



Lesson Module Checklist

- Slides
- WB
- Flash cards
- Page numbers
- 1st minute quiz - NA
- Web Calendar summary
- Web book pages
- Commands
- Dog script examples ready
- Materials uploaded
- Backup slides, CCC info, handouts on flash drive
- 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/>)



Instructor: **Rich Simms**

Dial-in: **888-450-4821**

Passcode: **761867**



Buzz



Carlos



Emily



Jon M.



Jon W.



Jordan



Joseph



JJ



Kiernan



Maria



Mathew



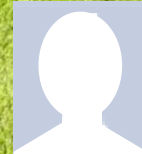
Mike C.



Michael F.



Mike M.



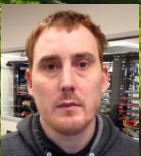
Nick L.



Rebecca



Ricardo



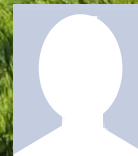
Robert



Steve



Tess



Tim



Troy

Quiz

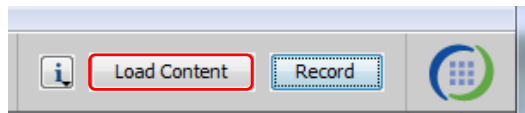
**No Quiz
Today !**



Whiteboard and Audio Setup

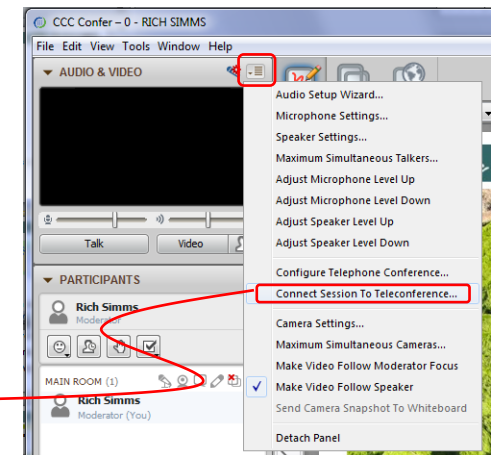
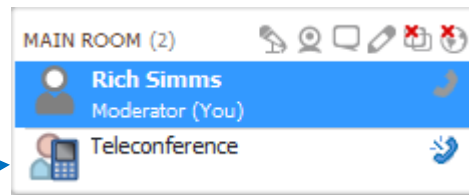


[] Preload White Board with cis*lesson??*-WB



[] Connect session to Teleconference

Session now connected to teleconference



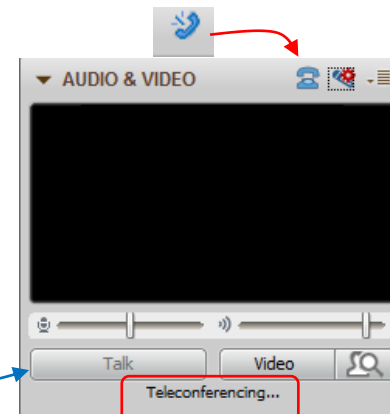
[] Is recording on?



Red dot means recording

[] Use teleconferencing, not mic

Should be greyed out



Keep wireless mic transmitter away from cell phone and podium if excess static occurs

Layout and share apps



foxit for slides

chrome

putty

vSphere Client

simben90@oslab:~
login as: simben90
simben90@oslab.cabrillo.edu's password:
Access denied
simben90@oslab.cabrillo.edu's password:
Last login: Mon Oct 8 18:58:43 2012 from
d.com

Current directory
source
destination

What command copies th

Part 1 - Flashcards questions (1 point each)

[Q1] What command shows the other users logged in to the computer?
[A1]

[Q2] What environment variable is used by the shell to determine which directories to search when locating a command?
[A2]

CIS 192
Getting Started
Virtual Machines
Tasks & Events
Alarms
Perf

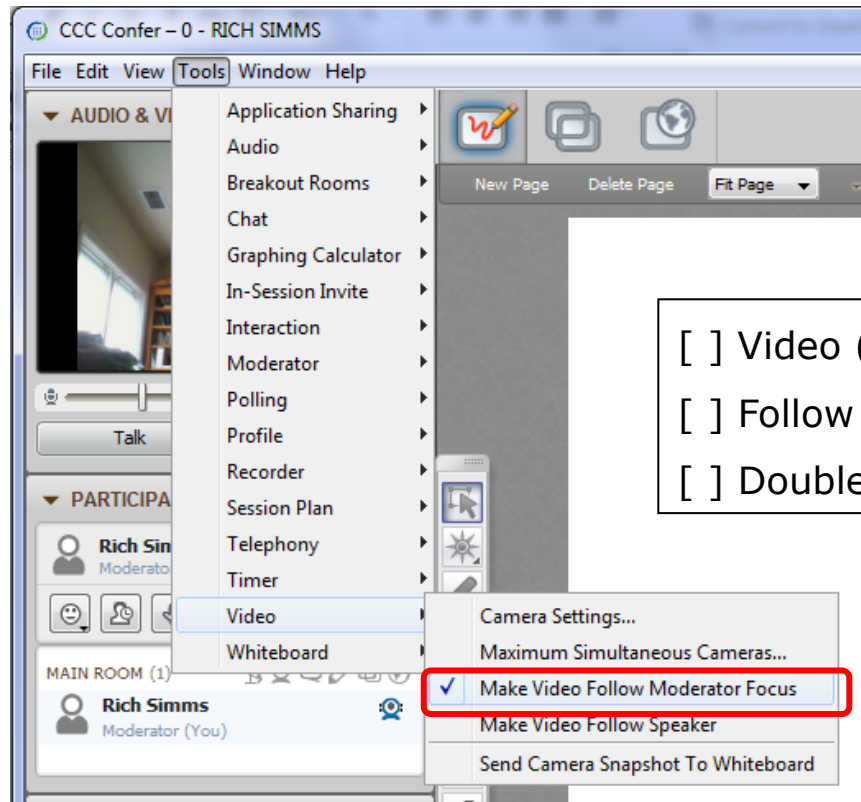
Recent Tasks

Name	Target	Status	Details	Initiated by

License Period: 331 days remaining | CISLAB/simms

6:52 AM
10/10/2012

Webcam for video



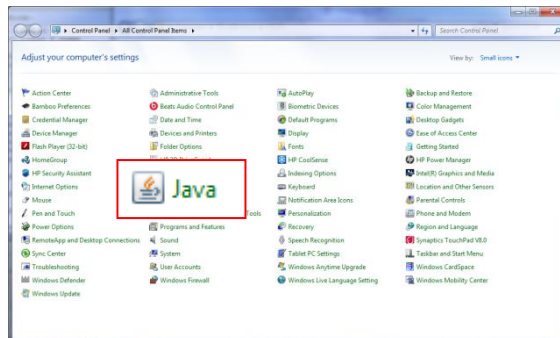
- ☐ Video (webcam) optional
- ☐ Follow moderator
- ☐ Double-click on postages stamps



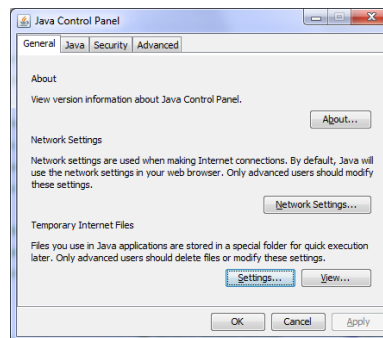
CCC Confer Troubleshooting



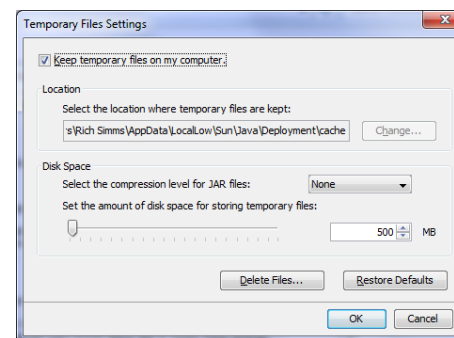
Control Panel (small icons)



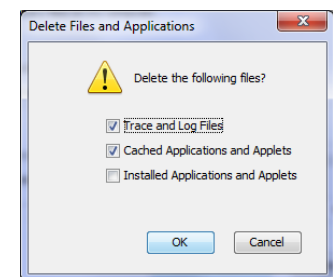
General Tab > Settings...



