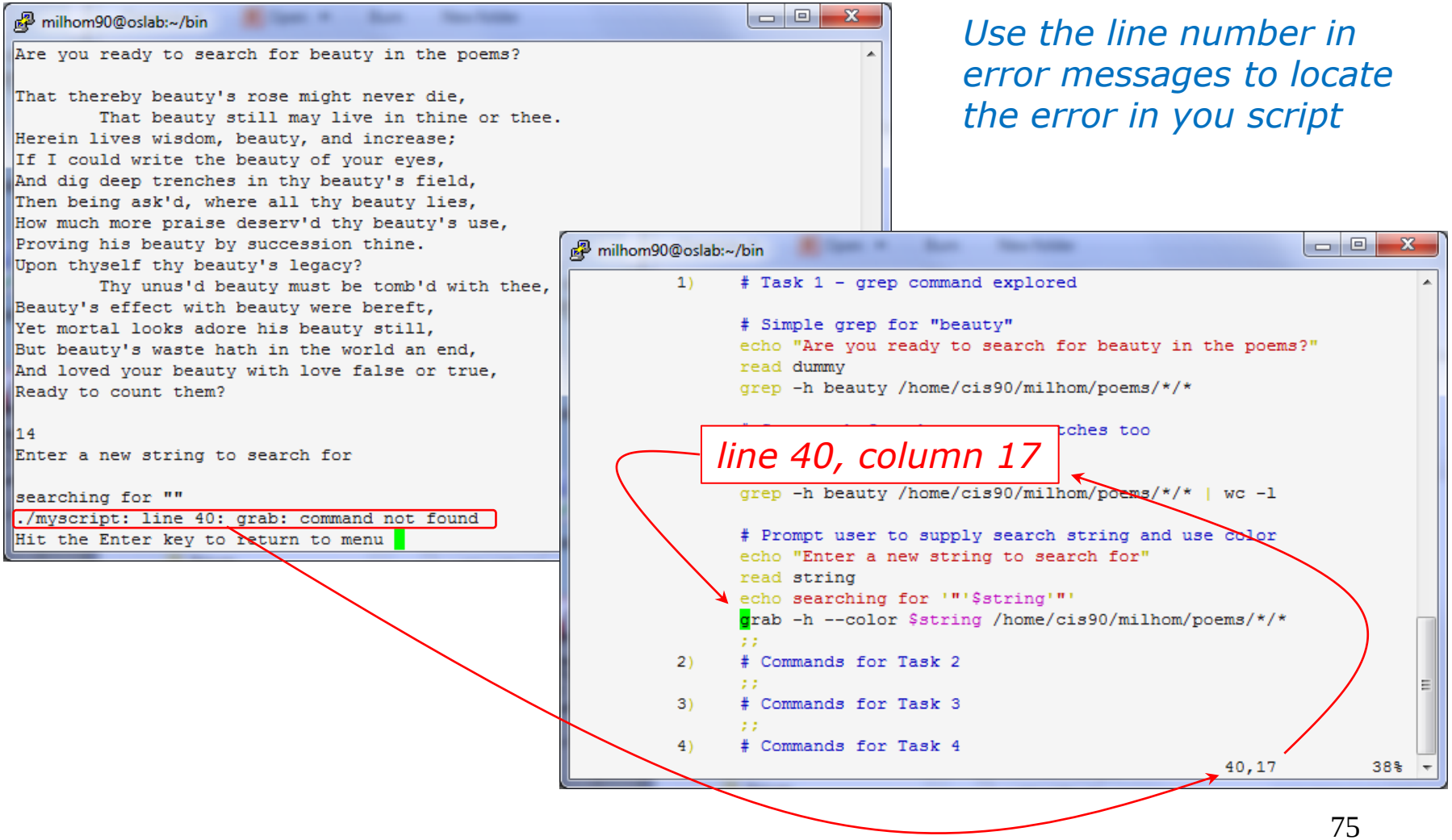
```
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3
Tic Tac Toe

/home/cis90/simben $ source mydate
Hola simben90
05/09/2012
Tic Tac Toe
red white blue

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

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

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

Scripting Tips vi

Line Numbers in errors and vi



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

Color Syntax

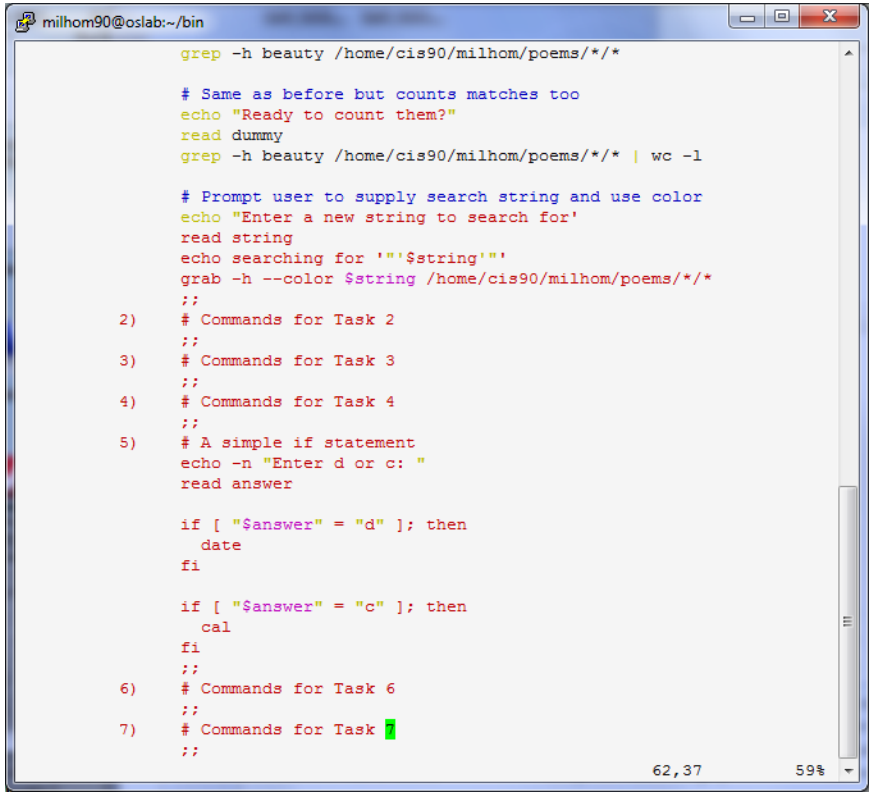
```
milhom90@oslab:~/bin
/home/cis90/milhom/bin $ ./myscript
./myscript: line 79: unexpected EOF while looking for matching `"'
./myscript: line 83: syntax error: unexpected end of file
/home/cis90/milhom/bin $
```

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

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

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

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

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

Use color syntax to spot unmatched quotes

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

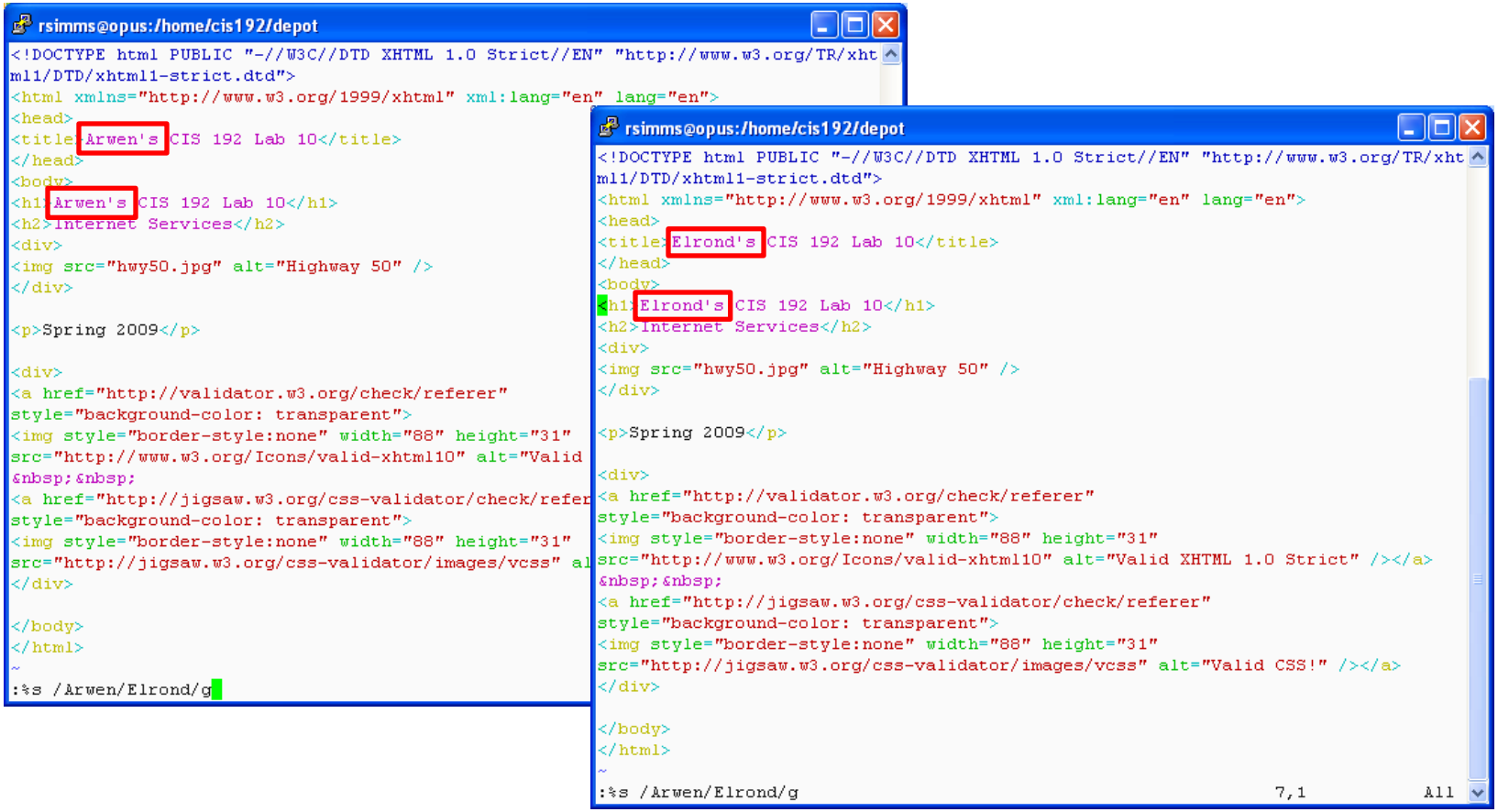
Color Syntax



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

Global search and replace with vi

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



Scripting Tips

sleep

(adding timed pauses)

Class Exercise

Make a new script in your bin directory

