



## Lesson Module Checklist

- Slides
- WB converted
  
- 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: **888-886-3951**

Passcode: **136690**



Francisco



Chris



Justin



Jesus



Shenghong



Paul



Roberto



Sam



Navin



Jimmy



Luis



Tommy



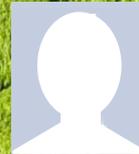
Abraham



Ann



Cameron



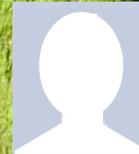
Cody



Alejandrino



Deane



Nadia



Richard Z.



Gabriel



Ryan



Takashi



Jeff



Nick



Jonathan



Shea



Matthew



James



Richard I.



Nicole

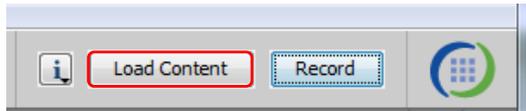


Aaron



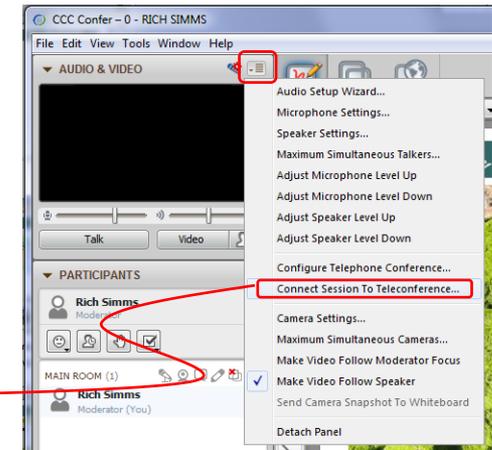
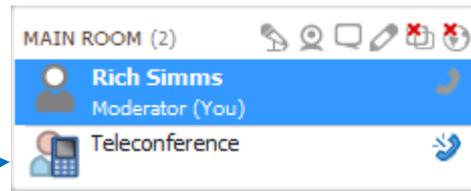
# Instructor CCC Confer checklist

[ ] Preload White Board

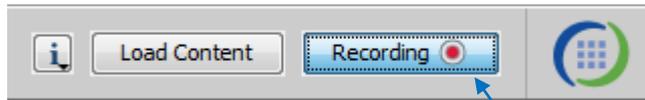


[ ] Connect session to Teleconference

*Session now connected to teleconference*



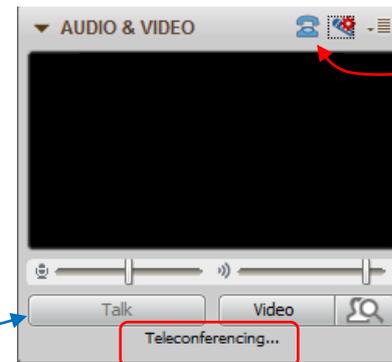
[ ] Is recording on?



*Red dot means recording*

[ ] Use teleconferencing, not mic

*Should be greyed out*



*Should show as this live "off hook" telephone handset icon and the Teleconferencing ... message displayed*



## Instructor CCC Confer checklist

The screenshot displays a Windows desktop with several applications open:

- CCC Confer**: A video conferencing window on the left showing a participant named Rich Simms. It includes controls for audio and video, a list of participants, and a chat window.
- foxit for slides**: A Foxit Reader window in the center-left displaying a PDF document titled 'cis90lesson07.pdf'. A red callout points to the application.
- chrome**: A Google Chrome browser window in the top-right displaying a webpage from 'simms-teach.com/docs/cis90/cis-90-TEST-1-Fall-12.pdf'. The page contains flashcard questions and answers. A red callout points to the browser.
- putty**: A terminal window in the center-bottom showing a login session for 'simben90@oslab:~'. The prompt is '/home/cis90/simben \$'. A red callout points to the terminal.
- vSphere Client**: A vCenter console window in the bottom-right showing the vSphere Client interface with a tree view of virtual machines. A red callout points to the application.

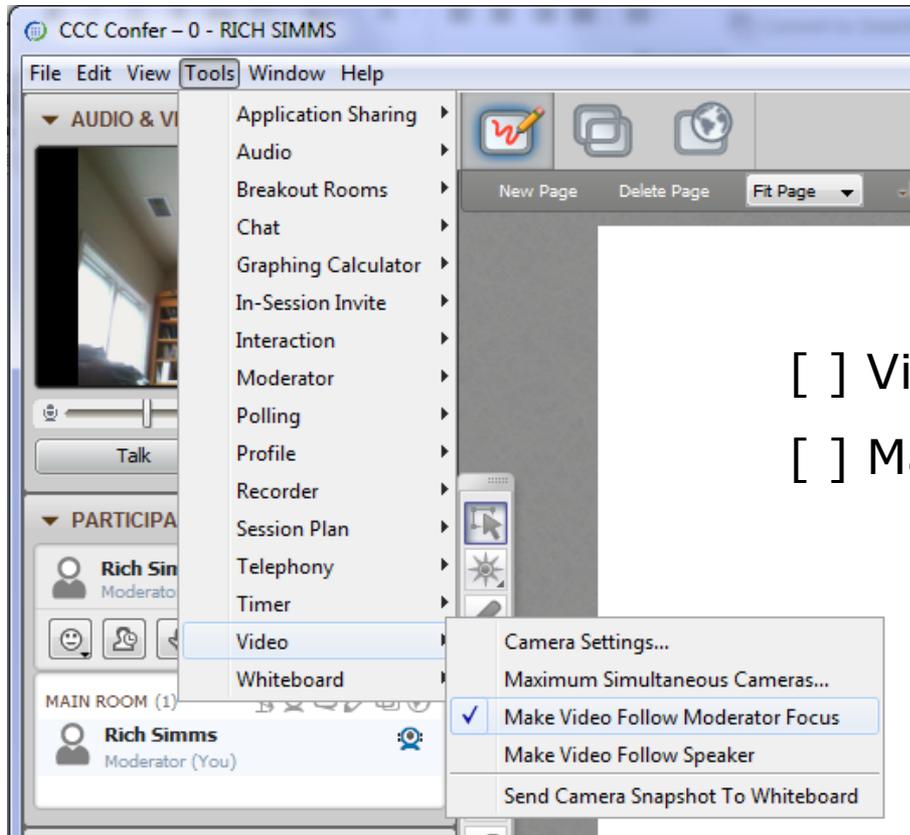
The desktop taskbar at the bottom shows icons for various applications, including Internet Explorer, Firefox, Chrome, and several instances of Microsoft Word and PowerPoint. The system tray in the bottom-right corner shows the time as 6:52 AM on 10/10/2012.

[ ] layout and share apps





## Instructor CCC Confer checklist

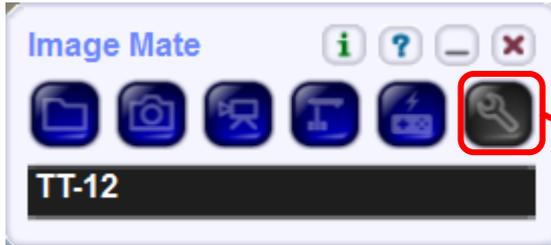


[ ] Video (webcam)

[ ] Make Video Follow Moderator Focus



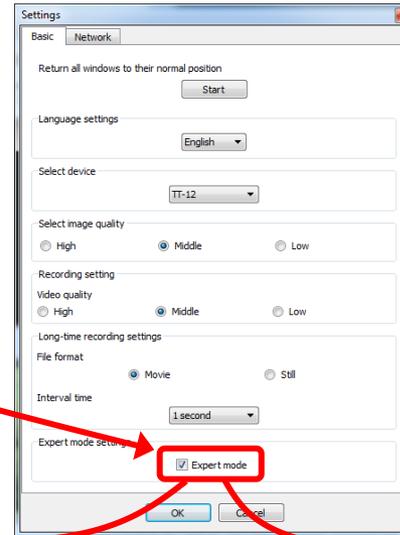
# Using Elmo with CCC Confer



Elmo rotated down to view side table



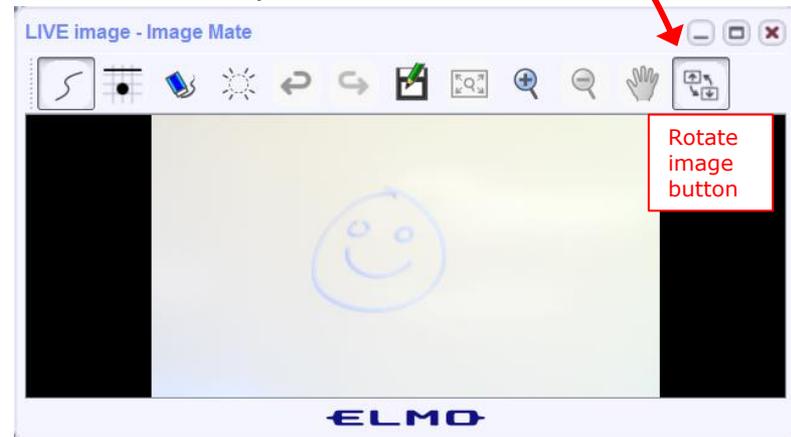
Run and share the Image Mate program just as you would any other app with CCC Confer