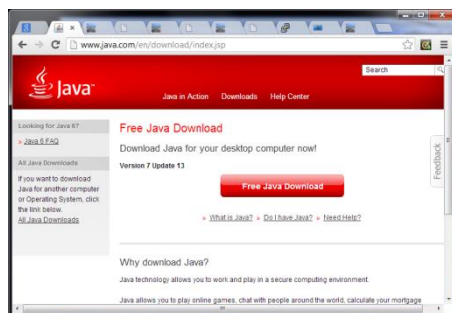
500MB cache size



Delete these



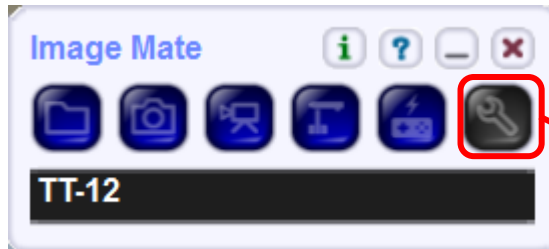
Google Java download



Universal Fix for CCC Confer:

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

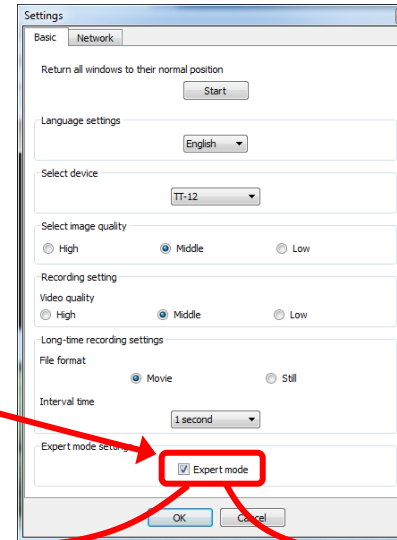
Using Elmo with CCC Confer



Elmo rotated down to view side table



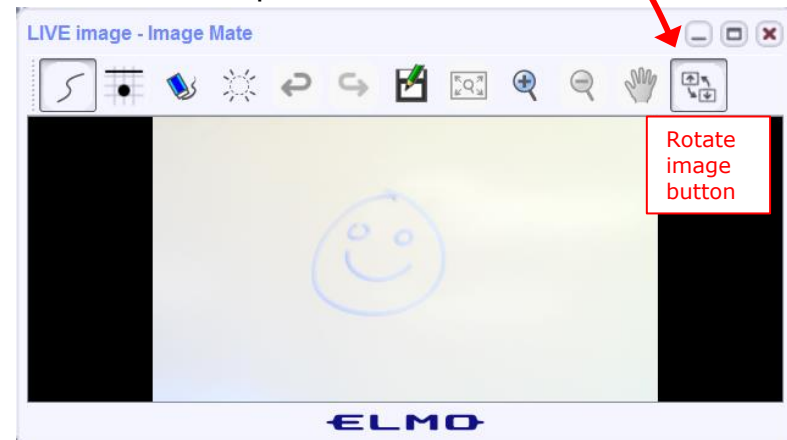
Run and share the Image Mate program just as you would any other app with CCC Confer



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

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

Elmo rotated up to view white board



More Shell Scripting

Objectives

- Use conditionals in scripts
- Transfer files between computers
- Archive directories using tar

Agenda

- 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>*

cp -r *<source directory> <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>

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

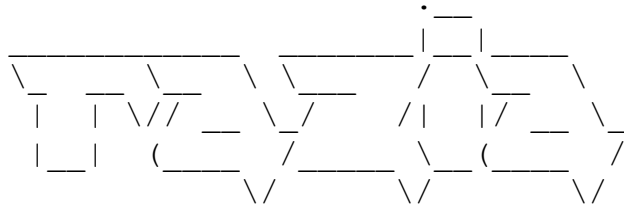


When copying files between systems it is necessary to use scp and specify non-standard **ports**, **login usernames**, and **hostnames** for the remote system

Copy a file from Opus to Razia

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

```
/home/cis90/simben $ ssh simben90@razia Login to Razia
Password:
Last login: Tue May 6 09:02:26 2014 from opus.cis.cabrillo.edu
```



Have a lot of fun...

```
razia:~> scp -P 2220 simben90@oslab.cishawks.net:bin/tryme . Copy file from Opus to Razia
The authenticity of host '[oslab.cishawks.net]:2220 ([2607:f380:80f:f425::230]:2220)'
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.cishawks.net]:2220,[2607:f380:80f:f425::230]:2220'
(RSA) to the list of known hosts.
simben90@oslab.cishawks.net's password:
tryme                               100% 174      0.2KB/s   00:00
razia:~> tryme
bash: tryme: command not found
razia:~> ./tryme
My name is "tryme"
I am pleased to make your acquaintance, Benji Simms
/tmp
```

Copy a file from Opus to Razia

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

*Use port 2220 to
connect to the
remote system*

*The full
hostname of the
remote system*

*Relative path to
source file on
remote system*

```
razia:~> scp -P 2220 simben90@oslab.cishawks.net:bin/tryme .
```

*The username
on the remote
system*

*Required : to
separate remote
hostname from local
pathname*

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

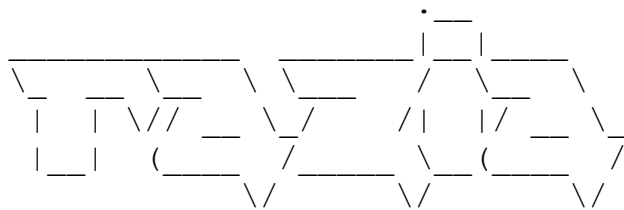
Copying a directory recursively from Opus to Razia

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

/home/cis90/simben \$ **ssh razia** *Login to Razia*

Password:

Last login: Tue May 6 09:27:44 2014 from opus.cis.cabrillo.edu



Have a lot of fun...

razia:~> **scp -r -P 2220 simben90@oslab.cishawks.net:bin bin** *Copy branch from Opus to Razia*

simben90@oslab.cishawks.net's password:

enlightenment	100%	3442	3.4KB/s	00:00
simple	100%	5010	4.9KB/s	00:00
treed	100%	190	0.2KB/s	00:00
zoom	100%	74	0.1KB/s	00:00
simple.c	100%	312	0.3KB/s	00:00
app	100%	220	0.2KB/s	00:00
home	100%	110	0.1KB/s	00:00
hi	100%	107	0.1KB/s	00:00
myscript	100%	10KB	10.3KB/s	00:00
I	100%	375	0.4KB/s	00:00
mylog	100%	1078	1.1KB/s	00:00
tryme	100%	174	0.2KB/s	00:00
datecal	100%	519	0.5KB/s	00:00
banner	100%	6160	6.0KB/s	00:00

razia:~>

Copying a directory recursively from Opus to Razia

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

Use port 2220 to connect to the remote system

The full hostname of the remote system

Path to source directory on remote system

```
razia:~> scp -r -P 2220 simben90@oslab.cishawks.net:bin bin
```

recursive copy

The username on the remote system

Required : to separate remote hostname from local pathname

the target directory

Class Activity

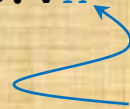
On Opus:

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 Razia:

Copy all your myscript project files to Razia:

```
ssh razia
```

```
hostname
```

```
scp opus:bin/myscript* bin/
```

Note: This will overwrite any files with the same name on Razia

Notes:

If the username is the same on the remote system you can drop the "username@"

If your nameserver can resolve the short hostname it can be used instead of the full hostname

If port 22 is used then the -P port option can be left off

tar

tar command

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

Basic tar command syntax

verbose
specify the archive file

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

creates an archive

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

views an archive's table of contents

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

extracts archive files to the current directory

Basic tar command syntax

The tar command was written before POSIX command line conventions

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

```
tar cvf <tarfile> <files-or-directory-to-archive>
```

are equivalent

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

```
tar tvf <tarfile>
```

are equivalent

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

```
tar xvf <tarfile>
```

are equivalent

Example

Backup and restore a directory

Archive your Blake directory of poems

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

*create
verbose
file*

*name of
archive file
(tarball)*

*pathname
to directory
to archive*

Example

Backup and restore a directory

*table of contents
verbose
file*

*name of
archive file
(tarball)*



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

View new archive's table of contents

Example

Backup and restore a directory

Clobber (remove) your directory of Blake poems

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

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

Example

Backup and restore a directory

No problem, we have a backup!

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

*extract
verbose
file*

*name of
archive file
(tarball)*

Restore your directory of Blake poems

tar
+
scp

Example

Copy archived directory to another system

Backup your bin directory

```

/home/cis90/simben $ ls bin
app      datecal      hi      I      myscript.v1  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*

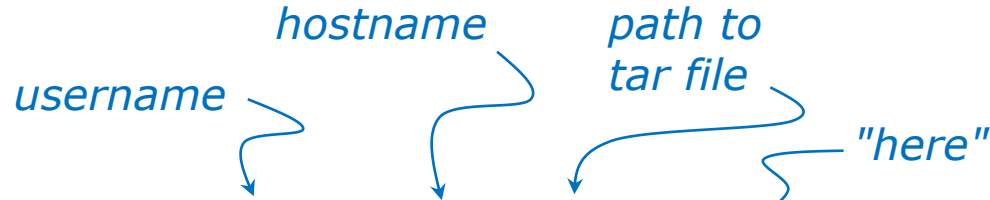


```
/home/cis90/simben $ ssh cis90@p06-arwen  
cis90@p06-arwen's password:  
Welcome to Linux Mint 15 Olivia (GNU/Linux 3.8.0-26-generic x86_64)  
  
Welcome to Linux Mint  
* Documentation: http://www.linuxmint.com  
Last login: Sun Sep  1 19:48:05 2013  
cis90@p06-arwen:~ >
```

Login to your own 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
```

*extract
verbose
file*

*name of
archive file
(tarball)*

```
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:~ >
```

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]
```

```
Fetch 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) 7-9:50AM.
 - Extra credit labs are due by 11:59PM .

Make backup copies of your script

modify, debug, modify, debug, ... rest

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

modify, debug, modify, debug, ... rest

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

modify, debug, modify, debug, ... rest

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

Fall 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 72.

Transfer Credit: CSU.