```
cd bin
vi example911
```

In vi add these lines then save:

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

Prepare and run your script

```
chmod +x example911
example911
```

Scripting Tips

$(cmd) and `cmd`

Shell Scripts

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

For example:

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

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

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

Using back tics around the command to evaluate

Shell Scripts

Sometimes you want to use the output of a command as an argument to another command

For example:

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

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

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

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

Class Activity
Scripting

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

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

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

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

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

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

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

Put your answer in the chat window

Scripting Tips

extracting a field from a record

/etc/passwd

```
[rsimms@opus ~]$ cat /etc/passwd
< snipped >
apache:x:48:48:Apache:/var/www:/sbin/nologin
simben90:x:1001:190:Benji Simms:/home/cis90/simben:/bin/bash
milhom90:x:1002:190:Homer Miller:/home/cis90/milhom:/bin/bash
rodduk90:x:1003:190:Duke Roddy:/home/cis90/rodduk:/bin/bash
< snipped >
```

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

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

myscript

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

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

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

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

myscript

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

Let's add a friendly Hello using the user logname

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

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

myscript

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

Now include the /etc/passwd info as well

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

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

myscript

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

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

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

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

myscript

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

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

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

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

myscript

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

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

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

        Enter Your Choice: 8
```

Hello Homer Miller

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

myscript

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

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

```
                Homer's CIS 90 Final Project
        1) Color
        2) My Find Command
        3) More practice
        4) Homer's friend made this one - Thank You
        5) Task 5
        6) Exit

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

Class Exercise

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

Make a new script in your bin directory

```
cd bin
vi example401
```

In vi add these lines then save:

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

Prepare and run your script

```
chmod +x example401
example401
```

Read your mail to view your new message

```
mail
```

Scripting Tips

simple if statement

myscript

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

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

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

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

        ;;
```

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

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

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

myscript

```
          Homer's CIS 90 Final Project
        1) Color
        2) My Find Command
        3) More practice
        4) Examples - test file attributes
        5) Examples - simple if statement
        6) Examples - logic
        10) Exit

        Enter Your Choice: 5
Enter d or c: d
Sun May 17 10:00:35 PDT 2009
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) Color
        2) My Find Command
        3) More practice
        4) Examples - test file attributes
        5) Examples - simple if statement
        6) Examples - logic
        10) Exit

        Enter Your Choice: 5
Enter d or c: c
      May 2009
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
Hit the Enter key to return to menu
```

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

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

Class Exercise

Run the previous example task

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

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

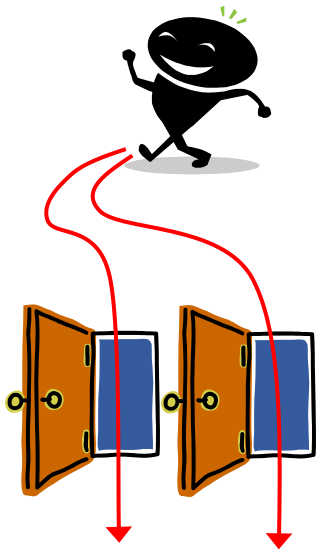
```
vi ~milhom90/bin/myscript
```

Scripting Tips

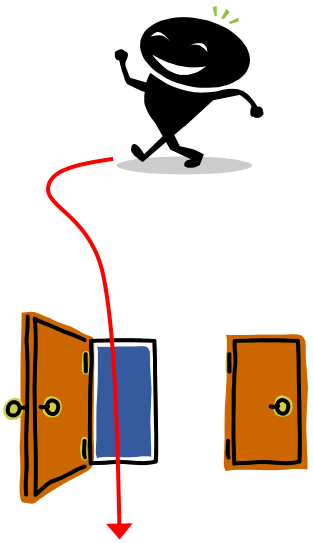
if statement with "or"