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





# Instructor CCC Confer checklist

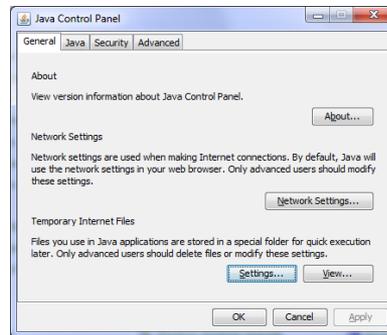
Universal Fix for CCC Confer:

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

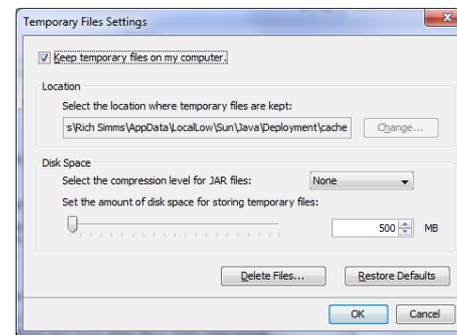
Control Panel (small icons)



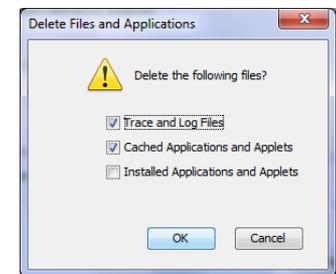
General Tab > Settings...



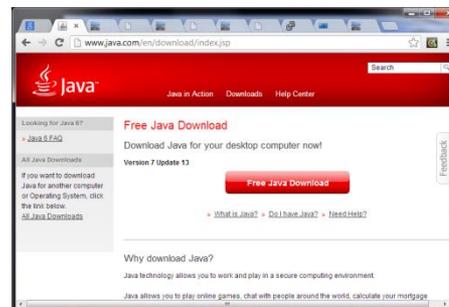
500MB cache size



Delete these



Google Java download



## Quiz

Please answer these questions **in the order** shown:

**See electronic white board**

**email answers to: [risimms@cabrillo.edu](mailto:risimms@cabrillo.edu)**

**(answers must be emailed within the first few minutes of class for credit)** 9

## Quiz

**No Quiz  
Today !**

## 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 branch>* *<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 branch> <target directory>



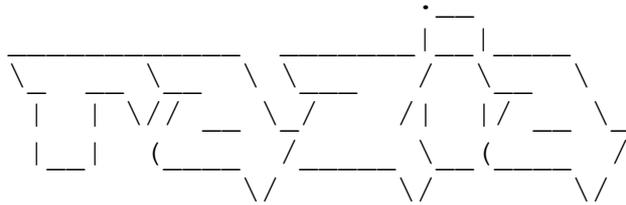
When copying files between systems it is necessary to use scp and specify non-standard **ports**, **login usernames**, and **hostnames** for the remote system

## Copy a file from Opus to Razia

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

Password:

Last login: Tue May 6 09:02:26 2014 from opus.cis.cabrillo.edu



Have a lot of fun...

*Let's log into a different system. This could be your home system for example.*

*Let's copy one file from Opus to this system*

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

```
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
```

*If you copy tryme to a directory that is not on your path then use ./tryme to run it*

## 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*

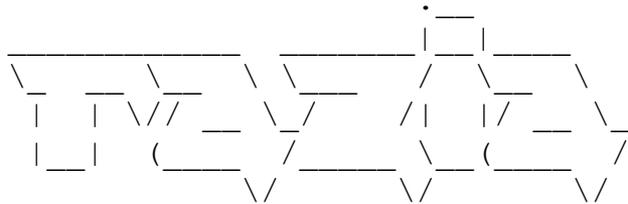
*. ("here") is a shortcut for specifying the current directory on the local system as the target directory*

## Copying a branch of the tree from Opus to Razia

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

Password:

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



Have a lot of fun...

*Login to the other system  
again if needed*

*Copy your local bin directory  
on Opus and EVERYTHING in  
it to this system*

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

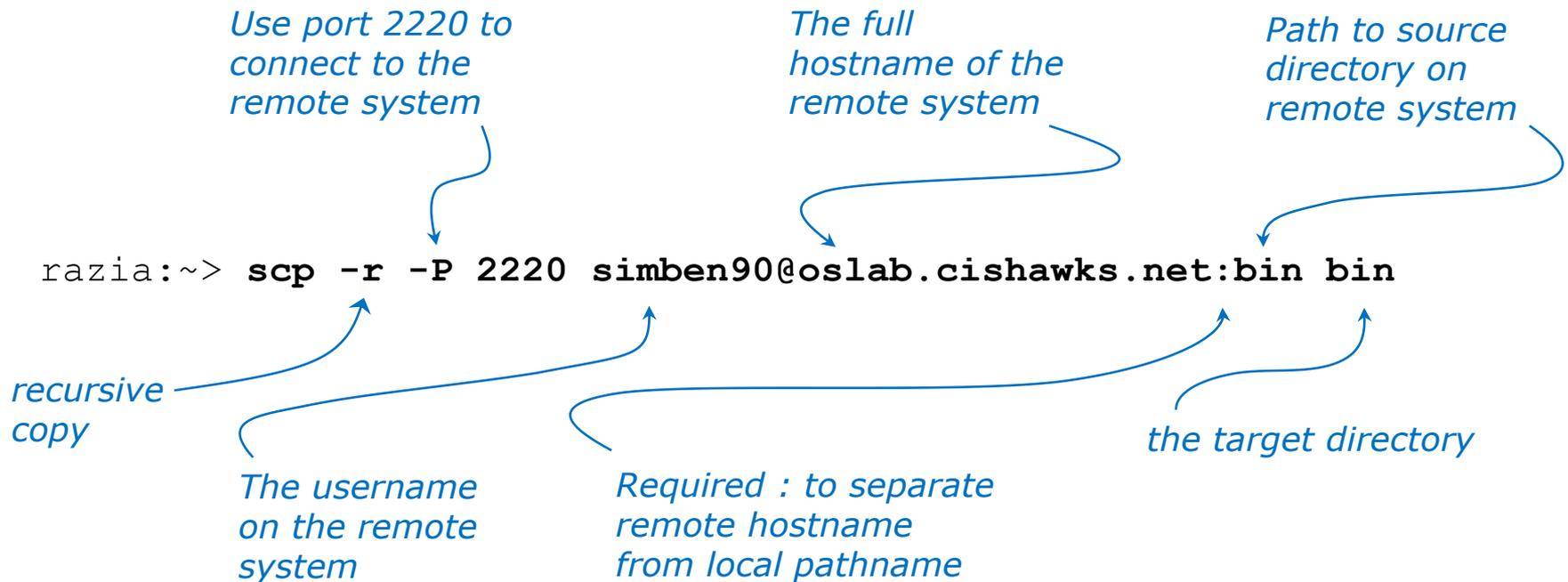
```
simben90@oslab.cishawks.net's password:
```

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 branch> <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 your Arya:

Copy your myscript files to your Arya VM:

```
ssh cis90@arya-xx Use your own Arya VM
```

```
hostname
```

```
mkdir bin
```

```
scp -P 2220 xxxxxxx90@oslab.cis.cabrillo.edu:bin/myscript* bin
```

*Use your own Opus username*

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



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  tryme
banner  enlightenment home    myscript  treed        zoom
```

```
/home/cis90/simben $ tar cvf bin.tar bin/
```

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

*create  
verbose  
file*

*name of  
archive file  
(tarball)*

*pathname  
to directory  
to archive*

# Example

## Copy archived directory to another system

*View your bin archive*

```
/home/cis90/simben $ ls -l bin.tar
-rw-rw----. 1 simben90 cis90 40960 Dec  2 07:47 bin.tar

/home/cis90/simben $ tar tvf bin.tar
drwxr-x--- simben90/cis90      0 2014-12-02 07:41 bin/
-r-xr-xr-- simben90/cis90 3442 2014-08-06 11:52 bin/enlightenment
-r-xr-x--- simben90/cis90   190 2001-07-20 15:04 bin/treed
-r-xr-x--- simben90/cis90    74 2001-07-20 15:18 bin/zoom
-rwxrwx--x simben90/cis90   546 2014-12-02 07:40 bin/myscript.v1
-r-xr-x--- simben90/cis90   220 2004-04-22 18:51 bin/app
-rwxr-xr-x simben90/cis90   103 2014-11-13 10:16 bin/home
-r-xr-x--- simben90/cis90   107 2001-07-20 21:06 bin/hi
-rwxrwxr-x simben90/cis90 10513 2014-12-02 07:41 bin/myscript
-r-xr-x--- simben90/cis90   375 2003-10-20 18:36 bin/I
-r-xr-x--- simben90/cis90   174 2004-03-04 13:02 bin/tryme
-r-xr-x--- simben90/cis90   519 2014-08-06 11:53 bin/datecal
-r-xr-x--- simben90/cis90  6160 2003-08-28 22:39 bin/banner
/home/cis90/simben $
```



# Example

Copy archived directory to another system

*username* *hostname* *port* *path to tar file* *"here"*