Section	Days	Times	Units	Instructor	Room
84743	T	01:00PM-04:05PM	3.00	R.Simms	OL
Section 84743 is an ONLINE course. Meets weekly throughout the semester online during the scheduled times by remote technology. For details, see instructor's web page at go.cabrillo.edu/online .					
86576	T	01:00PM-04:05PM	3.00	R.Simms	828

CIS 191AB UNIX/Linux Installation, Configuration and Administration

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

Section	Days	Times	Units	Instructor	Room
84737	TH	01:00PM-05:05PM	4.00	M.Matera	OL
&	Arr.	Arr.		M.Matera	OL
Section 84737 is an ONLINE course. Meets weekly throughout the semester online at the scheduled times by remote technology with an additional 4 hr 5 min online lab per week. For details, see instructor's web page at go.cabrillo.edu/online .					
86577	TH	01:00PM-05:05PM	4.00	M.Matera	828
&	Arr.	Arr.		M.Matera	OL
Section 86577 is a Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 4 hr 5 min online lab per week. 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).
- Room 828 on campus.
- Timed test (no 11:59PM grace period)

	5/21	Test #3 (the final exam) Time • 7:00AM - 9:50AM in Room 828 Materials • Test (download)		5 posts Lab X1 Lab X2
--	------	---	--	---

- If you are a long distance student, contact the instructor for options.

<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 5/5/2014

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

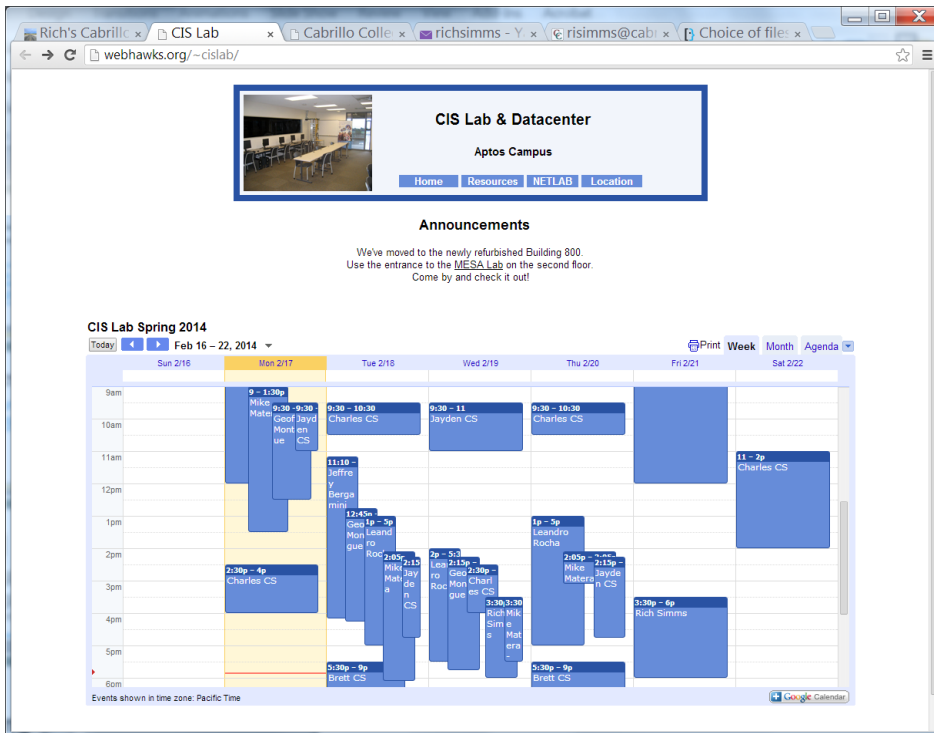
Percentage	Total Points	Letter Grade	Pass/No Pass
90% or higher	504 or higher	A	Pass
80% to 89.9%	448 to 503	B	Pass
70% to 79.9%	392 to 447	C	Pass
60% to 69.9%	336 to 391	D	No pass
0% to 59.9%	0 to 335	F	No pass

alatar: 70% (318 of 450 points)
anborn: 82% (373 of 450 points)
aragorn: 78% (354 of 450 points)
arwen: 107% (483 of 450 points)
celebrian: 97% (440 of 450 points)
dwalin: 94% (425 of 450 points)
eomer: 92% (418 of 450 points)
faramir: 96% (432 of 450 points)
frodo: 79% (357 of 450 points)
gwaihir: 105% (473 of 450 points)
ioareth: 94% (423 of 450 points)
legolas: 90% (407 of 450 points)

marhari: 73% (329 of 450 points)
orome: 75% (340 of 450 points)
pippen: 66% (298 of 450 points)
quickbeam: 92% (414 of 450 points)
rian: 0% (0 of 450 points)
samwise: 84% (379 of 450 points)
strider: 88% (396 of 450 points)
treebeard: 105% (475 of 450 points)
tulkas: 88% (399 of 450 points)
ulmo: 85% (386 of 450 points)

CIS Lab Schedule

<http://webhawks.org/~cislab/>



Not submitting tests or lab work?

Would like some additional help?

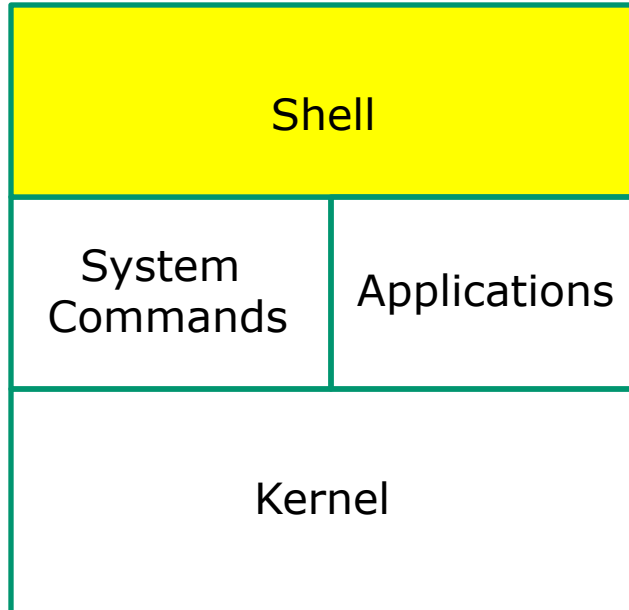
Come to the CIS Lab to work with classmates, lab assistants and instructors on Lab assignments.

Rich is in the lab Wednesdays and Fridays from 3:30 - 6:00 PM

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

```
simben90@oslab:~  
  
*****  
*          Spring 2014 CIS 90 Online Projects          *  
*****  
1) Benji S.  
2) Buzz T.  
3) Carlos P.  
4) Duke R.  
5) Elijah D.  
6) Emily G.  
7) Enrique R.  
8) Homer M.  
9) JJ R.  
10) Jon M.  
11) Jon W.  
12) Jordan V.  
13) Joseph K.  
14) Kiernan B.  
15) Maria G.  
16) Mathew H.  
17) Michael F.  
18) Mike C.  
19) Mike M.  
20) Nick L.  
21) Patrick M.  
22) Rebecca L.  
23) Ricardo C.  
24) Robert L.  
25) Steve P.  
26) Tess F.  
27) Tim W.  
28) Troy R.  
  
99) Exit  
  
Enter Your Choice: █
```

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

```
simben90@oslab:~  
  
Benji, please Enter an option number from the list below:  
  
1) What is today?  
2) The users on oslab.cabrillo.edu  
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: █
```

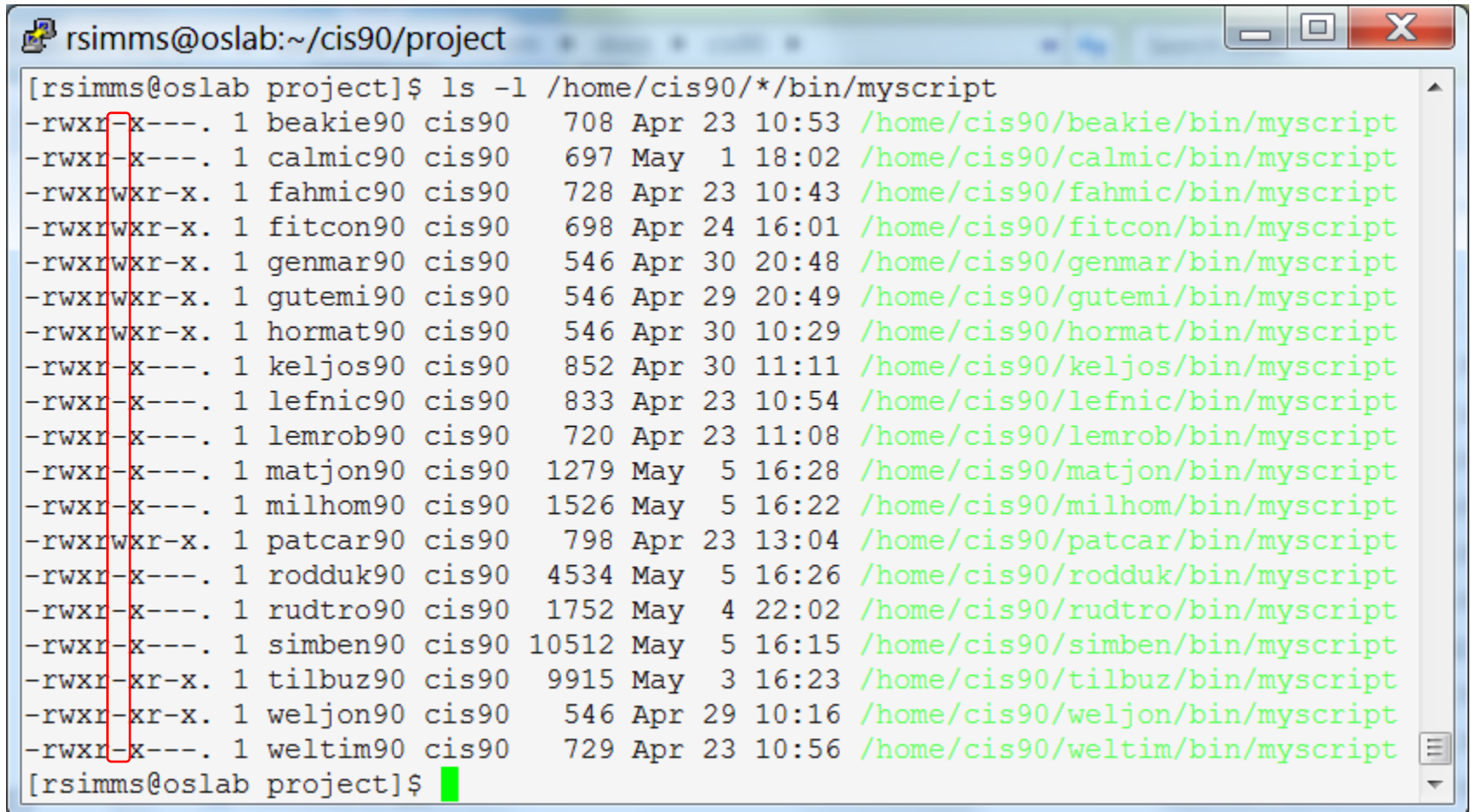
Don't forget to do this!