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

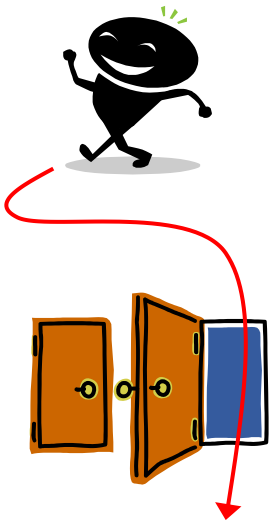
OR logic



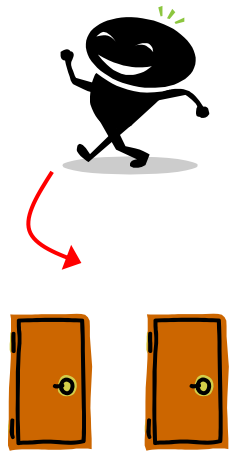
Yes



Yes



Yes



No

myscript

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

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

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

        ;;
```

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

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

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

myscript

```
          Homer's CIS 90 Final Project
      1) 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
      10) 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
```

date is run because user typed a d

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

myscript

```
          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
      10) Exit

      Enter Your Choice: 6
 Enter d or c: D
Wed May 20 05:07:38 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*

Class Exercise

Make a new script in your bin directory

```
cd bin
vi example654
```

In vi add these lines then save:

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

Prepare and run your script

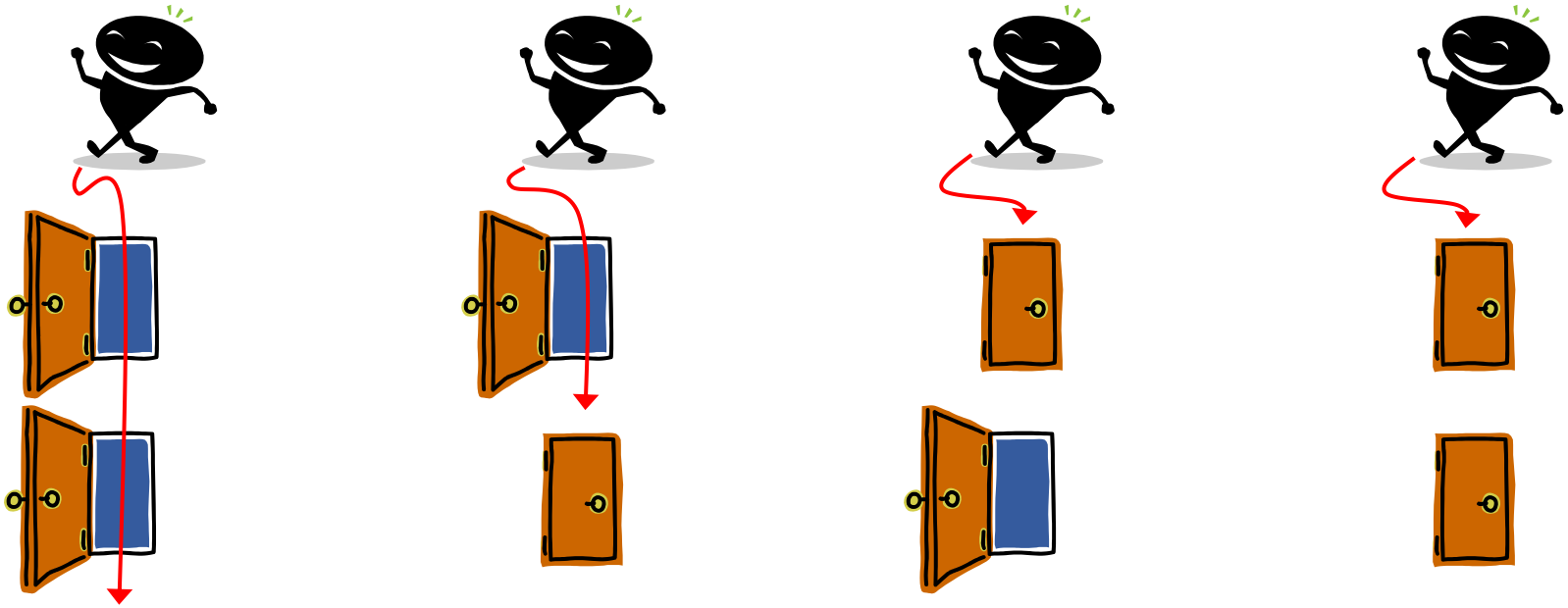
```
chmod +x example654
example654
```

Scripting Tips

if statements with "and"

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

AND logic



myscript

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

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

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

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

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

&& *means "and"*

myscript

```
 Homer's 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 - IF with OR logic
        7) Examples - IF with AND logic
        8) Examples - cut command to get name from /etc/passwd
        10) 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 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: 7
Is the furnace on or off? on
Is there a fire in the fireplace (yes or no)? no
It is warm in here
Hit the Enter key to return to menu
```

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

Class Exercise

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

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

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

Scripting Tips

if

file types

myscript

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

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

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

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

tests to see if it's a regular file

tests to see if it's a directory

myscript

```
        Homer's CIS 90 Final Project
1) 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
10) Exit

Enter Your Choice: 4
The files in this directory are:
1976.egg
Anon
Blake
Shakespeare
Yeats
Which file are you interested in? : 1976.egg
Here are some details about 1976.egg:
1976.egg: ASCII English text, with escape sequences
1976.egg is a regular file
Here is long listing of the 1976.egg file:
-rw-r--r-- 1 squid squid 734 Apr  8 10:01 1976.egg
Hit the Enter key to return to menu
```

myscript

```
        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
10) Exit

Enter Your Choice: 4
The files in this directory are:
1976.egg
Anon
Blake
Shakespeare
Yeats
Which file are you interested in? : Anon
Here are some details about Anon:
Anon: directory
Anon is a directory
Here is a long listing of the Anon directory:
drwxr-xr-x 2 milhom90 cis90 4096 Apr  8 10:01 Anon
Hit the Enter key to return to menu
```

Additional file attributes to test for:

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

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

Class Exercise

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

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

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

Scripting Tips

if then else statement

myscript

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

```
                Homer's CIS 90 Final Project
        1) Color
        2) My Find Command
        3) More practice
        4) Homer's friend made this one - Thank You
        5) Task 5
        6) 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
Wednesday
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
Wednesday, December 03, 2008
Hit the Enter key to return to menu
```

Scripting Tips

Using the set command

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

[rsimms@opus scripts]$ echo $1
dogs

[rsimms@opus scripts]$ echo $2
cats

[rsimms@opus scripts]$ echo $3
birds

[rsimms@opus scripts]$ echo $4
humans

[rsimms@opus scripts]$ echo $#
4

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

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

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

$# is set to the total number of arguments.