```
cis90@Arya-35:~$ scp -P 2220 simben90@oslab.cis.cabrillo.edu:bin.tar .  
simben90@oslab.cis.cabrillo.edu's password:  
bin.tar 100% 40KB 40.0KB/s  
00:00
```

```
cis90@Arya-35:~$ ls -l bin.tar  
-rw-rw---- 1 cis90 cis90 40960 Dec 2 07:52 bin.tar  
cis90@Arya-35:~$
```

*Note how  
archive files are  
shown in red*

*Copy your bin archive from Opus to Arya*

# Example

Copy archived directory to another system

```
cis90@Arya-35:~$ tar xvf bin.tar
bin/
bin/enlightenment
bin/treed
bin/zoom
bin/myscript.v1
bin/app
bin/home
bin/hi
bin/myscript
bin/I
bin/tryme
bin/datecal
bin/banner
cis90@Arya-35:~$
```

*extract  
verbose  
file*

*name of  
archive file  
(tarball)*

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

```
cis90@Arya-35:~$ ls bin
app      datecal      hi      I      myscript.v1  tryme
banner  enlightenment  home  myscript  treed      zoom
cis90@Arya-35:~$
```

# Example

## Copy archived directory to another system

```
cis90@Arya-35:~$ echo $PATH  
/home/cis90/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/  
bin:/usr/games:/usr/local/games  
cis90@Arya-35:~$
```

*Since the bin directory is on the cis90 user's path we can run myscript by typing its name. Otherwise we would have to type a full pathname to it.*

```
cis90@Arya-35:~$ myscript  
/home/cis90/bin/myscript: line 44: finger: command not found  
What is your first name? ^C  
cis90@Arya-35:~$
```

 *Hit Ctrl-C to abort myscript*

*Oops ... the finger command used by Benji's script has not been installed on Arya*

# Example

## Copy archived directory to another system

```
cis90@Arya-35:~$ sudo apt-get install finger
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  finger
0 upgraded, 1 newly installed, 0 to remove and 145 not upgraded.
Need to get 0 B/17.3 kB of archives.
After this operation, 68.6 kB of additional disk space will be used.
Selecting previously unselected package finger.
(Reading database ... 290787 files and directories currently installed.)
Preparing to unpack .../finger_0.17-15_amd64.deb ...
Unpacking finger (0.17-15) ...
Processing triggers for man-db (2.6.7.1-1) ...
Setting up finger (0.17-15) ...
cis90@Arya-35:~$
```

*Use sudo to install  
finger as the root  
supersuser*

# Example

## Copy archived directory to another system

*Run myscript file in the bin directory*

```
cis90@Arya-35:~$ myscript
```

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

- 1) What is today?
- 2) The users on Arya-35
- 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@Arya-35:~$
```



# Housekeeping

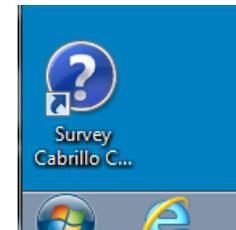


## Next Class

**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. December 18<sup>th</sup> (see calendar)
  - Final Exam (Test #3) 1-3:50AM.
  - Extra credit labs are due by 11:59PM .
5. Survey on room 828 equipment  
<https://www.surveymonkey.com/s/CTEFall2014>



# 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
```

## Spring 2015 Linux Classes and Prerequisites

### 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. Prerequisite: CIS 72. Transfer Credit: CSU.

Section	Days	Times	Units	Instructor	Room
88445	W	09:00AM-12:05PM	3.00	R.Simms	OL
&	Arr.	Arr.		R.Simms	OL

Section 88445 is an ONLINE course. Meets weekly throughout the semester online during the scheduled times by remote technology with an additional 50 min arranged online lab per week. For details, see instructor's web page at [go.cabrillo.edu/online](http://go.cabrillo.edu/online).

Section	Days	Times	Units	Instructor	Room
88446	W	09:00AM-12:05PM	3.00	R.Simms	828
&	Arr.	Arr.		R.Simms	OL

Section 88446 is a Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 50 min online lab per week. For details, see instructor's web page at [go.cabrillo.edu/online](http://go.cabrillo.edu/online).

### CIS 81 Networking Fundamentals and Theory (Cisco CCNA 1)

Presents networking protocols, standards, concepts, and terminology including Ethernet, ARP, ICMP, IP addressing, subnetting, switches, hubs, routers, TCP, UDP, OSI Model and other standards and protocols. Hybrid Requisite: Completion of or concurrent enrollment in CIS 72. Recommended Preparation: Eligibility for MATH 154.

Transfer Credit: CSU.

Section	Days	Times	Units	Instructor	Room
87038	M	05:30PM-09:35PM	4.00	R.Graziani	828
&	Arr.	Arr.		R.Graziani	OL

Section 87038 is a Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 50 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](http://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 skills. Recommended Preparation: CIS 191AB

Section	Days	Times	Units	Instructor	Room
88451	Arr.	Arr.	4.00	M.Matera	OL

Section 88451 is an ONLINE course. For details, see instructor's web page at [go.cabrillo.edu/online](http://go.cabrillo.edu/online).

Section	Days	Times	Units	Instructor	Room
88453	TH	05:30PM-09:35PM	4.00	M.Matera	828
&	Arr.	Arr.		M.Matera	OL

Section 88453 is an Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 50 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](http://go.cabrillo.edu/online).

### CIS 193AB UNIX/Linux Security Administration

Teaches how to perform the tasks and examine the strategies of UNIX/Linux host, files, and network security management. Prerequisites: CIS 192AB Recommended Preparation: CIS 175

Section	Days	Times	Units	Instructor	Room
88454	Arr.	Arr.	4.00	M.Matera	OL

Section 88454 is an ONLINE course. For details, see instructor's web page at [go.cabrillo.edu/online](http://go.cabrillo.edu/online).

Section	Days	Times	Units	Instructor	Room
88455	TH	10:00AM-02:05PM	4.00	M.Matera	829
&	Arr.	Arr.		M.Matera	OL

Section 88455 is a Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 50 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](http://go.cabrillo.edu/online).

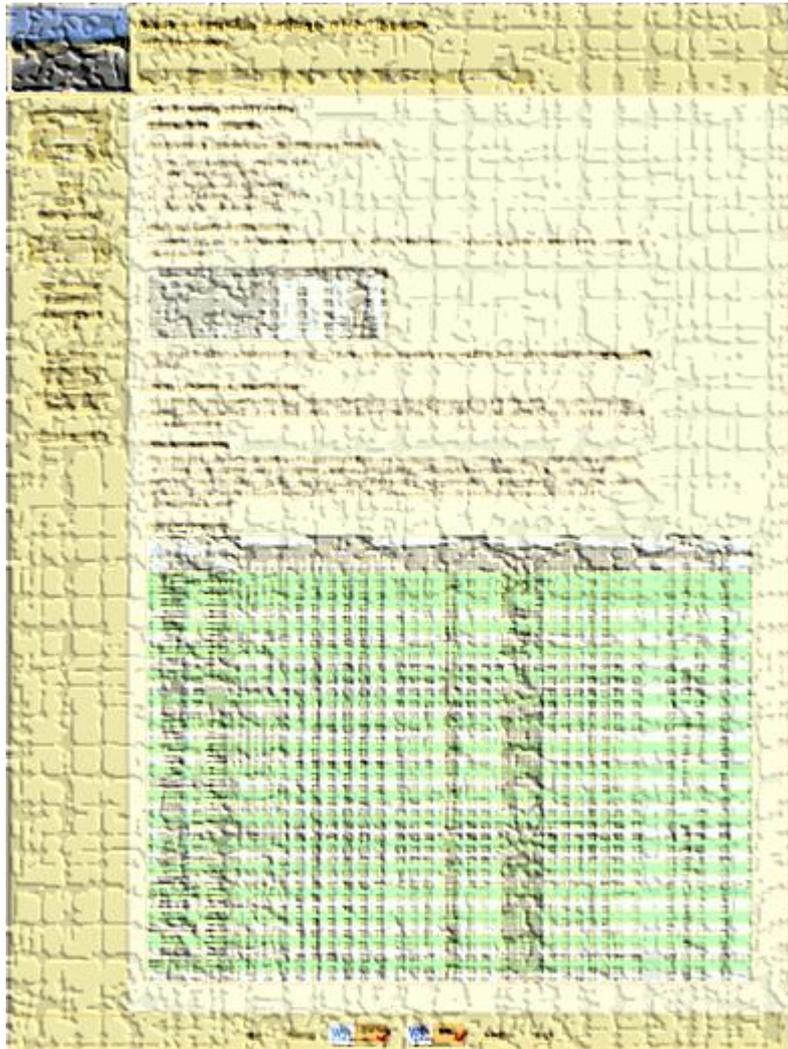
## Final Exam

Test #3 (final exam) is **THURSDAY** December 18 1:00-3:50PM

	12/18	<p><b>Test #3 (the final exam)</b></p> <p><b>Time</b></p> <ul style="list-style-type: none"> <li>• 1:00PM - 3:50PM in Room 828</li> </ul> <p><b>Materials</b></p> <ul style="list-style-type: none"> <li>• Test (<a href="#">blackboard</a>)</li> </ul> <p><b>CCC Confer</b></p> <ul style="list-style-type: none"> <li>• <a href="#">Enter virtual classroom</a></li> <li>• <a href="#">Class archives</a></li> </ul>		<p><a href="#">5 posts</a></p> <p><a href="#">Lab X1</a></p> <p><a href="#">Lab X2</a></p>
--	-------	--	--	--

- All students will take the test at the same time.
- Working students will need to plan ahead to take time off from work for the test.

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



The screenshot shows a web browser window with a yellow background. The main content is a table with multiple columns and rows, likely representing student names and their scores. The table is partially obscured by a large, semi-transparent watermark that reads 'GRADE' in large, bold, blue letters. The table has several columns, with the first column containing names and the subsequent columns containing numerical values. The text is small and difficult to read due to the watermark and the low resolution of the screenshot.

# 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

### Points that could have been earned:

10 quizzes:	30 points
10 labs:	300 points
2 tests:	60 points
3 forum quarters:	60 points
<b>Total:</b>	<b>450 points</b>

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

*Use Sam's new Python command to see how many points you have earned and how many more you need for the grade you want:*