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
```

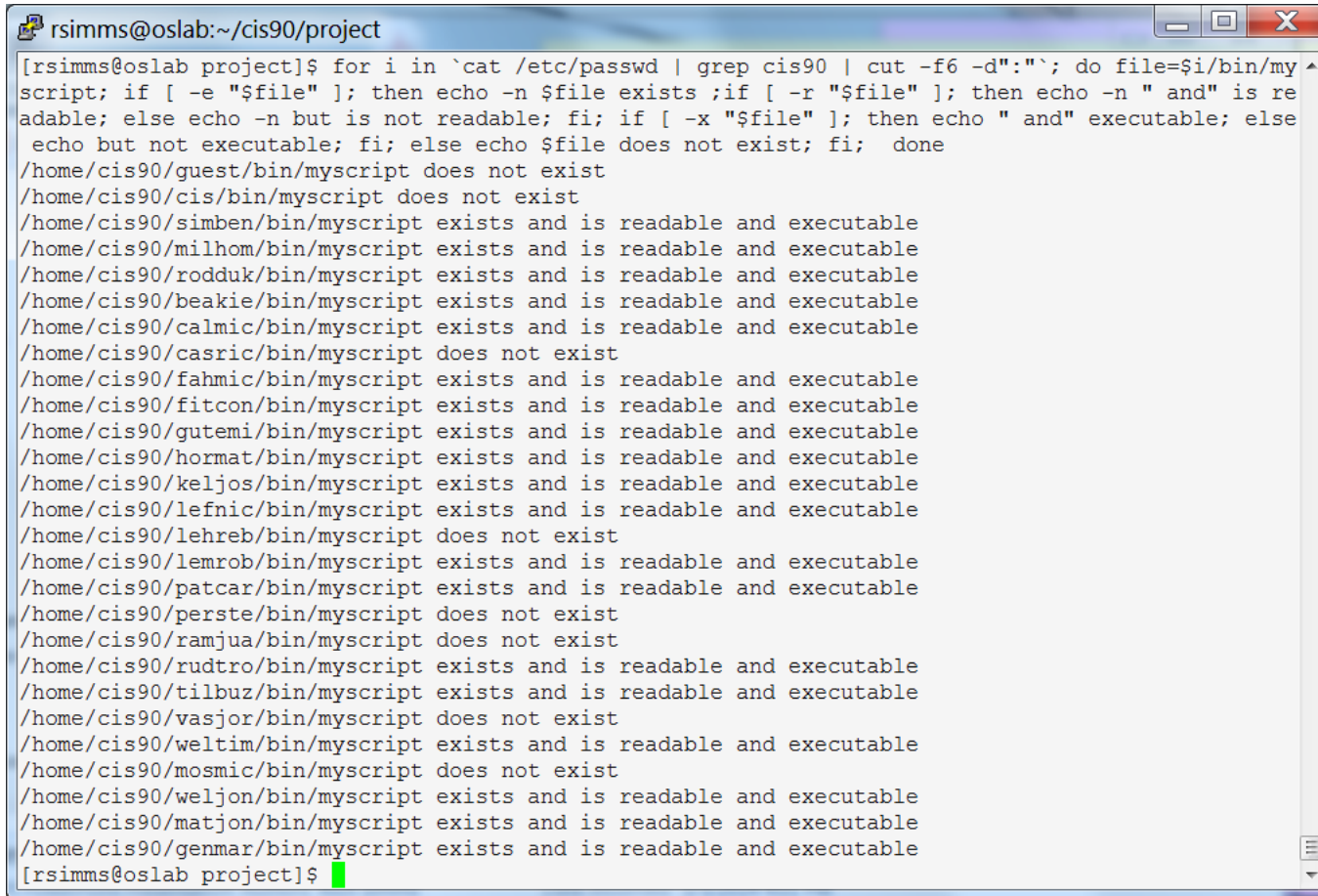


```
rsimms@oslab:~/cis90/project
[rsimms@oslab project]$ ls -l /home/cis90/*/bin/myscript
-rwxr-x---. 1 beakie90 cis90  708 Apr 23 10:53 /home/cis90/beakie/bin/myscript
-rwxr-x---. 1 calmic90 cis90  697 May  1 18:02 /home/cis90/calmic/bin/myscript
-rwxrwxr-x. 1 fahmic90 cis90  728 Apr 23 10:43 /home/cis90/fahmic/bin/myscript
-rwxrwxr-x. 1 fitcon90 cis90  698 Apr 24 16:01 /home/cis90/fitcon/bin/myscript
-rwxrwxr-x. 1 genmar90 cis90  546 Apr 30 20:48 /home/cis90/genmar/bin/myscript
-rwxrwxr-x. 1 gutemi90 cis90  546 Apr 29 20:49 /home/cis90/gutemi/bin/myscript
-rwxrwxr-x. 1 hormat90 cis90  546 Apr 30 10:29 /home/cis90/hormat/bin/myscript
-rwxr-x---. 1 keljos90 cis90  852 Apr 30 11:11 /home/cis90/keljos/bin/myscript
-rwxr-x---. 1 lefnic90 cis90  833 Apr 23 10:54 /home/cis90/lefnic/bin/myscript
-rwxr-x---. 1 lemrob90 cis90  720 Apr 23 11:08 /home/cis90/lemrob/bin/myscript
-rwxr-x---. 1 matjon90 cis90 1279 May  5 16:28 /home/cis90/matjon/bin/myscript
-rwxr-x---. 1 milhom90 cis90 1526 May  5 16:22 /home/cis90/milhom/bin/myscript
-rwxrwxr-x. 1 patcar90 cis90  798 Apr 23 13:04 /home/cis90/patcar/bin/myscript
-rwxr-x---. 1 rodduk90 cis90 4534 May  5 16:26 /home/cis90/rodduk/bin/myscript
-rwxr-x---. 1 rudtro90 cis90 1752 May  4 22:02 /home/cis90/rudtro/bin/myscript
-rwxr-x---. 1 simben90 cis90 10512 May  5 16:15 /home/cis90/simben/bin/myscript
-rwxr-xr-x. 1 tilbuz90 cis90  9915 May  3 16:23 /home/cis90/tilbuz/bin/myscript
-rwxr-xr-x. 1 weljon90 cis90  546 Apr 29 10:16 /home/cis90/weljon/bin/myscript
-rwxr-x---. 1 weltim90 cis90  729 Apr 23 10:56 /home/cis90/weltim/bin/myscript
[rsimms@oslab project]$
```

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
```