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

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

[rsimms@opus bin]$ echo $3
banner

[rsimms@opus bin]$ echo $7
I

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

[rsimms@opus bin]$ echo $#
20

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

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

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

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

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

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

[rsimms@opus scripts]$ echo $2
rsimms

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

[rsimms@opus scripts]$ echo $4
Rich

[rsimms@opus scripts]$ echo $5
Simms

[rsimms@opus scripts]$ firstname=$4

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

Another way to get a user's first name

Class Exercise

Make a new script in your bin directory

```
cd bin
    vi example777
```

In vi add these lines to your script then save:

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

Prepare and run your script

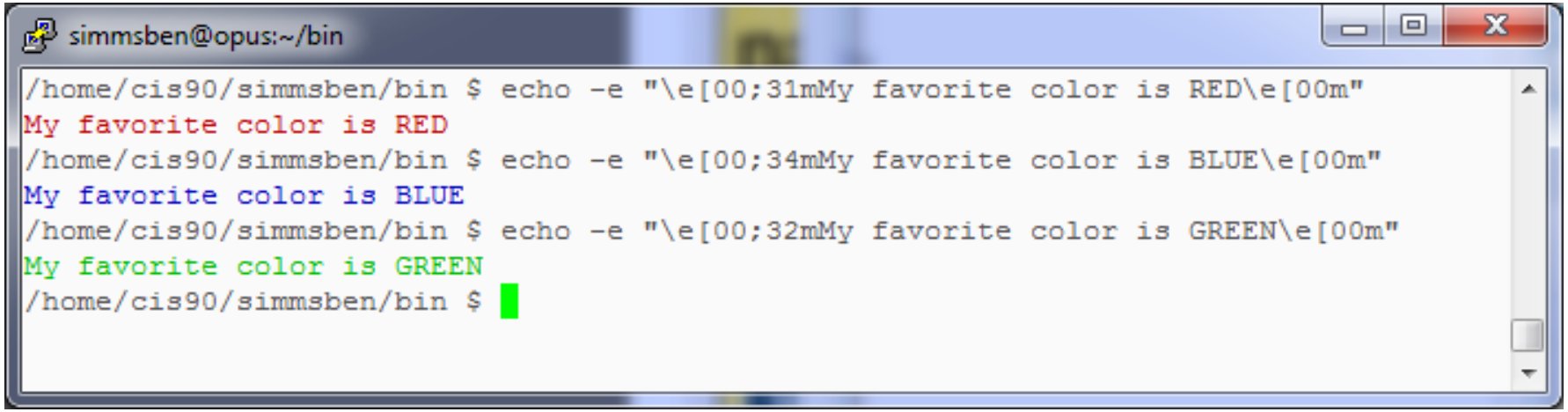
```
    chmod +x example777
    example777
```

Scripting Tips

color

Using Color

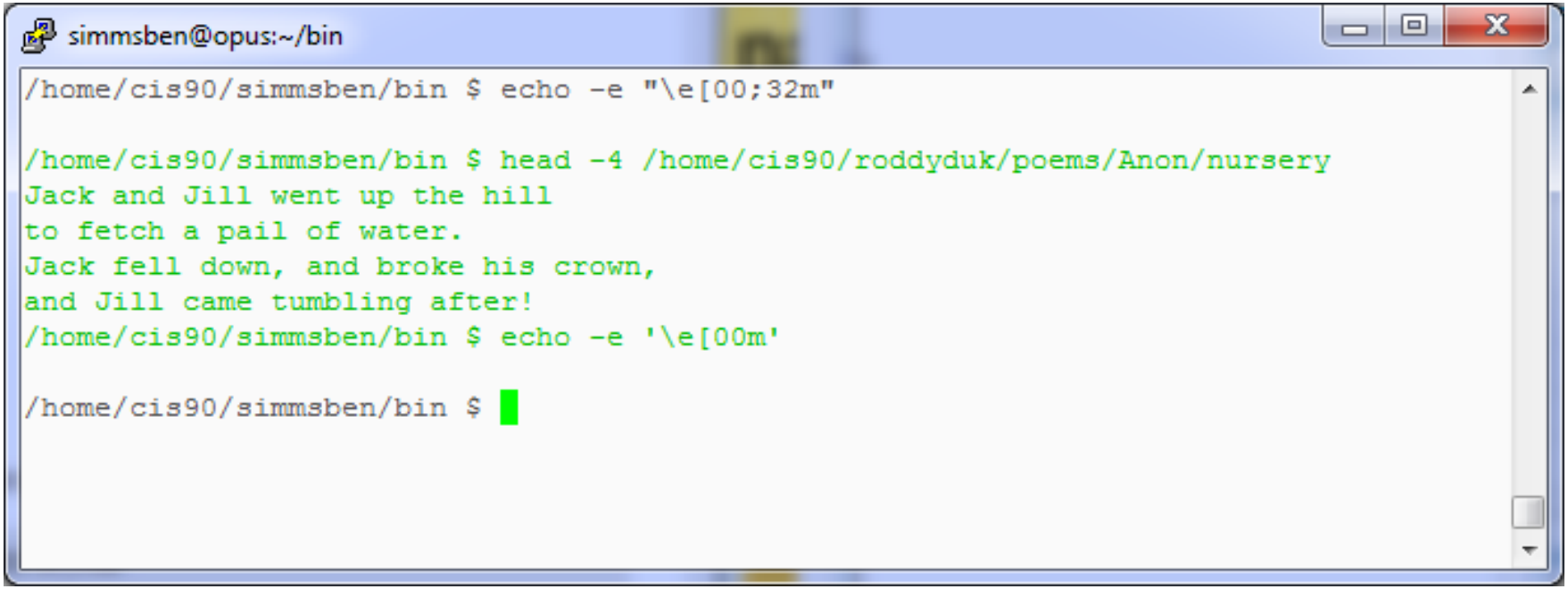
Black 0;30	Green 0;32	Red 0;31	Brown 0;33
Dark Gray 1;30	Light Green 1;32	Light Red 1;31	Yellow 1;33
Blue 0;34	Cyan 0;36	Purple 0;35	Light Gray 0;37
Light Blue 1;34	Light Cyan 1;36	Light Purple 1;35	White 1;37



