Lesson Module Status
- Slides – draft
- Properties - done
- Flash cards –
- First minute quiz – done
- Web calendar summary – done
- Web book pages – done
- Commands – done
- Lab – done
- Supplies () - na
- Class PC's – na
- Chocolates - bringing

- CCC Confer wall paper – done
- Materials uploaded – done
- Backup headset charged – nope
- Backup slides, CCC info, handouts on flash drive - done

- Check that room headset is charged – done

CIS 90 - Lesson 14

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

Quiz

No Quiz Today !

[ ] Has the phone bridge been added?

[ ] Is recording on?

[ ] Does the phone bridge have the mike?

[ ] Share slides, putty (rsimms, simmsben, roddyduk), Chrome and Eko VM

[ ] Disable spelling on PowerPoint

More Shell Scripting

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

✵ = hands on exercise for topic

Questions

Questions

Any questions on:

- Project?
- Extra credit Labs?
- Previous course material?

Housekeeping

Previous material and assignment

1. No labs due today

2. Project is due midnight on 5/26. That’s one week from now. If you haven't started yet, now would be a good time!

3. Extra credit labs are due midnight 6/2.

Fall 2011 Linux Courses

	Fall 2011	Waitlisted	CIS-191AB-73605 (73605) UNIX/Linux Inst, Confg, Admin	Main Campus	08/29/2011-12/17/2011 Lecture Tuesday 06:00PM - 08:05PM, Computer Information Labs, Room 2501 (more)...	J. Griffin	0 / 24 / 3	4.00	
	Fall 2011	Waitlisted	CIS-90-72345 (72345) Intro to UNIX/Linux	Main Campus	08/29/2011-12/17/2011 Lecture Wednesday 01:15PM - 04:20PM, Computer Information Labs, Room 2501 (more)...	J. Griffin	0 / 24 / 4	3.00	
	Fall 2011	Open	CIS-192A-73604 (73604) UNIX/Linux TCP/IP Admin	Main Campus	10/25/2011-12/13/2011 Lecture Tuesday 01:00PM - 05:10PM, Computer Information Labs, Room 2501 (more)...	R. Simms	11 / 24 / 0	2.00	

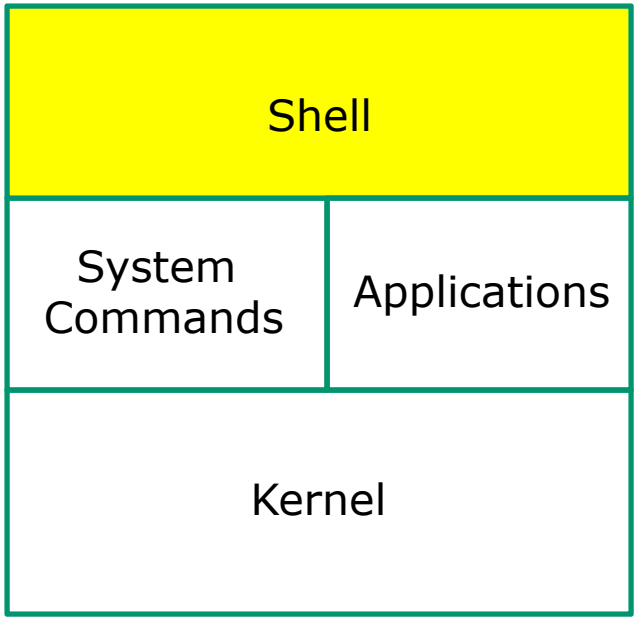
1. CIS 191 and CIS 192 will be offered on the same day, but the 192A is only for the second 8 weeks
2. CIS 191 is a hybrid class that will meet 2 hours a week (Tuesday evening) in the classroom and another two hour session from a lecture archive which the student can choose when to view. Labs are also part of the class and that time can be done remotely and scheduled to the student's convenience.
3. 3) CIS 192 will hold classes in the classroom and online simultaneously. Students may attend either way.

   If there are a few students who could not possibly make the CIS 191 two hours on campus, they can contact Jim for possible ways to make the class still work.

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/sysconfig/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

Class Exercise

Scripting

How many of the commands in /bin are really scripts?

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

How many of the commands in /usr/bin are really scripts?