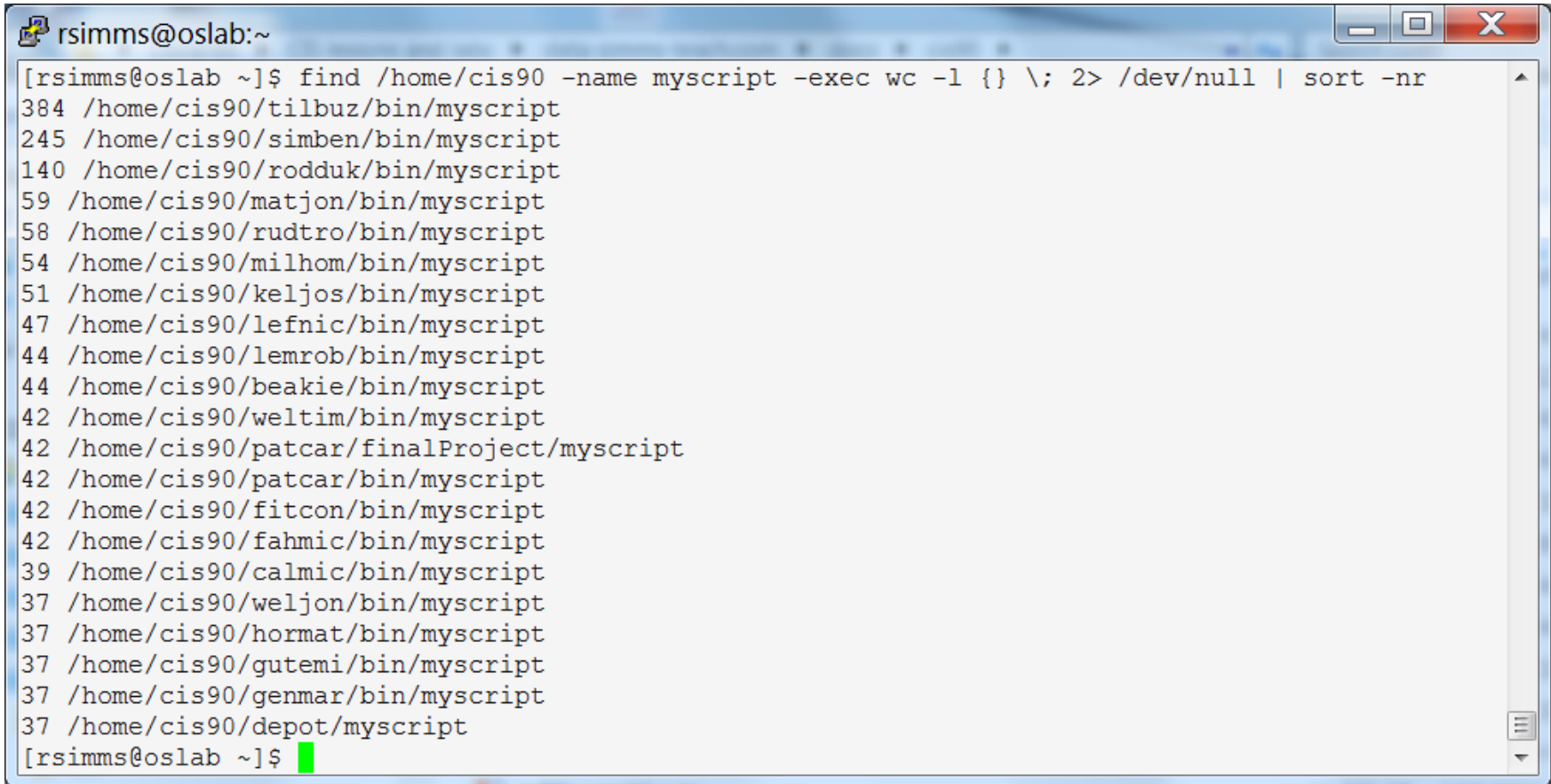


```
rsimms@oslab:~/cis90/project
[rsimms@oslab project]$ for i in `cat /etc/passwd | grep cis90 | cut -f6 -d":"`; do file=$i/bin/my
script; if [ -e "$file" ]; then echo -n $file exists ;if [ -r "$file" ]; then echo -n " and" is re
adable; else echo -n but is not readable; fi; if [ -x "$file" ]; then echo " and" executable; else
echo but not executable; fi; else echo $file does not exist; fi; done
/home/cis90/guest/bin/myscript does not exist
/home/cis90/cis/bin/myscript does not exist
/home/cis90/simben/bin/myscript exists and is readable and executable
/home/cis90/milhom/bin/myscript exists and is readable and executable
/home/cis90/rodduk/bin/myscript exists and is readable and executable
/home/cis90/beakie/bin/myscript exists and is readable and executable
/home/cis90/calmic/bin/myscript exists and is readable and executable
/home/cis90/casric/bin/myscript does not exist
/home/cis90/fahmic/bin/myscript exists and is readable and executable
/home/cis90/fitcon/bin/myscript exists and is readable and executable
/home/cis90/gutemi/bin/myscript exists and is readable and executable
/home/cis90/hormat/bin/myscript exists and is readable and executable
/home/cis90/keljos/bin/myscript exists and is readable and executable
/home/cis90/lefnic/bin/myscript exists and is readable and executable
/home/cis90/lehreb/bin/myscript does not exist
/home/cis90/lemrob/bin/myscript exists and is readable and executable
/home/cis90/patcar/bin/myscript exists and is readable and executable
/home/cis90/perste/bin/myscript does not exist
/home/cis90/ramjua/bin/myscript does not exist
/home/cis90/rudtro/bin/myscript exists and is readable and executable
/home/cis90/tilbuz/bin/myscript exists and is readable and executable
/home/cis90/vasjor/bin/myscript does not exist
/home/cis90/weltim/bin/myscript exists and is readable and executable
/home/cis90/mosmic/bin/myscript does not exist
/home/cis90/weljon/bin/myscript exists and is readable and executable
/home/cis90/matjon/bin/myscript exists and is readable and executable
/home/cis90/genmar/bin/myscript exists and is readable and executable
[rsimms@oslab project]$
```

*a one line
command
using semi-
colons!*

Project Status

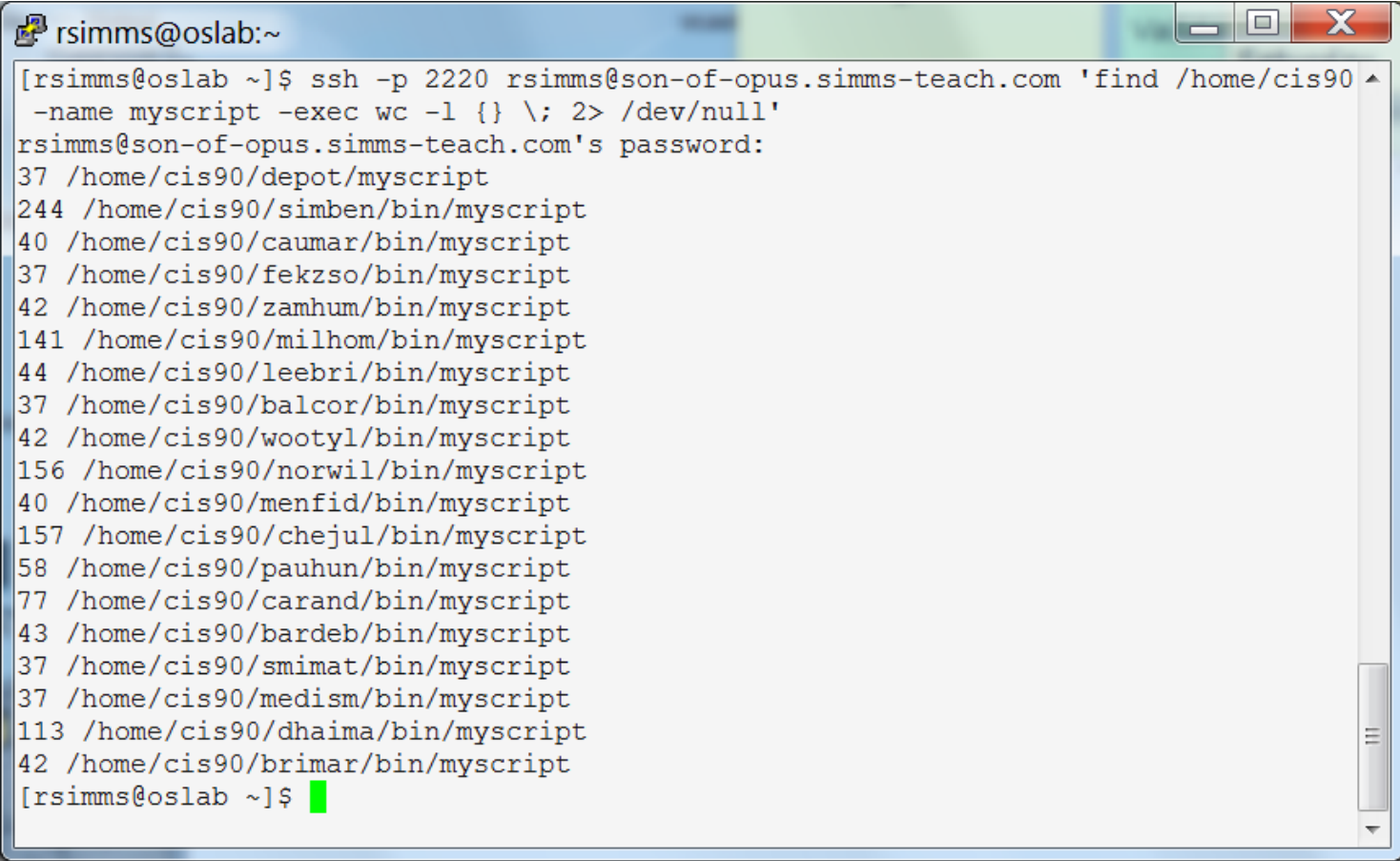
```
find /home/cis90 -name myscript -exec wc -l {} \; 2> /dev/null | sort -nr
```



```
rsimms@oslab:~  
[rsimms@oslab ~]$ find /home/cis90 -name myscript -exec wc -l {} \; 2> /dev/null | sort -nr  
384 /home/cis90/tilbuz/bin/myscript  
245 /home/cis90/simben/bin/myscript  
140 /home/cis90/rodduk/bin/myscript  
59 /home/cis90/matjon/bin/myscript  
58 /home/cis90/rudtro/bin/myscript  
54 /home/cis90/milhom/bin/myscript  
51 /home/cis90/keljos/bin/myscript  
47 /home/cis90/lefnic/bin/myscript  
44 /home/cis90/lemrob/bin/myscript  
44 /home/cis90/beakie/bin/myscript  
42 /home/cis90/weltim/bin/myscript  
42 /home/cis90/patcar/finalProject/myscript  
42 /home/cis90/patcar/bin/myscript  
42 /home/cis90/fitcon/bin/myscript  
42 /home/cis90/fahmic/bin/myscript  
39 /home/cis90/calmic/bin/myscript  
37 /home/cis90/weljon/bin/myscript  
37 /home/cis90/hormat/bin/myscript  
37 /home/cis90/gutemi/bin/myscript  
37 /home/cis90/genmar/bin/myscript  
37 /home/cis90/depot/myscript  
[rsimms@oslab ~]$
```