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

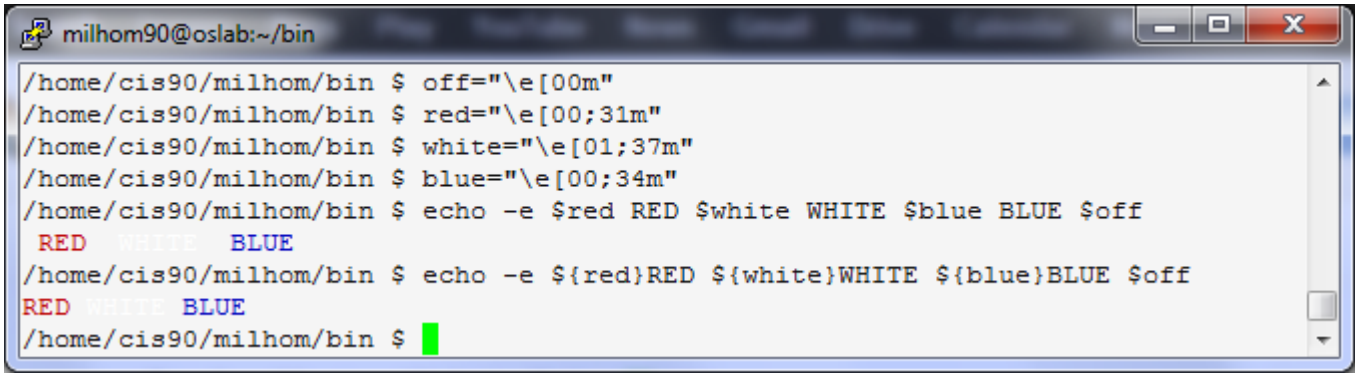
Source: http://hacktux.com/bash/colors

Using Color



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

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




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

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

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

Class Exercise

Make a new script in your bin directory

```
cd bin
vi example4271
```

In vi add these lines to your script then save:

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

Prepare and run your script

```
chmod +x example4271
example4271
```

Scripting Tips

home directories and user names

Going from CIS 90 home directory name ➔ username

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

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

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

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

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

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

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

Going from CIS 90 home directory name ➔ username

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

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

Going from CIS 90 username ➔ home directory name

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

This variable holds your username

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

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

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

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

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

And this is how you could use it

Scripting Tips

Simple for loop

for loop example

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

for loop example

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

Class Exercise

Make a new script in your bin directory

```
cd bin
vi example808
```

In vi add these lines to your script then save:

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

Prepare and run your script

```
chmod +x example808
example808
```

Wrap up

Commands:

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

Next Class

Project is due next week!

Project Workshop

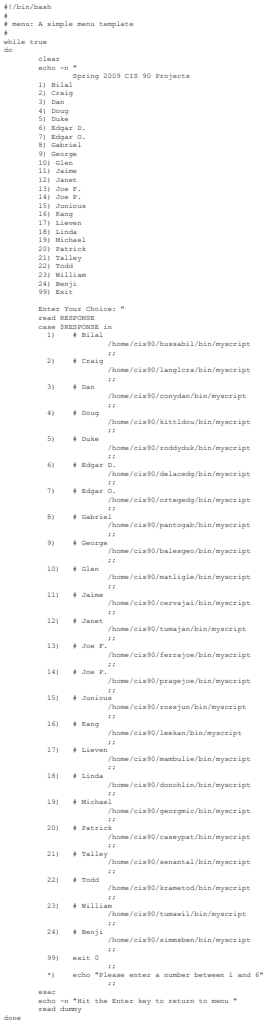
- Make sure you get one "starter" task scripted and working before leaving class today.
- Grade your starter script using the Final Project rubric

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

Backup

Final Project

What is allscripts and myscript?



```
#
# menu: A simple menu template
#
while true
do
clear
echo -n "
CIS 90 Final Project
1) Task 1
2) Task 2
3) Task 3
4) Task 4
5) Task 5
6) Exit
Enter Your Choice: "
read RESPONSE
case $RESPONSE in
1) # Commands for Task 1
;;
2) # Commands for Task 2
;;
3) # Commands for Task 3
;;
4) # Commands for Task 4
;;
5) # Commands for Task 5
;;
6) exit 0
;;
*) echo "Please enter a number between 1 and 6"
;;
esac
echo -n "Hit the Enter key to return to menu "
read dummy
done
```

allscripts

```
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
    clear
    echo -n "
        Spring 2009 cis 90 Projects
1) Bilal
2) Craig
3) Dan
4) Doug
5) Duke
6) Edgar D.
7) Edgar D.
8) Gabriel
9) George
10) Glen
40) Songul
12) Janet
13) Joe F.
14) Joe P.
15) Junior
16) Kang
17) Lieven
18) Linda
19) Michael
20) Patrick
21) Talley
22) Todd
23) William
24) Wenli
99) Exit
Enter Your Choice: "
    read RESPONSE
    case $RESPONSE in
        1)    # Bilal
              /home/cis90/husaabil/bin/myscript
              ''
        2)    # Craig
              /home/cis90/langlcra/bin/myscript
              ''
        3)    # Dan
              /home/cis90/cosydan/bin/myscript
              ''
        4)    # Doug
              /home/cis90/kittidou/bin/myscript
              ''
        5)    # Duke
              /home/cis90/roddyduk/bin/myscript
              ''
        6)    # Edgar D.
              /home/cis90/delacedj/bin/myscript
              ''
        7)    # Edgar D.
              /home/cis90/ortegedj/bin/myscript
              ''
        8)    # Gabriel
              /home/cis90/pantogab/bin/myscript
              ''
        9)    # George
              /home/cis90/balesgeo/bin/myscript
              ''
        10)   # Glen
              /home/cis90/matliglm/bin/myscript
              ''
        40)     # Songul
              /home/cis90/messison/bin/myscript
              ''
        12)   # Janet
              /home/cis90/tumajan/bin/myscript
              ''
        13)   # Joe F.
              /home/cis90/ferrajoe/bin/myscript
              ''
        14)   # Joe P.
              /home/cis90/pragejoe/bin/myscript
              ''
        15)   # Junior
              /home/cis90/rosajun/bin/myscript
              ''
        16)   # Kang
              /home/cis90/leekan/bin/myscript
              ''
        17)   # Lieven
              /home/cis90/maebulie/bin/myscript
              ''
        18)   # Linda
              /home/cis90/donohlin/bin/myscript
              ''
        19)   # Michael
              /home/cis90/georgmic/bin/myscript
              ''
        20)   # Patrick
              /home/cis90/caseypat/bin/myscript
              ''
        21)   # Talley
              /home/cis90/sesantal/bin/myscript
              ''
        22)   # Todd
              /home/cis90/krametod/bin/myscript
              ''
        23)   # William
              /home/cis90/tumawil/bin/myscript
              ''
        24)   # Wenli
              /home/cis90/simswhen/bin/myscript
              ''
        99)     exit 0
              ''
        *)    echo "Please enter a number between 1 and 4"
              ''
    esac
    echo -e "Hit the Enter key to return to menu "
    read dummy
done
```

The while statement in allscripts will loop through the code forever

A case statement is used to run the appropriate myscript file in the student's bin directory. This is specified using an absolute filename.