```
grades <LOR codename>
```

*Use Sam's new Python command to see when the next assignments are due:*

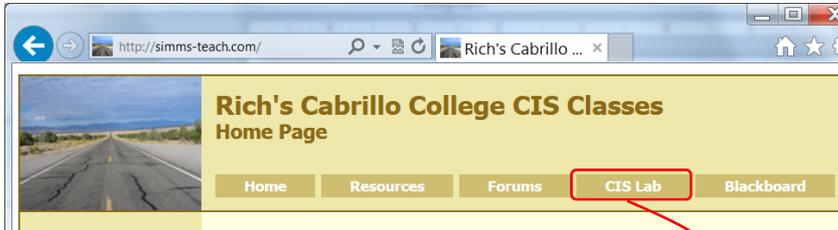
```
schedule
```

*Jesse's script is still available too:*

```
checkgrades <LOR codename>
```

## CIS Lab Schedule

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

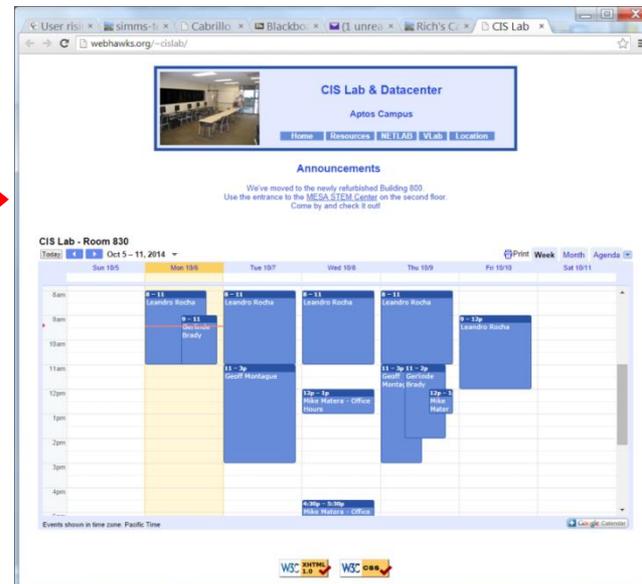


*Not submitting tests or lab work?*

*If you would like some additional help come over to the CIS Lab.*

*Leandro, Geoff and Nick  
(Mondays 10-4) are all CIS 90  
Alumni.*

*Michael is the other Linux  
instructor.*



Or hang around after class. Rich has his office hours right after each class in Room 828.

# Free CIS 90 Tutoring Available

<http://www.cabrillo.edu/services/tutorials/>

The screenshot shows the website for the Tutorials Center at Cabrillo College. The page is titled 'TUTORIALS' and includes a navigation menu with options like 'ABOUT', 'ACADEMICS/CAREERS', 'ADMISSIONS', 'CLASS SCHEDULES', 'REGISTRATION', and 'WEBADVISOR'. The main content area is divided into several sections:

- ANNOUNCEMENTS & DEADLINES:** Lists new subjects for Spring 2014, including American Sign Language, Computer Applications/Business Technology (CABT), Computer and Information Systems (CIS), and History 17A.
- Welcome to the Tutorials Center!:** States that free peer tutoring is offered to students. It lists several conditions: tutoring is by appointment, sessions are weekly, sessions are in small groups (1-2 hours), and students should come directly to the TC office to schedule.
- The following classes are being tutored for Spring 2014:** A list of classes including Accounting, American Sign Language, Biology, Computer Applications/Business Technology (CABT), **Computer and Information Systems (CIS) 81, 90, 172** (highlighted with a red box), and Chemistry.
- CONTACT INFORMATION:** Provides the location (Room 1080A), phone number (831.479.6470), email (tutorialscenter@cabrillo.edu), and coordinator (Lori Chavez).



Matt Smithey

All students interested in tutoring in CIS 90, 172, and 81 classes need to come directly to the Tutorials Center to schedule, register and fill out some paperwork. This is just a one-time visit.

The tutoring will take place at the STEM center and they will log in and log out on a computer you have designated (I will figure out exactly what that means).

Matt is available M: 9:00-5:00, T: 9-11 and 2-5, Wed: 9-12 and Th: 9-11 and 3-5.

# More Free CIS 90 Tutoring Available

The screenshot shows a web browser window displaying a forum post on the phpBB platform. The browser tabs include "(1 unread) - rich", "User risimms loc", "Cabrillo College", and "Rich's Cabrillo C". The address bar shows the URL "oslab.cis.cabrillo.edu/forum/viewtopic.php?f=101&t=3324&sid=63dda9cf0a544936a540e216474d4c16".

The forum header is blue and features the phpBB logo with the tagline "creating communities". The forum title is "Cabrillo College: Computer and Information Systems", described as a "Forum for students in the Computer Networking and System Administration and/or Computer Support Specialist programs". A search bar and "Advanced search" link are present.

The breadcrumb trail reads: "Board index < Cabrillo College Fall 2014 Courses < CIS 90 - Fall 2014". Navigation links for "FAQ", "Register", and "Login" are visible.

The main post title is "Do you need tutoring?". It includes a "POSTREPLY" button, a search box for the topic, and a "Search" button. The post statistics show "5 posts • Page 1 of 1".

The post content, by user "Takashi Tamasu" (joined Wed Jan 29, 2014, 3:46 pm), reads: "I belong to the AGS (Alpha Gamma Sigma) Honor Society at Cabrillo and one of the functions this club does is offer FREE tutoring. One of the tutors listed CIS 90 as one of the classes that he is willing to tutor. If someone needs tutoring you can either submit a tutor request form at our site or tell me a number and when you can be reached at that number to arrange tutoring. [https://sites.google.com/site/cabrilloa ... edirects=0](https://sites.google.com/site/cabrilloa...edirects=0)

BTW I am the tutor coordinator for AGS

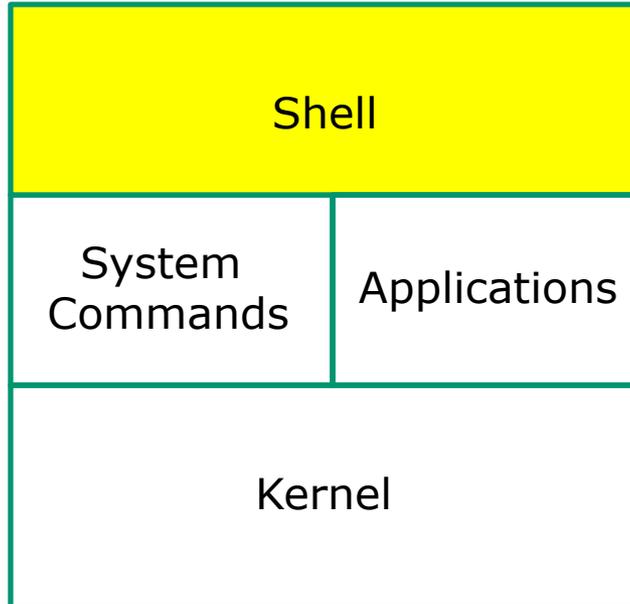
cheers Takashi



# 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): <ul style="list-style-type: none"> <li>Requirements for each task:               <ul style="list-style-type: none"> <li>Minimum of 10 "original" script command lines</li> <li>Has one or more non-generic comments to explain what it is doing</li> <li>Has user interaction</li> </ul> </li> </ul>
25	You don't have to do all of these but do at least five: <ul style="list-style-type: none"> <li>Redirecting stdin (5 points)</li> <li>Redirecting stdout (5 points)</li> <li>Redirecting stderr (5 points)</li> <li>Use of permissions (5 points)</li> <li>Use of filename expansion characters (5 points)</li> <li>Use of absolute path (5 points)</li> <li>Use of relative path (5 points)</li> <li>Use of a PID (5 points)</li> <li>Use of inodes (5 points)</li> <li>Use of links (5 points)</li> <li>Use of scheduling (5 points)</li> <li>Use of a GID or group (5 points)</li> <li>Use of a UID or user (5 points)</li> <li>Use of a /dev/tty device (5 points)</li> <li>Use of a signal (5 points)</li> <li>Use of piping (5 points)</li> <li>Use of an environment variable (5 points)</li> <li>Use of /bin/mail (5 points)</li> <li>Use of a conditional (5 points)</li> </ul> The maximum for this section is 25 points.
5	Present your script to the class
<b>Points lost</b>	
-15	Fails to run from <b>allscripts</b>
-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: <ul style="list-style-type: none"> <li>Doesn't give full credit to the original author</li> <li>Doesn't indicate where the code was obtained from</li> <li>Doesn't include licensing terms</li> <li>Violates copyright or licensing terms</li> </ul>
<b>Extra credit</b>	
30	Up to three additional tasks (10 points each)

*This applies to each individual task*

*This applies to the project as a whole*