Project Status

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



```
rsimms@oslab:~
[rsimms@oslab ~]$ ssh -p 2220 rsimms@son-of-opus.simms-teach.com 'find /home/cis90
-name myscript -exec wc -l {} \; 2> /dev/null'
rsimms@son-of-opus.simms-teach.com's password:
37 /home/cis90/depot/myscript
244 /home/cis90/simben/bin/myscript
40 /home/cis90/caumar/bin/myscript
37 /home/cis90/fekzso/bin/myscript
42 /home/cis90/zamhum/bin/myscript
141 /home/cis90/milhom/bin/myscript
44 /home/cis90/leeabri/bin/myscript
37 /home/cis90/balcor/bin/myscript
42 /home/cis90/wootyl/bin/myscript
156 /home/cis90/norwil/bin/myscript
40 /home/cis90/menfid/bin/myscript
157 /home/cis90/chejul/bin/myscript
58 /home/cis90/pauhun/bin/myscript
77 /home/cis90/carand/bin/myscript
43 /home/cis90/bardeb/bin/myscript
37 /home/cis90/smimat/bin/myscript
37 /home/cis90/medism/bin/myscript
113 /home/cis90/dhaima/bin/myscript
42 /home/cis90/brimar/bin/myscript
[rsimms@oslab ~]$
```

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*

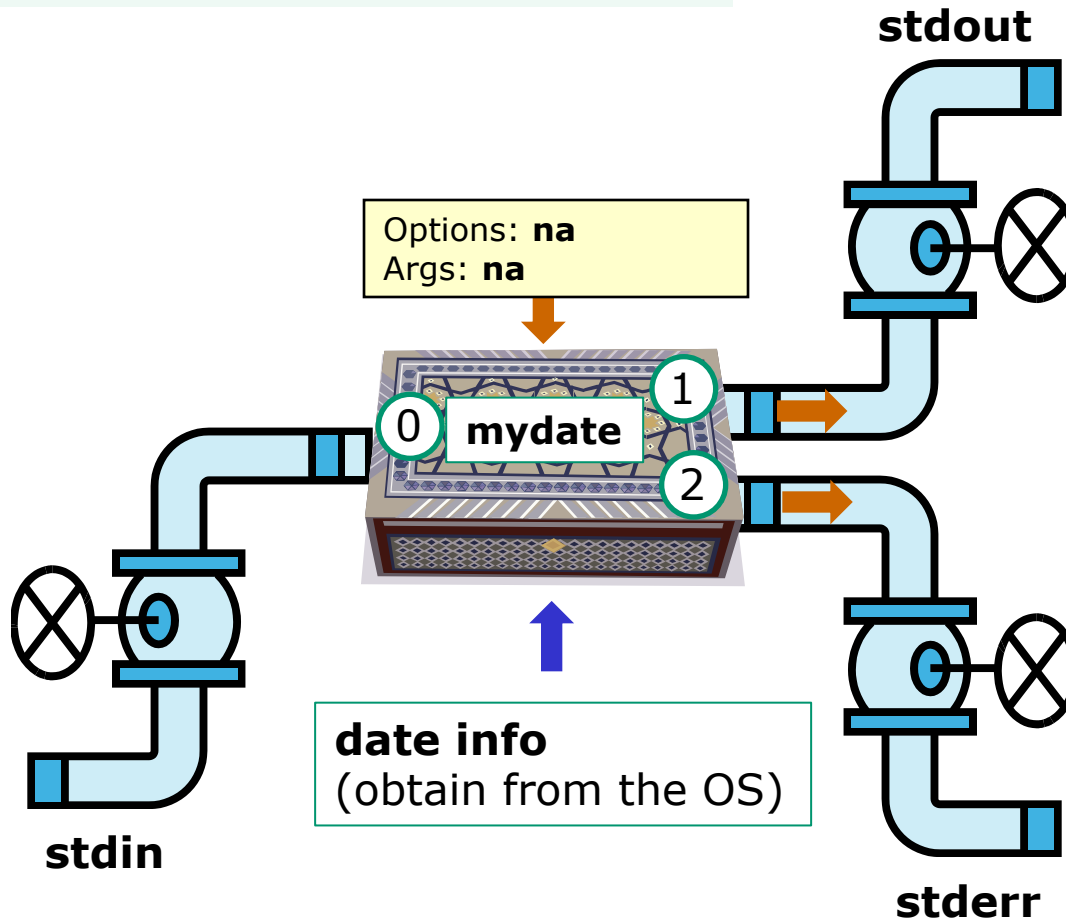
*Don't initialize
them yet*

```
/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
```