`/home/cis90/messison/bin/myscript`

*For case 99 the **exit** command is called which causes the script to terminate. The return code of 0 means success.*

myscript

```
#
# menu: A simple menu template
#
while true
do
      clear
      echo -n "
      CIS 90 Final Project
      1) Task 1
      2) Task 2
      3) Task 3
      4) Task 4
      5) Task 5
      6) Exit
      Enter Your Choice: "
      read RESPONSE
      case $RESPONSE in
            1) # Commands for Task 1
            ;;
            2) # Commands for Task 2
            ;;
            3) # Commands for Task 3
            ;;
            4) # Commands for Task 4
            ;;
            5) # Commands for Task 5
            ;;
            6) exit 0
            ;;
            *) echo "Please enter a number between 1 and 6"
            ;;
      esac
      echo -n "Hit the Enter key to return to menu "
      read dummy
done
```

*The outer while statement will loop forever. The only way out is the **exit** command in case 6)*

myscript

```
#
# menu: A simple menu template
#
while true
do
      clear
      echo -n "
      CIS 90 Final Project
      1) Task 1
      2) Task 2
      3) Task 3
      4) Task 4
      5) Task 5
      6) Exit
      Enter Your Choice: "
      read RESPONSE
      case $RESPONSE in
            1) # Commands for Task 1
            ;;
            2) # Commands for Task 2
            ;;
            3) # Commands for Task 3
            ;;
            4) # Commands for Task 4
            ;;
            5) # Commands for Task 5
            ;;
            6) exit 0
            ;;
            *) echo "Please enter a number between 1 and 6"
            ;;
      esac
      echo -n "Hit the Enter key to return to menu "
      read dummy
done
```

This is a single echo command that prints a menu for the user

myscript

```
#
# menu: A simple menu template
#
while true
do
      clear
      echo -n "
      CIS 90 Final Project
      1) Task 1
      2) Task 2
      3) Task 3
      4) Task 4
      5) Task 5
      6) Exit
      Enter Your Choice: "
      read RESPONSE
      case $RESPONSE in
            1) # Commands for Task 1
            ;;
            2) # Commands for Task 2
            ;;
            3) # Commands for Task 3
            ;;
            4) # Commands for Task 4
            ;;
            5) # Commands for Task 5
            ;;
            6) exit 0
            ;;
            *) echo "Please enter a number between 1 and 6"
            ;;
      esac
      echo -n "Hit the Enter key to return to menu "
      read dummy
done
```

This is a case statement. One case for each task. Note the end of the case statement is case spelled backwards!

myscript

```
#
# menu: A simple menu template
#
while true
do
      clear
      echo -n "
      CIS 90 Final Project
      1) Task 1
      2) Task 2
      3) Task 3
      4) Task 4
      5) Task 5
      6) Exit
      Enter Your Choice: "
      read RESPONSE
      case $RESPONSE in
            1) # Commands for Task 1
            ;;
            2) # Commands for Task 2
            ;;
            3) # Commands for Task 3
            ;;
            4) # Commands for Task 4
            ;;
            5) # Commands for Task 5
            ;;
            6) exit 0
            ;;
            *) echo "Please enter a number between 1 and 6"
            ;;
      esac
      echo -n "Hit the Enter key to return to menu "
      read dummy
done
```

*The **read** command gets input from the user and stores it in a variable.*

*The variable to use is specified as the argument on the **read** command.*

scp

Copying files between systems

scp

Copy commands **copy file(s)** to a **Destination**

- cp
  - copies files on the same system

```
cp /etc/hosts .
cp riddel1 riddle2 riddles/
cp tally tally.v1
```

*For the **cp** command each argument is a pathname*

- scp
  - copies files between systems:

```
scp milhom90@oslab.cabrillo.edu:/etc/hosts .
scp riddle1 riddle2 cis90@P1-Hugo:riddles/
scp -P 425 rsimms@frodo.simms-teach.com:tally tally.v1
```

*For the **scp** command, arguments for remote files must include username, hostname, pathname and optionally a port.*

The @ and : separators are always required with scp

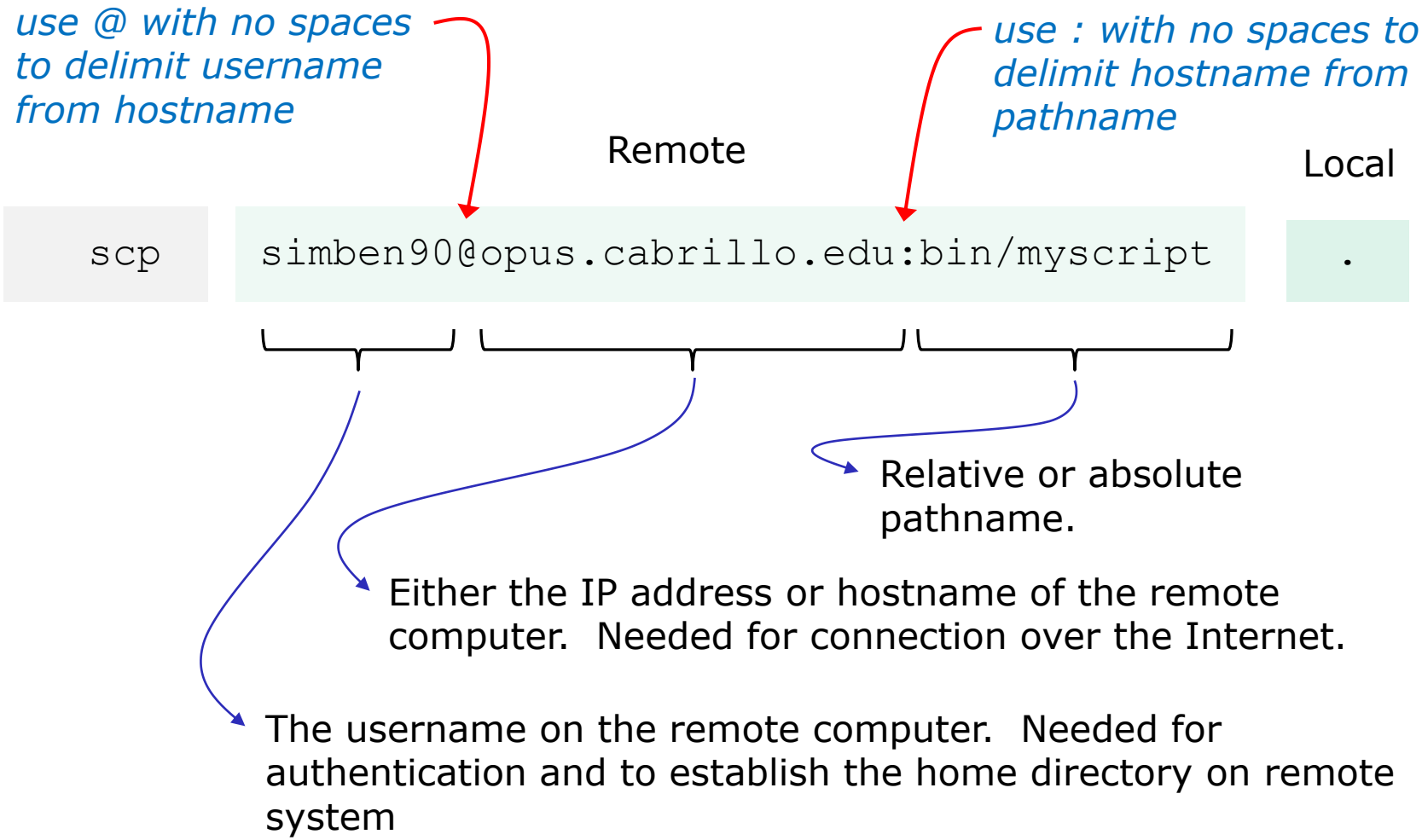
scp

	Remote	Local
`scp`	`simben90@opus.cabrillo.edu:bin/myscript`	`.`

Copy the file myscript from simben90's home bin/ directory on the remote system Opus to "here"

scp example

Copying project file on Opus to local Linux system



Copying a file from Opus to Sun-Hwa
(initiated from Sun-Hwa)