```
rsimms@oslab:/home/cis90/bin
*****
*           Fall 2014 CIS 90 Online Projects           *
*****
1) Aaron
2) Abraham
3) Alejandrino
4) Ann
5) Benji
6) Cameron
7) Chris
8) Cody
9) Deane
10) Duke
11) Francisco
12) Gabriel
13) Homer
14) James
15) Jeff
16) Jesus
17) Jimmy
18) Jonathan
19) Joshua
20) Justin
21) Luis
22) Matthew
23) Nadia
24) Navin
25) Nick
26) Nicole
27) Paul
28) Richard I.
29) Richard Z.
30) Roberto
31) Ronald
32) Ryan
33) Sam
34) Shea
35) Shenghong
36) Takashi
37) Tommy

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 <b>allscripts</b>
-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: <ul style="list-style-type: none"> <li>• Doesn't give full credit to the original author</li> <li>• Doesn't indicate where the code was obtained from</li> <li>• Doesn't include licensing terms</li> <li>• Violates copyright or licensing terms</li> </ul>

## Project Status

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

```
simben90@oslab:~/
/home/cis90/simben $ ls -l /home/cis90/*/bin/myscript
-rwxr-x---. 1 albjon90 cis90 1053 Nov 25 15:57 /home/cis90/albjon/bin/myscript
-rwxrwxrwx. 1 asngab90 cis90 517 Dec 1 09:34 /home/cis90/asngab/bin/myscript
-rwxr-x---. 1 ayalui90 cis90 546 Nov 25 14:30 /home/cis90/ayalui/bin/myscript
-rwxr-x---. 1 bincam90 cis90 1866 Nov 27 12:43 /home/cis90/bincam/bin/myscript
-rwxr-x---. 1 bownic90 cis90 546 Nov 25 14:30 /home/cis90/bownic/bin/myscript
-rwxr-x---. 1 boyjef90 cis90 3518 Dec 1 17:21 /home/cis90/boyjef/bin/myscript
-rwxr-x---. 1 dobtho90 cis90 546 Nov 25 15:07 /home/cis90/dobtho/bin/myscript
-rwxrwxr-x. 1 howmil90 cis90 2249 Nov 29 12:54 /home/cis90/howmil/bin/myscript
-rwxr-x---. 1 isoric90 cis90 704 Nov 18 14:27 /home/cis90/isoric/bin/myscript
-rwxrwxr-x. 1 keichr90 cis90 546 Nov 18 13:53 /home/cis90/keichr/bin/myscript
-rwx-----. 1 lamnav90 cis90 4757 Nov 30 23:41 /home/cis90/lamnav/bin/myscript
-rwxr-x---. 1 lishe90 cis90 4008 Nov 30 19:38 /home/cis90/lishe/bin/myscript
-rwxr-x---. 1 locaar90 cis90 896 Nov 25 17:02 /home/cis90/locaar/bin/myscript
-rwxrwxr-x. 1 milhom90 cis90 794 Nov 20 14:53 /home/cis90/milhom/bin/myscript
-rwxrwxr-x. 1 nieabr90 cis90 546 Nov 25 20:21 /home/cis90/nieabr/bin/myscript
-rwxr-x---. 1 nordak90 cis90 941 Nov 25 16:15 /home/cis90/nordak/bin/myscript
-rwxr-x---. 1 pikann90 cis90 1211 Dec 1 09:22 /home/cis90/pikann/bin/myscript
-rwxr-x---. 1 porrya90 cis90 1694 Nov 25 15:49 /home/cis90/porrya/bin/myscript
-rwxr-x---. 1 quifra90 cis90 1183 Nov 25 15:43 /home/cis90/quifra/bin/myscript
-rwxr-x---. 1 rodjus90 cis90 721 Nov 18 14:13 /home/cis90/rodjus/bin/myscript
-rwxrwxr-x. 1 simben90 cis90 10513 Dec 2 07:41 /home/cis90/simben/bin/myscript
-rwxr-x--x. 1 smimat90 cis90 546 Nov 24 10:12 /home/cis90/smimat/bin/myscript
-rwxr-x---. 1 tamjim90 cis90 1581 Nov 25 20:23 /home/cis90/tamjim/bin/myscript
-rwxr-x---. 1 tamtak90 cis90 2203 Dec 1 12:35 /home/cis90/tamtak/bin/myscript
-rwxrwx--x. 1 urijes90 cis90 519 Nov 24 18:01 /home/cis90/urijes/bin/myscript
-rwxr-x---. 1 wrenic90 cis90 710 Nov 25 15:07 /home/cis90/wrenic/bin/myscript
-rwxr-x---. 1 zemric90 cis90 982 Nov 25 16:14 /home/cis90/zemric/bin/myscript
/home/cis90/simben $
```

*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
```

```
simben90@oslab:~
/home/cis90/simben $ for i in `cat /etc/passwd | grep cis90 | cut -f6 -d":"`; do file=${i}/bin/myscri
pt; if [ -e "$file" ]; then echo -n $file exists ;if [ -r "$file" ]; then echo -n " and" is readabl
e; else echo -n " but" is not readable; fi; if [ -x "$file" ]; then echo " and" executable; else ec
ho 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 does not exist
/home/cis90/diljam/bin/myscript does not exist
/home/cis90/locaar/bin/myscript exists and is readable and executable
/home/cis90/tranad/bin/myscript does not exist
/home/cis90/nieabr/bin/myscript exists and is readable and executable
/home/cis90/smimat/bin/myscript exists and is readable and executable
/home/cis90/bowmic/bin/myscript exists and is readable and executable
/home/cis90/boyjef/bin/myscript exists and is readable and executable
/home/cis90/dobtho/bin/myscript exists and is readable and executable
/home/cis90/espale/bin/myscript does not exist
/home/cis90/pikann/bin/myscript exists and is readable and executable
/home/cis90/quifra/bin/myscript exists and is readable and executable
/home/cis90/rodjus/bin/myscript exists and is readable and executable
/home/cis90/tamjim/bin/myscript exists and is readable and executable
/home/cis90/tamtak/bin/myscript exists and is readable and executable
/home/cis90/urijes/bin/myscript exists and is readable and executable
/home/cis90/wrenic/bin/myscript exists and is readable and executable
/home/cis90/zahpau/bin/myscript does not exist
/home/cis90/zemric/bin/myscript exists and is readable and executable
/home/cis90/howmil/bin/myscript exists and is readable and executable
/home/cis90/albjon/bin/myscript exists and is readable and executable
/home/cis90/asngab/bin/myscript exists and is readable and executable
/home/cis90/atirob/bin/myscript does not exist
/home/cis90/ayalui/bin/myscript exists and is readable and executable
/home/cis90/bincam/bin/myscript exists and is readable and executable
/home/cis90/desmat/bin/myscript does not exist
/home/cis90/isoric/bin/myscript exists and is readable and executable
/home/cis90/keichr/bin/myscript exists and is readable and executable
/home/cis90/lamnav/bin/myscript exists but is not readablebut not executable
/home/cis90/lishe/bin/myscript exists and is readable and executable
/home/cis90/nordak/bin/myscript exists and is readable and executable
/home/cis90/porrya/bin/myscript exists and is readable and executable
/home/cis90/specod/bin/myscript does not exist
/home/cis90/tinsam/bin/myscript does not exist
/home/cis90/frocar/bin/myscript does not exist
/home/cis90/simben $
```

*a one line  
command  
using semi-  
colons!*

## Project Status

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