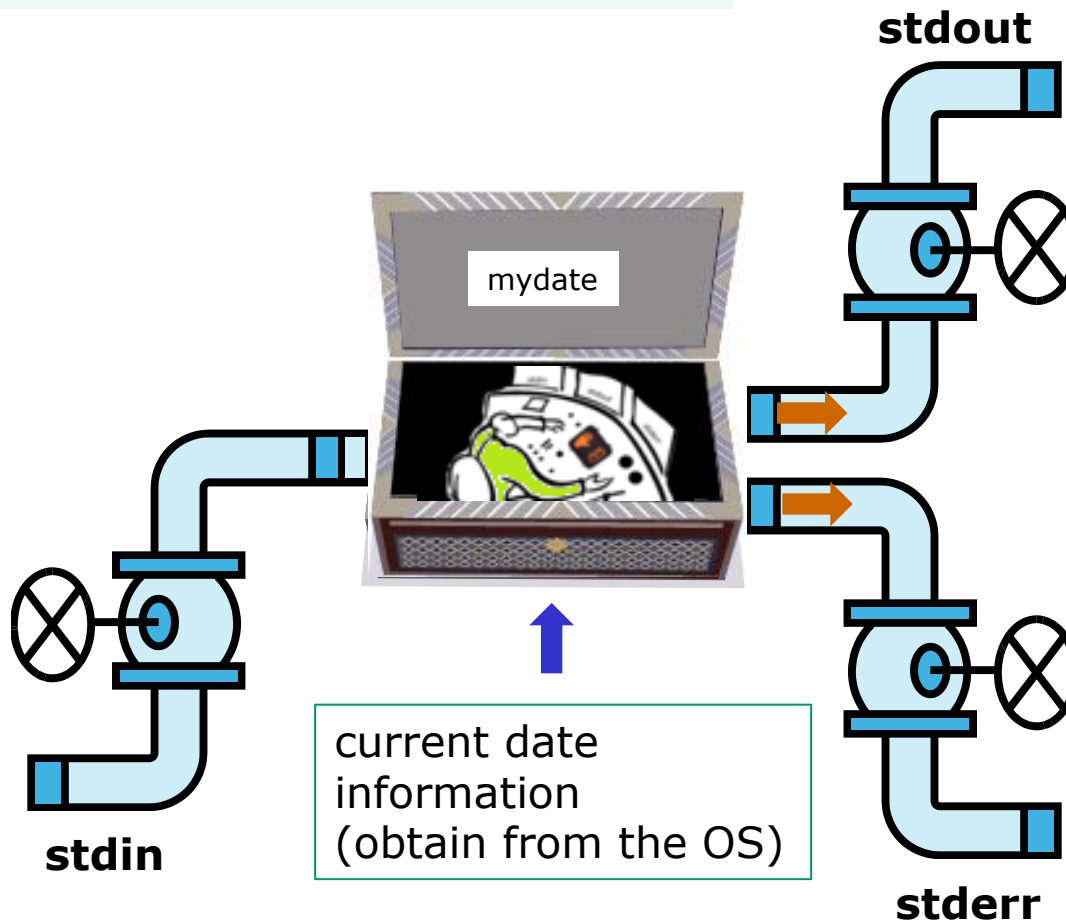
```
Hola simben90  
05/16/2013
```

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

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

Running a Script

```
$ mydate
```

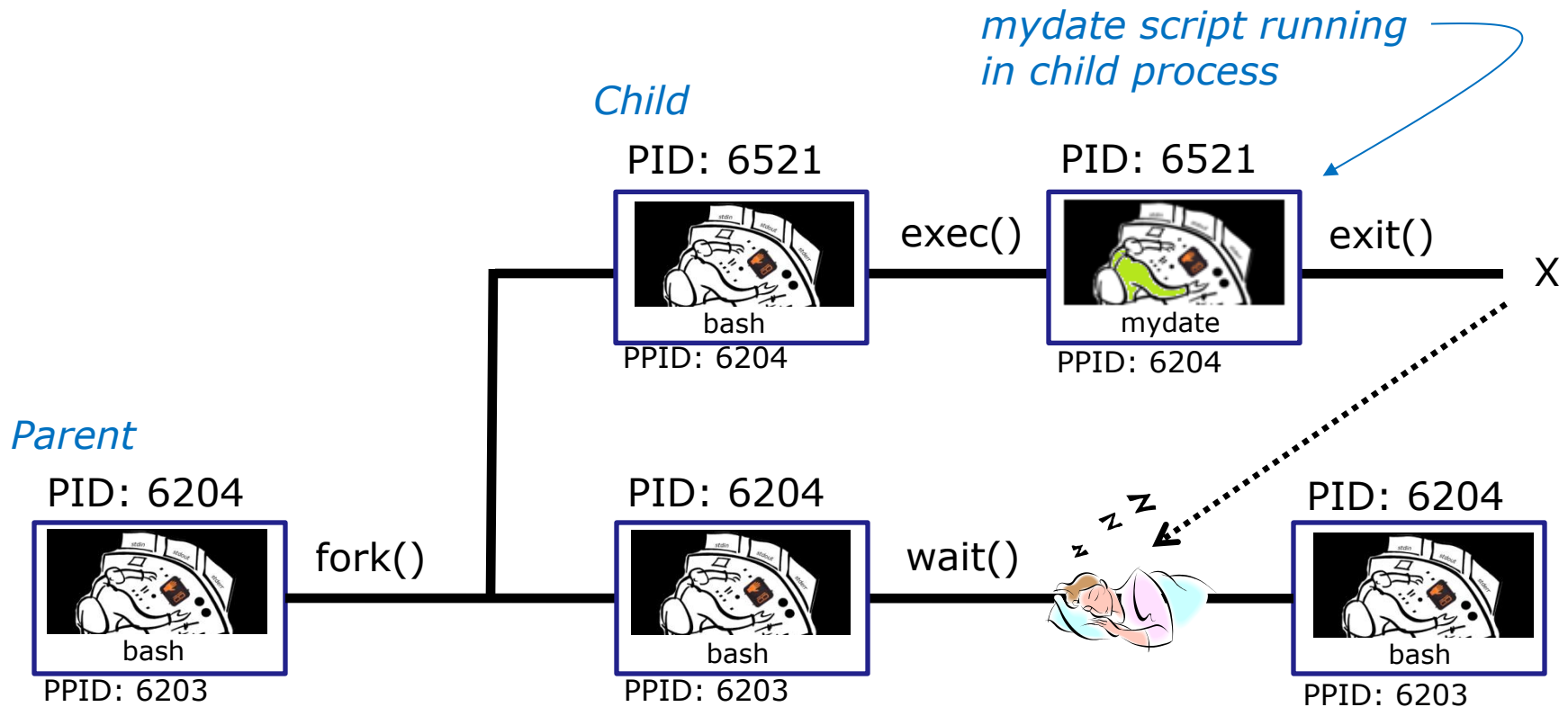


```
Hola simben90  
05/16/2013
```

*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
```

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

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

Why no Tic Tac Toe output?

Running a Script

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

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

```
/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
```

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
```

*Add these
new lines*

```
/home/cis90/simben $ mydate
```

```
Hola simben90
```

```
05/09/2012
```

```
Tic Tac Toe
```

```
red white blue
```

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

```
/home/cis90/simben $ echo $myvar1 $myvar2 $myvar3
```

```
Tic Tac Toe
```

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
```

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

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

```
}  
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

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

That thereby beauty's rose might never die,
    That beauty still may live in thine or thee.
Herein lives wisdom, beauty, and increase;
If I could write the beauty of your eyes,
And dig deep trenches in thy beauty's field,
Then being ask'd, where all thy beauty lies,
How much more praise deserv'd thy beauty's use,
Proving his beauty by succession thine.
Upon thyself thy beauty's legacy?

    Thy unus'd beauty must be tomb'd with thee,
Beauty's effect with beauty were bereft,
Yet mortal looks adore his beauty still,
But beauty's waste hath in the world an end,
And loved your beauty with love false or true,
Ready to count them?

14
Enter a new string to search for

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

```
milhom90@oslab:~/bin

1)  # Task 1 - grep command explored

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

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

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

2)  # Commands for Task 2
    ;;

3)  # Commands for Task 3
    ;;

4)  # Commands for Task 4
```

40,17 38%

Color Syntax

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

```
milhom90@oslab:~/bin

grep -h beauty /home/cis90/milhom/poems/*/*

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

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

2) # Commands for Task 2
;;

3) # Commands for Task 3
;;

4) # Commands for Task 4
;;

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

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

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

6) # Commands for Task 6
;;

7) # Commands for Task 7
;;
```

Use color syntax to spot unmatched quotes

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

Color Syntax

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

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

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

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

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

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

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

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

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

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

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

%s /oldstring/newstring/g



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 ticks 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:

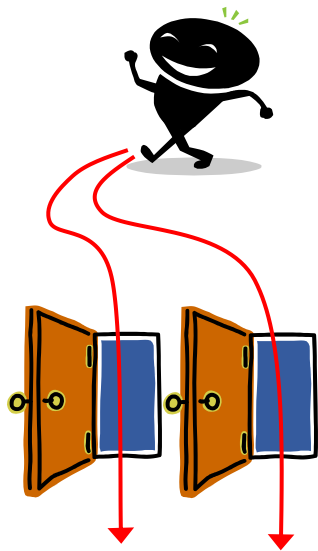
```
vi ~milhom90/bin/myscript
```

Scripting Tips

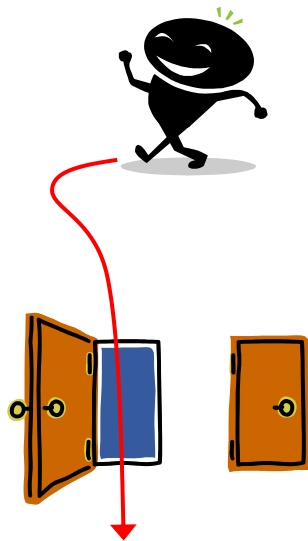
if statement with "or"

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

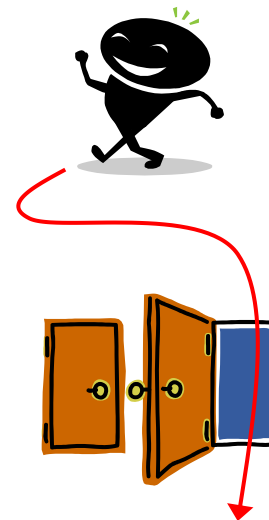
OR logic



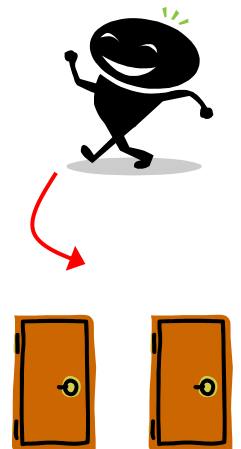
Yes



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

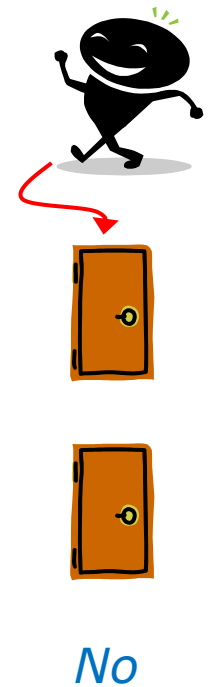
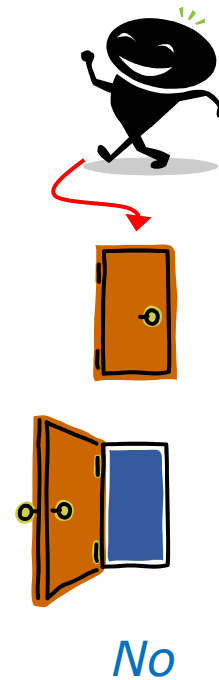
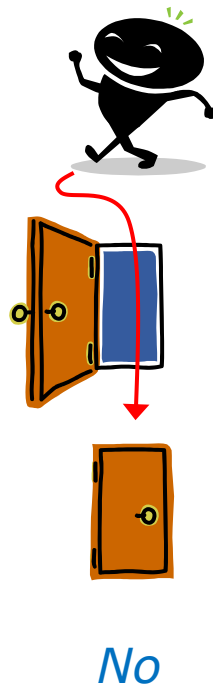
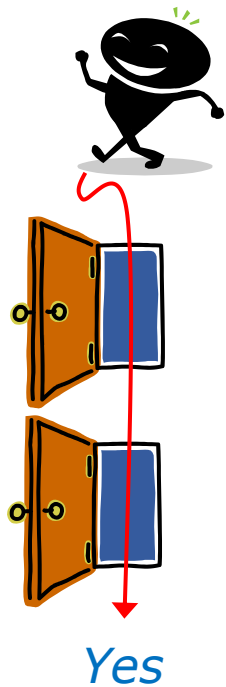
```
chmod +x example654  
example654
```

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

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

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

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

*tests to see
if it's a
directory*

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

myscript

Homer's CIS 90 Final Project

- 1) 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**

File



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

Directory





Additional file attributes to test for:

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

Class Exercise

Run the previous example task

- run **allscripts**
- select Homer's script
- select Task **4**

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

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

Scripting Tips

if then else statement

myscript

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

Homer's CIS 90 Final Project

- 1) 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 newsript old program quiet quiet.bak script treed tryme  
typescript zoom
```

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

```
[rsimms@opus bin]$ echo $3  
banner
```

```
[rsimms@opus bin]$ echo $7  
I
```

```
[rsimms@opus bin]$ echo $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

Dark Gray 1;30

Blue 0;34

Light Blue 1;34

Green 0;32

Light Green 1;32

Cyan 0;36

Light Cyan 1;36

Red 0;31

Light Red 1;31

Purple 0;35

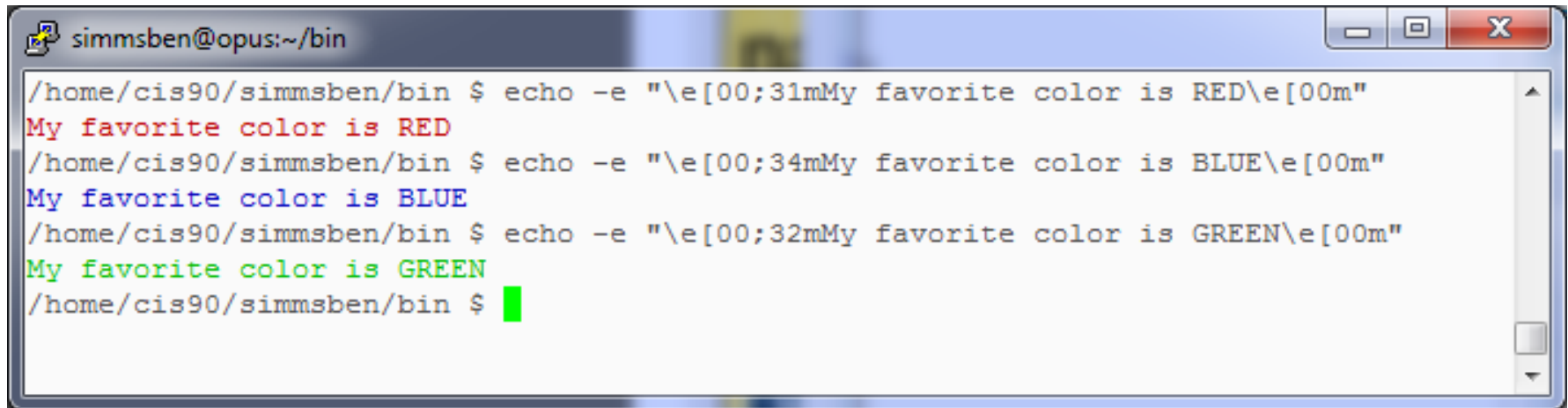
Light Purple 1;35

Brown 0;33

Yellow 1;33

Light Gray 0;37

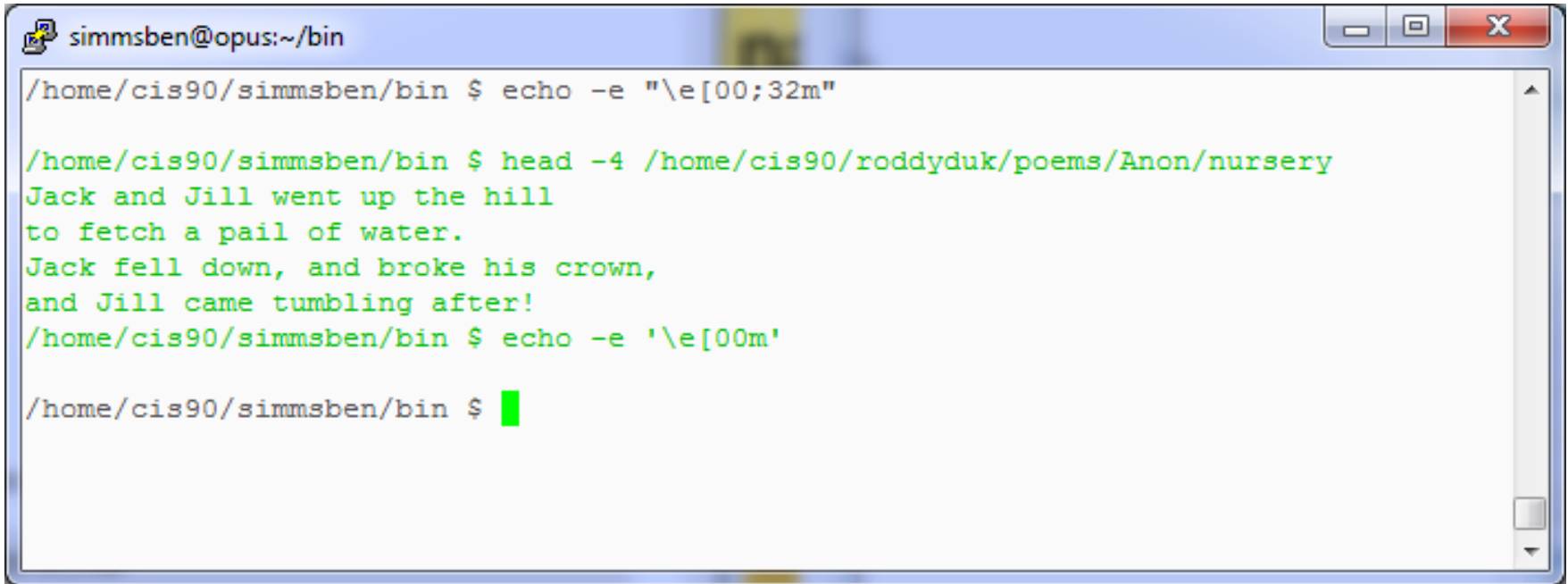
White 1;37

A terminal window titled 'simmsben@opus:~/bin' showing three commands and their color-coded outputs. The first command uses '\e[00;31m' to print 'My favorite color is RED' in red. The second uses '\e[00;34m' to print 'My favorite color is BLUE' in blue. The third uses '\e[00;32m' to print 'My favorite color is GREEN' in green. The prompt is a green block character.

```
simmsben@opus:~/bin
/home/cis90/simmsben/bin $ echo -e "\e[00;31mMy favorite color is RED\e[00m"
My favorite color is RED
/home/cis90/simmsben/bin $ echo -e "\e[00;34mMy favorite color is BLUE\e[00m"
My favorite color is BLUE
/home/cis90/simmsben/bin $ echo -e "\e[00;32mMy favorite color is GREEN\e[00m"
My favorite color is GREEN
/home/cis90/simmsben/bin $ █
```

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

Using Color

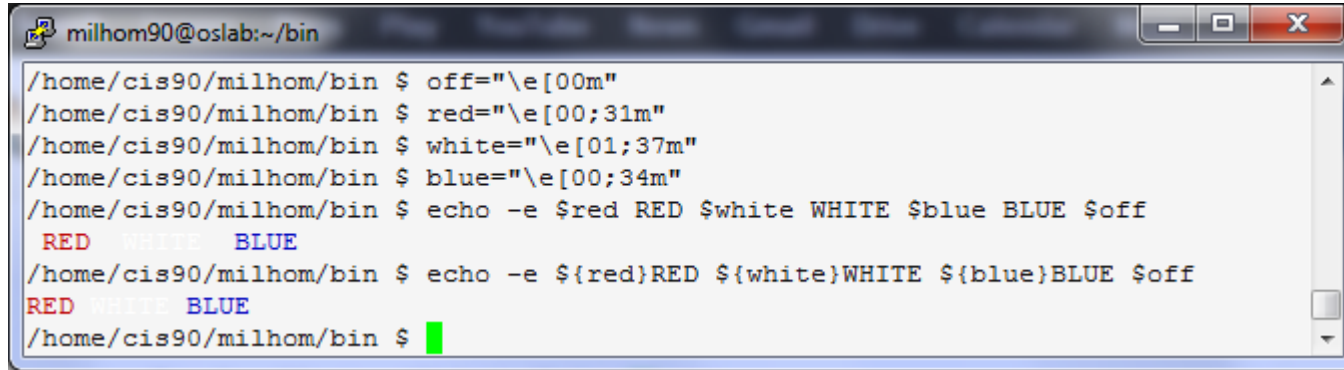
A terminal window titled 'simmsben@opus:~/bin' with standard window controls. The terminal shows a sequence of commands and their outputs. The first command 'echo -e "\e[00;32m"' results in a blank line. The second command 'head -4 /home/cis90/roddyduk/poems/Anon/nursery' outputs four lines of text in green. The third command 'echo -e '\e[00m'' results in a blank line. The prompt is followed by a green cursor.

```
simmsben@opus:~/bin
/home/cis90/simmsben/bin $ echo -e "\e[00;32m"

/home/cis90/simmsben/bin $ head -4 /home/cis90/roddyduk/poems/Anon/nursery
Jack and Jill went up the hill
to fetch a pail of water.
Jack fell down, and broke his crown,
and Jill came tumbling after!
/home/cis90/simmsben/bin $ echo -e '\e[00m'

/home/cis90/simmsben/bin $ █
```

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



```

milhom90@oslab:~/bin
/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
/home/cis90/milhom/bin $ █
  
```

```

/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
scp
tar
if then else
[]

- extract filename from pathname
- secure copy command
- archive command
- 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