On Opus

```
/home/cis90/simben $ head -n1 ../depot/scrooge
Stave 2: The First of the Three Spirits
/home/cis90/simben $
```

On Sun-Hwa

```
[CISLAB\simben90@sun-hwa ~]$ head -n1 scrooge
head: cannot open `scrooge' for reading: No such file or directory

[CISLAB\simben90@sun-hwa ~]$ scp simben90@opus.cabrillo.edu:../depot/scrooge .
simben90@opus.cabrillo.edu's password:
scrooge                                       100%   33KB  33.1KB/s   00:00
[CISLAB\simben90@sun-hwa ~]$

[CISLAB\simben90@sun-hwa ~]$ head -n1 scrooge
Stave 2: The First of the Three Spirits
[CISLAB\simben90@sun-hwa ~]$
```

Copying multiple files from Opus to Sun-Hwa
(initiated from Sun-Hwa)

On Opus

```
/home/cis90/simben $ ls bin
app     datecal         hi  myscript     myscript.v2  simple.c  tryme
banner  enlightenment   I   myscript.v1  simple       treed     zoom
/home/cis90/simben $
```

On Sun-Hwa

```
[CISLAB\simben90@sun-hwa ~]$ ls bin
ls: cannot access bin: No such file or directory
[CISLAB\simben90@sun-hwa ~]$ mkdir bin
[CISLAB\simben90@sun-hwa ~]$ scp simben90@opus:bin/my* bin/
simben90@opus's password:
myscript                                      100%   10KB  10.2KB/s   00:00
myscript.v1                                   100%   10KB  10.2KB/s   00:00
myscript.v2                                   100%   10KB  10.2KB/s   00:00
[CISLAB\simben90@sun-hwa ~]$
```

Copying a file from Sun-Hwa to Opus
(initiated from Sun-Hwa)

On Opus

```
/home/cis90/simben $ ls file25
ls: cannot access file25: No such file or directory
```

On Sun-Hwa

```
[CISLAB\simben90@sun-hwa ~]$ echo "I love Linux" > file25
[CISLAB\simben90@sun-hwa ~]$ scp file25 simben90@opus:
The authenticity of host 'opus (172.30.5.20)' can't be established.
RSA key fingerprint is 7d:32:80:b9:52:32:c8:dc:3b:16:0e:ba:8c:fd:79:ef.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'opus,172.30.5.20' (RSA) to the list of known hosts.
simben90@opus's password:
file25                                        100%   13     0.0KB/s   00:00
[CISLAB\simben90@sun-hwa ~]$
```

```
/home/cis90/simben $ cat file25
I love Linux
```

Copying a file from Sun-Hwa to Opus
(initiated from Opus)

On Sun-Hwa

```
[CISLAB\simben90@sun-hwa ~]$ echo "I love dogs" > file15
[CISLAB\simben90@sun-hwa ~]$
```

On Opus

```
/home/cis90/simben $ cat file15
cat: file15: No such file or directory
/home/cis90/simben $

/home/cis90/simben $ scp cislab\\simben90@sun-hwa:file15 .
cislab\simben90@sun-hwa's password:
file15                                        100%   12     0.0KB/s   00:00
/home/cis90/simben $

/home/cis90/simben $ cat file15
I love dogs
/home/cis90/simben $
```

Copying a file from Sun-Hwa to Opus and renaming it
(initiated from Sun-Hwa)

On Opus

```
/home/cis90/simben $ cat iloveunix
cat: iloveunix: No such file or directory
/home/cis90/simben $
```

On Sun-Hwa

```
[CISLAB\simben90@sun-hwa ~]$ echo "I love UNIX" > file35
[CISLAB\simben90@sun-hwa ~]$ scp file35 simben90@opus:iloveunix
simben90@opus's password:
file35                                        100%   12     0.0KB/s   00:00
[CISLAB\simben90@sun-hwa ~]$
```

```
/home/cis90/simben $ cat iloveunix
I love UNIX
/home/cis90/simben $
```

Class Activity

- On Opus, locate the *ptest.template* file in the CIS 90 *depot* directory
- Log into Sun-Hwa with: **ssh cislab***username***@sun-hwa**
- On Sun-Hwa, copy the *ptest03.template* file in the Opus CIS 90 *depot* directory to your Sun-Hwa home directory renaming it *ptest03* at the same time

if [*it worked*]; **then**
  write the command you used on Sun-Hwa into the chat window
else
  write the error message you got into the chat window
fi