```
simben90@oslab:~
/home/cis90/simben $ find /home/cis90 -name myscript -exec wc -l {} \; 2> /dev/null | sort
245 /home/cis90/simben/bin/myscript
166 /home/cis90/simat/final-project/myscript
115 /home/cis90/boyjef/bin/myscript
112 /home/cis90/lishe/bin/myscript
105 /home/cis90/bincam/bin/myscript
87 /home/cis90/tamtak/bin/myscript
80 /home/cis90/porrya/bin/myscript
73 /home/cis90/howmil/bin/myscript
69 /home/cis90/tamjim/bin/myscript
57 /home/cis90/isoric/myscript
55 /home/cis90/pikann/bin/myscript
54 /home/cis90/quifra/bin/myscript
54 /home/cis90/quifra/bin/bin/myscript
54 /home/cis90/albjon/bin/myscript
49 /home/cis90/zemric/finalproject/myscript
49 /home/cis90/zemric/bin/myscript
47 /home/cis90/nordak/bin/myscript
47 /home/cis90/locaar/bin/myscript
42 /home/cis90/wrenic/bin/myscript
42 /home/cis90/rodjus/bin/myscript
42 /home/cis90/milhom/bin/myscript
42 /home/cis90/isoric/bin/myscript
37 /home/cis90/simat/bin/myscript
37 /home/cis90/quifra/myscript
37 /home/cis90/nieabr/bin/myscript
37 /home/cis90/keichr/bin/myscript
37 /home/cis90/dobtho/bin/myscript
37 /home/cis90/depot/myscript
37 /home/cis90/bownic/bin/myscript
37 /home/cis90/ayalui/bin/myscript
37 /home/cis90/asngab/bin/myscript
36 /home/cis90/urijes/bin/myscript
2 /home/cis90/bownic/myscript
0 /home/cis90/bincam/myscript
/home/cis90/simben $
```



# 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*

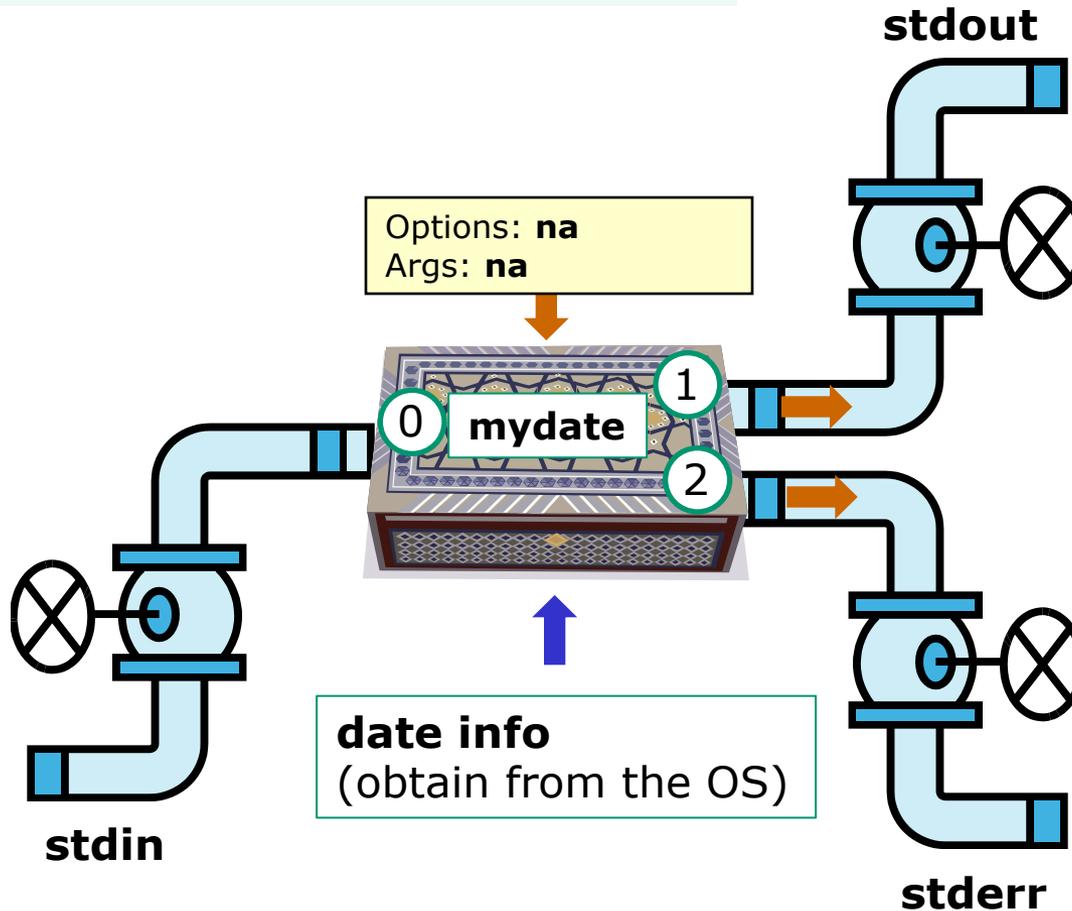
*Don't initialize  
them yet*

```
/home/cis90/simben $ mydate  
Hola simben90  
12/02/2014  
  
/home/cis90/simben $
```

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

# Running a Script

```
$ mydate
```



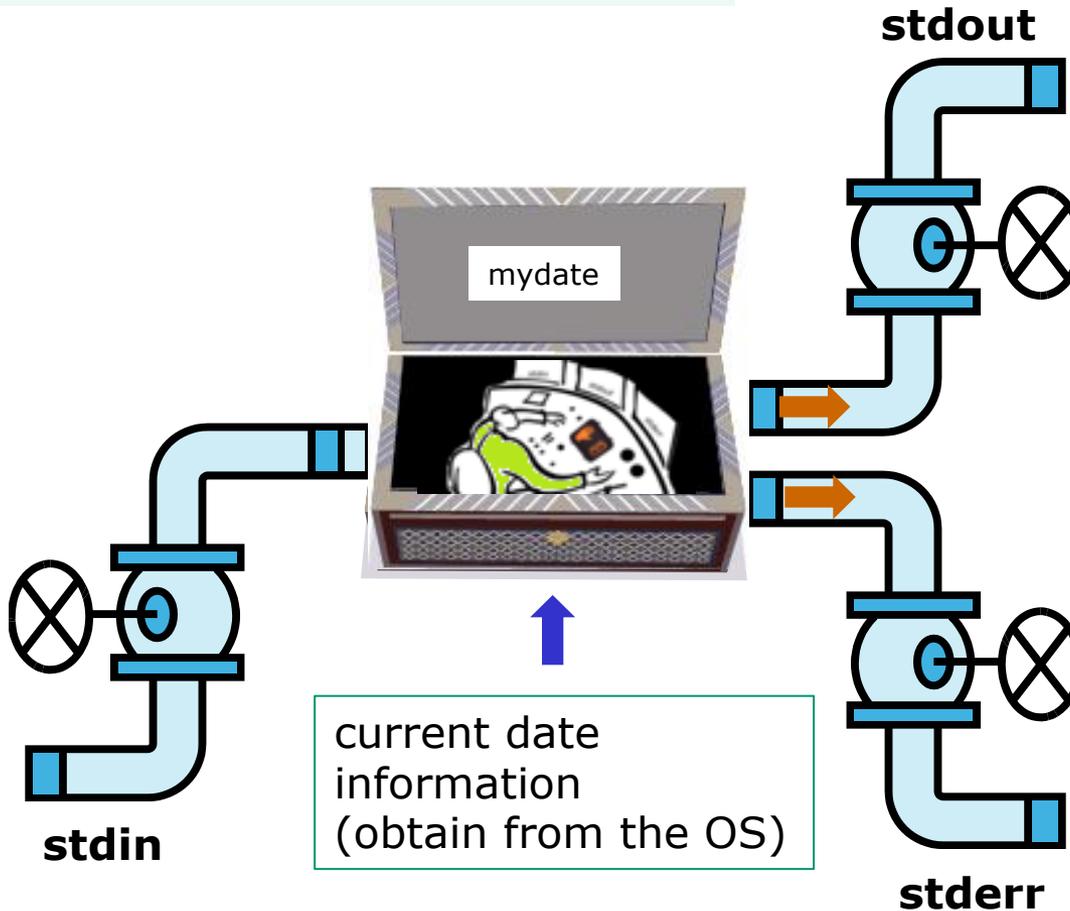
```
Hola simben90  
12/02/2014
```

*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
```