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

Project

Getting Started

Getting started on the final project
(If you haven't done this already)

1. Create a file in your bin directory named myscript:
   - Copy from /home/cis90ol/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

Instructor reminder: run testscripts to see current status

```
simmsben@opus:~
/home/cis90ol/simmsben $ allscripts

        ************************************************************
        *               Spring 2011 CIS 90 Online Projects         *
        ************************************************************
        1) Bobby        7) Eric         13) Josh        19) Terry
        2) Chris        8) Gabriel      14) Marisol     20) Tommy
        3) Craig        9) Geoffrey     15) Merrick     21) Yu-Chen
        4) Dan M.       10) Jason       16) Quinton
        5) Daniel W.    11) Jeff        17) Tajvia
        6) Emanuel      12) Jesse       18) Tanner

        ************************************************************
        *               Examples and Hall of Fame                  *
        ************************************************************
        50) Duke        51) Benji       52) Junio

        99) Exit

        Enter Your Choice:
```

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

```
simmsben@opus:~

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

1) What is today?
2) The users on opus.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:
```

Final Project

What is allscripts and myscript?

```
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
        clear
        echo -n "
                Spring 2009 CIS 90 Projects
        1) Bilal
        2) Craig
        3) Dan
        4) Doug
        5) Duke
        6) Edgar D.
        7) Edgar O.
        8) Gabriel
        9) George
        10) Glen
        11) Jaime
        12) Janet
        13) Joe F.
        14) Joe P.
        15) Junious
        16) Kang
        17) Lieven
        18) Linda
        19) Michael
        20) Patrick
        21) Talley
        22) Todd
        23) William
        24) Benji
        99) Exit

        Enter Your Choice: "
        read RESPONSE
        case $RESPONSE in
          1)    # Bilal
                        /home/cis90/hussabil/bin/myscript
                        ;;
          2)    # Craig
                        /home/cis90/langlcra/bin/myscript
                        ;;
          3)    # Dan
                        /home/cis90/conydan/bin/myscript
                        ;;
          4)    # Doug
                        /home/cis90/kittldou/bin/myscript
                        ;;
          5)    # Duke
                        /home/cis90/roddyduk/bin/myscript
                        ;;
          6)    # Edgar D.
                        /home/cis90/delacedg/bin/myscript
                        ;;
          7)    # Edgar O.
                        /home/cis90/ortegedg/bin/myscript
                        ;;
          8)    # Gabriel
                        /home/cis90/pantogab/bin/myscript
                        ;;
          9)    # George
                        /home/cis90/balesgeo/bin/myscript
                        ;;
          10)   # Glen
                        /home/cis90/matligle/bin/myscript
                        ;;
          11)   # Jaime
                        /home/cis90/cervajai/bin/myscript
                        ;;
          12)   # Janet
                        /home/cis90/tumajan/bin/myscript
                        ;;
          13)   # Joe F.
                        /home/cis90/ferrajoe/bin/myscript
                        ;;
          14)   # Joe P.
                        /home/cis90/pragejoe/bin/myscript
                        ;;
          15)   # Junious
                        /home/cis90/rossjun/bin/myscript
                        ;;
          16)   # Kang
                        /home/cis90/leekan/bin/myscript
                        ;;
          17)   # Lieven
                        /home/cis90/mambulie/bin/myscript
                        ;;
          18)   # Linda
                        /home/cis90/donohlin/bin/myscript
                        ;;
          19)   # Michael
                        /home/cis90/georgmic/bin/myscript
                        ;;
          20)   # Patrick
                        /home/cis90/caseypat/bin/myscript
                        ;;
          21)   # Talley
                        /home/cis90/senantal/bin/myscript
                        ;;
          22)   # Todd
                        /home/cis90/krametod/bin/myscript
                        ;;
          23)   # William
                        /home/cis90/tumawil/bin/myscript
                        ;;
          24)   # Benji
                        /home/cis90/simmsben/bin/myscript
                        ;;
          99)   exit 0
                        ;;
          *)    echo "Please enter a number between 1 and 6"
                        ;;
        esac
        echo -n "Hit the Enter key to return to menu "
        read dummy
done
```