tar

Archiving files

tar command

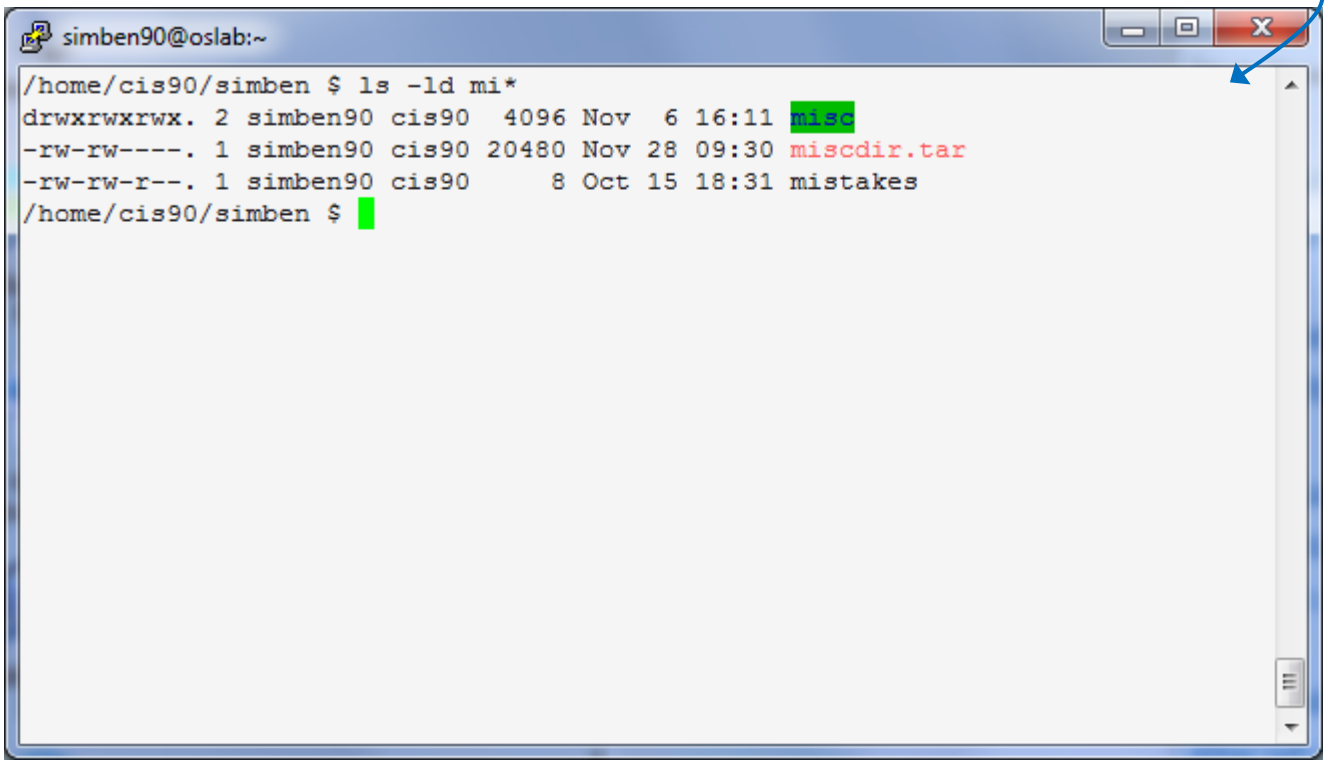
Create a tarball out of our local misc directory

```
/home/cis90/simben $ ls misc
file.dos  fruit  manpage  mystery  salad  tiurf
what_am_i
/home/cis90/simben $
```

```
/home/cis90/simben $ tar cvf miscdir.tar misc/
misc/
misc/fruit
misc/file.dos
misc/salad
misc/mystery
misc/what_am_i
misc/manpage
misc/tiurf
/home/cis90/simben $
```

tar command

Tarballs show as red in listings



tar command

View contents of a tarball

```
/home/cis90/simben $ tar tvf miscdir.tar
drwxrwxrwx simben90/cis90    0 2012-11-06 16:11 misc/
-rw-r--r-- simben90/cis90   78 2004-10-26 16:36 misc/fruit
-rw-r--r-- simben90/cis90  148 2001-07-20 22:54 misc/file.dos
-rw-r--r-- simben90/cis90   78 2004-04-17 12:13 misc/salad
lrwxrwxrwx simben90/cis90    0 2012-08-01 16:55 misc/mystery -> ../bin/enlightenment
-rw-r--r-- simben90/cis90  352 2001-07-20 15:04 misc/what_am_i
-rw-r--r-- simben90/cis90 10576 2001-07-20 20:58 misc/manpage
-rw-rw-r-- simben90/cis90    78 2012-10-15 09:25 misc/tiurf
/home/cis90/simben $
```

tar command

On another Linux system (Sun-Hwa in VLab)

```
[CISLAB\simben90@sun-hwa ~]$ ls misc
ls: cannot access misc: No such file or directory
[CISLAB\simben90@sun-hwa ~]$
```

tar command

On another Linux system (Sun-Hwa in VLab)

```
[CISLAB\simben90@sun-hwa ~]$ ls misc
ls: cannot access misc: No such file or directory


[CISLAB\simben90@sun-hwa ~]$ scp simben90@opus:miscdir.tar .
simben90@opus's password:
miscdir.tar                                   100%   20KB  20.0KB/s   00:00
[CISLAB\simben90@sun-hwa ~]$


[CISLAB\simben90@sun-hwa ~]$ tar xvf miscdir.tar
misc/
misc/fruit
misc/file.dos
misc/salad
misc/mystery
misc/what_am_i
misc/manpage
misc/tiurf
[CISLAB\simben90@sun-hwa ~]$
```

Copy tarball from Opus

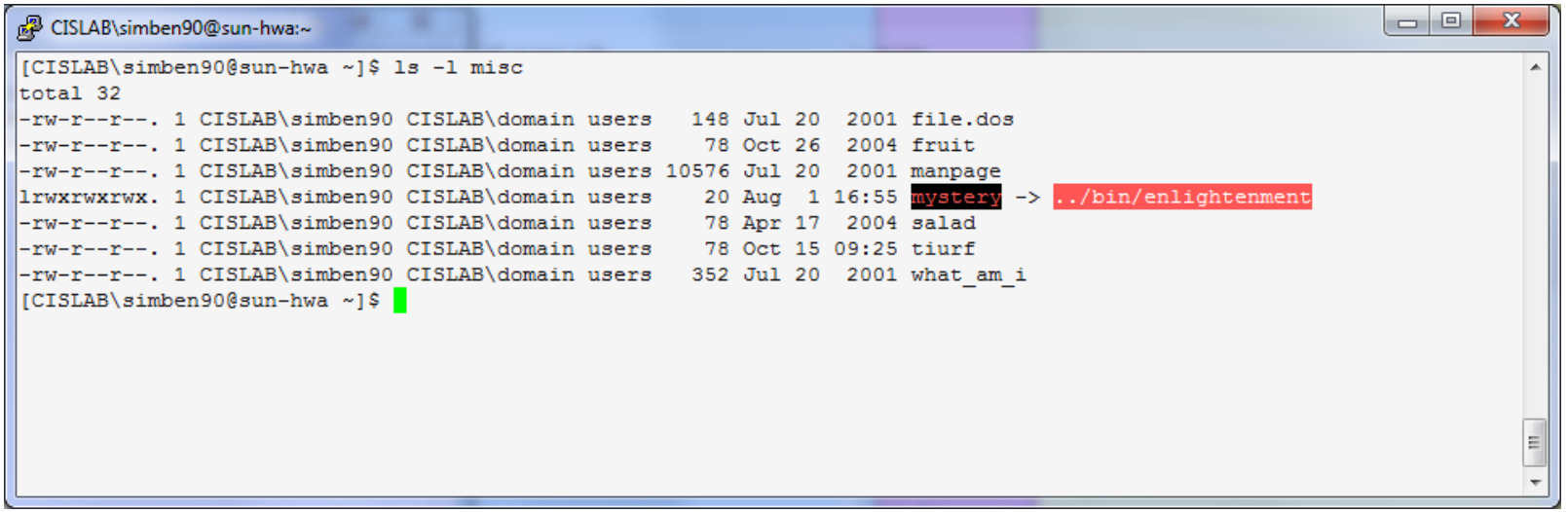
Extract tarball on Sun-Hwa

Note, misc/ directory is created and populated

Be careful, this will overwrite any files with the same name!

tar command

After extraction



Note: the symbolic link is broken because there is no enlightenment file in local bin directory on Sun-Hwa

Class Activity

Only do this if you have not started Lab X2 already:

- On Opus, locate the *dogs.tar* tarball in the CIS 90 *depot* directory
- Copy it to your home directory
- Extract the contents to your home directory
- List your new *dogs/* directory