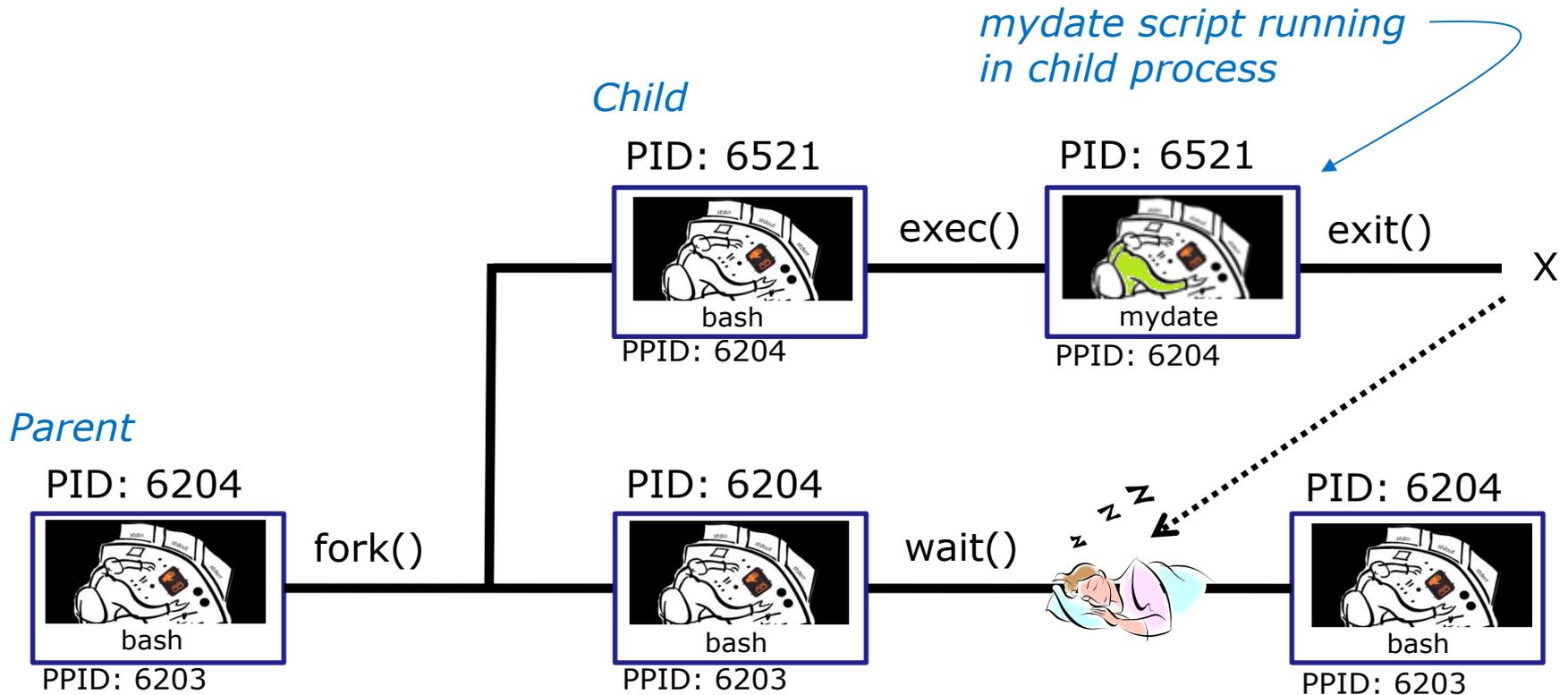


```
Hola simben90  
12/02/2014
```

*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  
12/02/2014
```

*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  
12/02/2014  
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  
12/02/2014  
Tic Tac
```

```
/home/cis90/simben $ export myvar3  
/home/cis90/simben $ mydate  
Hola simben90  
12/02/2014  
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
```

```
12/02/2014
```

```
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  
12/02/2014  
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

```
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
```

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

```
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/**

2) # Commands for Task 2
;;

3) # Commands for Task 3
;;

4) # Commands for Task 4

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/**
;;

40,17 38%
```

**line 40, column 17**

## 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
;;

62, 37 59%
```

*Use color syntax to spot unmatched quotes*

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

## Color Syntax

```

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

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

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

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

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

```

```

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*





# Scripting Tips

# sleep

(adding timed pauses)

## Class Exercise

Make a new script in your bin directory

```
cd bin  
vi example911
```

In vi add these lines then save:

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

Prepare and run your script

```
chmod +x example911  
example911
```

# Scripting Tips

`$(cmd)` and  
``cmd``

## Using \$(cmd)

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

For example:

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

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

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

*Using back tics around the command to evaluate*

## Using back tics

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

For example:

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

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

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

*Using `$( )` 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
```

< snipped >

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

*The 5<sup>th</sup> field of each row has the user's first and last name*

# myscript

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

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

Homer's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Examples - test file attributes
- 5) Examples - simple if statement
- 6) Examples - another if statement
- 7) Examples - logic
- 8) Examples - cut command to get name from /etc/passwd
- 9) Exit

Enter Your Choice: 8

**Wed Dec 3 14:00:53 PST 2008**

Hit the Enter key to return to menu

# myscript

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

*Let's add a friendly Hello using  
the user logname*

Homer's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Examples - test file attributes
- 5) Examples - simple if statement
- 6) Examples - another if statement
- 7) Examples - logic
- 8) Examples - cut command to get name from /etc/passwd
- 9) Exit

Enter Your Choice: 8

**Hello milhom90**

Wed Dec 3 14:07:07 PST 2008

Hit the Enter key to return to menu

# myscript

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

*Now include the  
/etc/passwd info  
as well*

## Homer's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Examples - test file attributes
- 5) Examples - simple if statement
- 6) Examples - another if statement
- 7) Examples - logic
- 8) Examples - cut command to get name from /etc/passwd
- 9) Exit

Enter Your Choice: 8

Hello milhom90

**milhom90:x:1156:103:Homer Miller:/home/cis90/milhom:/bin/bash**

Wed Dec 3 14:07:07 PST 2008

Hit the Enter key to return to menu

# myscript

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

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

Homer's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Examples - test file attributes
- 5) Examples - simple if statement
- 6) Examples - another if statement
- 7) Examples - logic
- 8) Examples - cut command to get name from /etc/passwd
- 9) Exit

Enter Your Choice: 8

Hello milhom90

**Homer Miller**

Wed Dec 3 14:07:07 PST 2008

Hit the Enter key to return to menu

# myscript

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

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

Homer's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Examples - test file attributes
- 5) Examples - simple if statement
- 6) Examples - another if statement
- 7) Examples - logic
- 8) Examples - cut command to get name from /etc/passwd
- 9) Exit

Enter Your Choice: 8

Hello milhom90

**Hello Homer Miller**

Wed Dec 3 14:07:07 PST 2008

Hit the Enter key to return to menu

# myscript

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

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

Homer's CIS 90 Final Project

- 1) Color
- 2) My Find Command
- 3) More practice
- 4) Examples - test file attributes
- 5) Examples - simple if statement
- 6) Examples - another if statement
- 7) Examples - logic
- 8) Examples - cut command to get name from /etc/passwd
- 9) Exit

Enter Your Choice: 8

**Hello Homer Miller**

Wed Dec 3 14:07:07 PST 2008

Hit the Enter key to return to menu

# myscript

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

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

Homer's CIS 90 Final Project

1) Color

2) My Find Command

3) More practice

4) Examples - test file attributes

5) Examples - simple if statement

6) Examples - another if statement

7) Examples - logic

8) Examples - cut command to get name from /etc/passwd

9) Exit

Enter Your Choice: 8

**Hello Homer**

Wed Dec 3 14:07:07 PST 2008

Hit the Enter key to return to menu

## Class Exercise

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

Make a new script in your `bin` directory

```
cd bin  
vi example401
```

In `vi` add these lines then save:

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

Prepare and run your script

```
chmod +x example401  
example401
```

Read your mail to view your new message

```
mail
```



# Scripting Tips

simple if  
statement

# myscript

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

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

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

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

      ;;
```

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

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

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

# myscript

Homer's CIS 90 Final Project

- 1) My favorite color
- 2) Getting started using grep command
- 3) Task 3
- 4) Task 4
- 5) Simple if statement
- 6) Task 6
- 7) Task 7
- 8) Getting your name
- 9) Exit

Enter Your Choice: **5**

Enter d or c: **d**

Tue Dec 2 09:22:39 PST 2014

Hit the Enter key to return to menu

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

*The **date** command runs  
because  $\$answer = d$*

# myscript

Homer's CIS 90 Final Project

- 1) My favorite color
- 2) Getting started using grep command
- 3) Task 3
- 4) Task 4
- 5) Simple if statement
- 6) Task 6
- 7) Task 7
- 8) Getting your name
- 9) Exit

Enter Your Choice: **5**

Enter d or c: **c**

```
December 2014
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

Hit the Enter key to return to menu

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

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

## Class Exercise

Run the previous example task

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

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

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



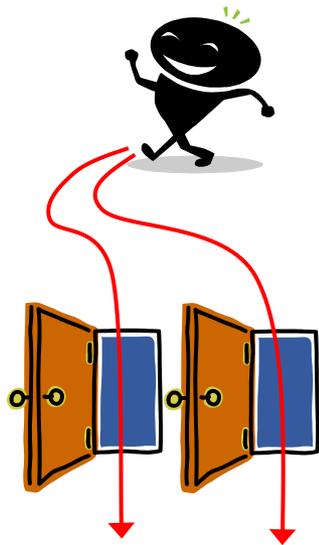
# Scripting Tips

## if statement with "or"

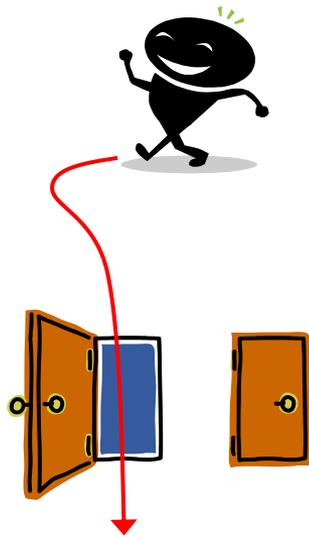


p	q	p or q
T	T	T
T	F	T
F	T	T
F	F	F

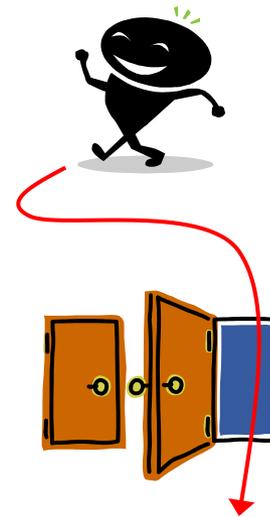
# OR logic



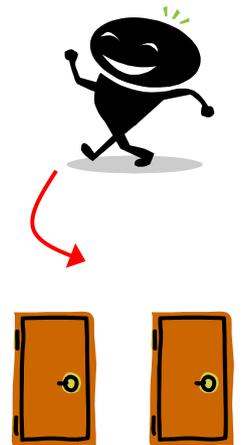
Yes



Yes



Yes



No

# myscript

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

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

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

;;
```

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

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

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

# myscript

Homer's CIS 90 Final Project

- 1) My favorite color
- 2) Getting started using grep command
- 3) Task 3
- 4) Task 4
- 5) Simple if statement
- 6) Another if statement
- 7) Task 7
- 8) Getting your name
- 9) Exit

Enter Your Choice: **6**

Enter d or c: **d**

Wed May 20 05:07:10 PDT 2009

Hit the Enter key to return to menu

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

***date** is run because user typed a "d"*

# myscript

Homer's CIS 90 Final Project

- 1) My favorite color
- 2) Getting started using grep command
- 3) Task 3
- 4) Task 4
- 5) Simple if statement
- 6) Another if statement
- 7) Task 7
- 8) Getting your name
- 9) Exit

Enter Your Choice: **6**

Enter d or c: **D**

Tue Dec 2 09:31:47 PST 2014

Hit the Enter key to return to menu

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

*date is run because user typed a "D"*

## Class Exercise

Make a new script in your bin directory

```
cd bin  
vi example654
```

In vi add these lines then save:

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

Prepare and run your script

```
chmod +x example654  
example654
```



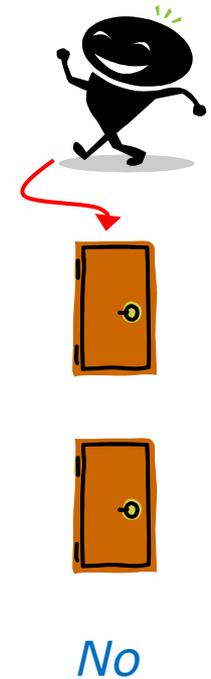
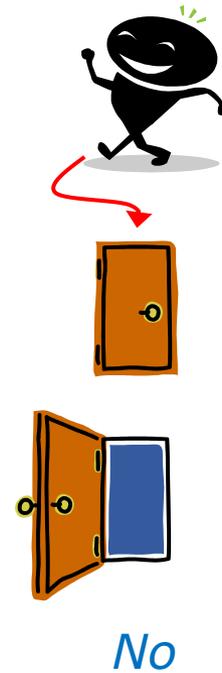
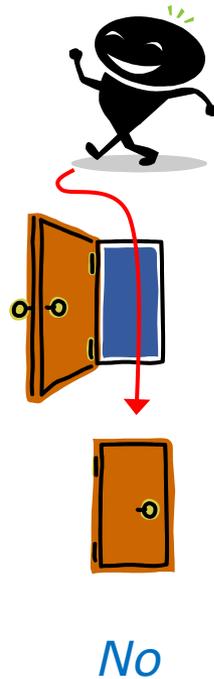
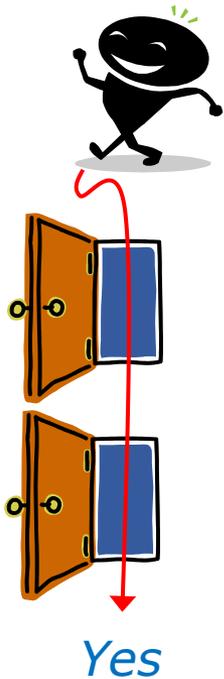
# Scripting Tips

## if statements with "and"



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

# AND logic



# myscript

```

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

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

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

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

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

```

**&&** means "and"

# myscript

Homer's CIS90 Final Project

- 1) My favorite color
- 2) Getting started using grep command
- 3) Task 3
- 4) Task 4
- 5) Simple if statement
- 6) Another if statement
- 7) Logic example
- 8) Getting your name
- 9) Exit

Enter Your Choice: **7**

Is the furnace on or off? **off**

Is there a fire in the fireplace (yes or no)? **no**

**It is really freezing in here**

Hit the Enter key to return to menu

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

# myscript

Homer's CIS90 Final Project

- 1) My favorite color
- 2) Getting started using grep command
- 3) Task 3
- 4) Task 4
- 5) Simple if statement
- 6) Another if statement
- 7) Logic example
- 8) Getting your name
- 9) Exit

Enter Your Choice: **7**

Is the furnace on or off? **on**

Is there a fire in the fireplace (yes or no)? **no**

**It is warm in here**

Hit the Enter key to return to menu

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

## Class Exercise

Run the previous example task

- run **allscripts**
- select Homer's script
- select Task **7** several times with different answers

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

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



# Scripting Tips

## if file types

# myscript

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

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

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

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

*tests to see  
if it's a  
directory*

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

# myscript

Homer's CIS 90 Final Project

- 1) My favorite color
- 2) Getting started using grep command
- 3) Task 3
- 4) More example IF statements
- 5) Simple if statement
- 6) Another if statement
- 7) Logic example
- 8) Getting your name
- 9) Exit

Enter Your Choice: **4**

The files in this directory are:

app

banner

enlightenment

< *snipped* >

Which file are you interested in? : **enlightenment**

Here are some details about enlightenment:

enlightenment: POSIX shell script text executable

enlightenment is a regular file

Here is long listing of the enlightenment file:

```
-rwxr-xr-x. 1 milhom90 cis90 3442 Aug  6 11:52 enlightenment
```

Hit the Enter key to return to menu

*a file*

# myscript

Homer's CIS 90 Final Project

- 1) My favorite color
- 2) Getting started using grep command
- 3) Task 3
- 4) More example IF statements
- 5) Simple if statement
- 6) Another if statement
- 7) Logic example
- 8) Getting your name
- 9) Exit

Enter Your Choice: **4**

The files in this directory are:

< *snipped* >

poems

< *snipped* >

Which file are you interested in? : poems

Here are some details about poems:

poems: directory

poems is a directory

Here is a long listing of the poems directory:

```
drwxr-xr-x. 8 milhom90 cis90 4096 Oct 28 15:48 poems
```

Hit the Enter key to return to menu

*a directory*



## Additional file attributes to test for:

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

## Class Exercise

Run the previous example task

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

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

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



# Scripting Tips

## if then else statement

# myscript

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

## Homer's CIS 90 Final Project

- 1) My favorite color
- 2) Getting started using grep command
- 3) An if-then-else statement
- 4) More example IF statements
- 5) Simple if statement
- 6) Another if statement
- 7) Logic example
- 8) Getting your name
- 9) Exit

Enter Your Choice: 3

Hello Homer Miller

**Wednesday**

**Wednesday, December 03, 2008**

Hit the Enter key to return to menu

*How can we do just  
one format or the  
other?*

# myscript

```

3)      # Commands for Task 3
        NAME=$(cat /etc/passwd | grep $LOGNAME | cut -f5 -d":" )
        echo "Hello $NAME"
        echo "$NAME, Do you like short or long dates?"
        echo -n "Enter 1 for short or 2 for long: "
        read ANSWER
        if [ "$ANSWER" = 1 ]; then
            date '+%A'
        else
            date '+%A, %B %d, %Y'
        fi
        ;;

```

*Prompt user for choice  
then use if-then-else  
statement*

```

        Enter Your Choice: 3
Hello Homer Miller
Homer Miller, Do you like short or long dates?
Enter 1 for short or 2 for long: 1
Tuesday
Hit the Enter key to return to menu

```

```

        Enter Your Choice: 3
Hello Homer Miller
Homer Miller, Do you like short or long dates?
Enter 1 for short or 2 for long: 2
Tuesday, December 02, 2014
Hit the Enter key to return to menu

```



# Scripting Tips

## 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

```

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

```
echo -e "\e[00;32m"
```

The image shows a terminal window titled 'simmsben@opus:~/bin'. The terminal output is as follows:

```

/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 $ █
  
```

```

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

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

*Curly braces are used to clearly 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