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

allscripts

```
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
  clear
  echo -n "
        Spring 2000 CIS 90 Projects
  1) Bilal
  2) Craig
  3) Dan
  4) Doug
  5) Duke
  6) Edgar G.
  7) Edgar O.
  8) Gabriel
  9) George
  10) Glen
  40) Songul
  12) Janet
  13) Joe F.
  14) Joe P.
  15) Junious
  16) Kang
  17) Lieven
  18) Linda
  19) Michael
  20) Patrick
  21) Talley
  22) Todd
  23) William
  24) Benji
  00) Exit

  Enter Your Choice: "
  read RESPONSE
  case $RESPONSE in
   1)   # Bilal
            /home/cis90/hussabil/bin/myscript
            ;;
   2)   # Craig
            /home/cis90/langlcra/bin/myscript
            ;;
   3)   # Dan
            /home/cis90/conydan/bin/myscript
            ;;
   4)   # Doug
            /home/cis90/kittldou/bin/myscript
            ;;
   5)   # Duke
            /home/cis90/roddyduk/bin/myscript
            ;;
   6)   # Edgar G.
            /home/cis90/delacedg/bin/myscript
            ;;
   7)   # Edgar O.
            /home/cis90/ortegedg/bin/myscript
            ;;
   8)   # Gabriel
            /home/cis90/pantogab/bin/myscript
            ;;
   9)   # George
            /home/cis90/balengeo/bin/myscript
            ;;
   10)  # Glen
            /home/cis90/matligle/bin/myscript
            ;;
   40)    # Songul
            /home/cis90/messison/bin/myscript
            ;;
   12)  # Janet
            /home/cis90/tummjan/bin/myscript
            ;;
   13)  # Joe F.
            /home/cis90/ferrajoe/bin/myscript
            ;;
   14)  # Joe P.
            /home/cis90/pragejoe/bin/myscript
            ;;
   15)  # Junious
            /home/cis90/rossjun/bin/myscript
            ;;
   16)  # Kang
            /home/cis90/leekan/bin/myscript
            ;;
   17)  # Lieven
            /home/cis90/mambulie/bin/myscript
            ;;
   18)  # Linda
            /home/cis90/donohlin/bin/myscript
            ;;
   19)  # Michael
            /home/cis90/georgmic/bin/myscript
            ;;
   20)  # Patrick
            /home/cis90/caneypat/bin/myscript
            ;;
   21)  # Talley
            /home/cis90/senantal/bin/myscript
            ;;
   22)  # Todd
            /home/cis90/krametod/bin/myscript
            ;;
   23)  # William
            /home/cis90/tumawil/bin/myscript
            ;;
   24)  # Benji
            /home/cis90/simmsben/bin/myscript
            ;;
   99)    exit 0
            ;;
   *)   echo "Please enter a number between 1 and 6"
            ;;
  esac
  echo -n "Hit the Enter key to return to menu "
  read dummy
done
```

The while statement in allscripts will loop through the code forever

A case statement is used to run the appropriate myscript file in the student's bin directory

/home/cis90/messison/bin/myscript

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

myscript

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

The outer while statement that loops forever

myscript

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

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

myscript

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

This is a case statement. One case for each task.

myscript

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

read commands get input from the user

Grading rubric (60 points maximum)

Possible Points	Requirements
30	Implementing all five tasks (6 points each): • Requirements for each task: – Minimum of 10 script command lines – Has comments to explain what it does – 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 a GID or group (5 points) • Use of a UID or user (5 points) • Use of a signal (5 points) • Use of piping (5 points) • Use of an environment variable (5 points) • Use of a comment (5 points) • Use of /bin/mail (5 points) • Use of a conditional (5 points) The maximum for this section are 25 points.
5	Present your script in front of the class
Points lost	
-15	Fails to run from /home/cis90/bin/allscripts
-15	The other users in the cis90 group are unable to read and execute your script.
-5	For each error message displayed
Extra credit	
30	Up to three additional tasks (10 points each)

This is how the final project will be graded

Project

Scripting a task

What takes longer?

Writing the script?

Or deciding what to script?

One way to get started … select a random command to highlight with a custom script

Commands

.	echo	lpstat	sort
at	env	ls	spell
banner	exit	mail	su
bash	export	man	tail
bc	file	mesg	tee
cal	find	mkdir	touch
cancel	finger	more	type
cat	grep	mv	umask
cd	head	passwd	uname
chgrp	history	ps	unset
chmod	id	pwd	vi
chown	jobs	rm	wc
clear	kill	rmdir	who
cp	ln	set	write
date	lp/lpr	sleep	xxd

This example will focus on the **grep** command. Research your command by reading the man page

rsimms@opus:~/cis90/project

```
GREP(1)                                                          GREP(1)

NAME
       grep, egrep, fgrep - print lines matching a pattern

SYNOPSIS
       grep [options] PATTERN [FILE...]
       grep [options] [-e PATTERN | -f FILE] [FILE...]

DESCRIPTION
       Grep  searches  the  named  input FILEs (or standard input if no files are
       named, or the file name - is given) for lines containing a  match  to  the
       given PATTERN.  By default, grep prints the matching lines.

       In addition, two variant programs egrep and fgrep are available.  Egrep is
       the same as grep -E.  Fgrep is the same as grep -F.

OPTIONS
       -A NUM, --after-context=NUM
              Print NUM lines of trailing context after matching lines.  Places a
              line containing -- between contiguous groups of matches.

       -a, --text
              Process a binary file as if it were text; this is equivalent to the
              --binary-files=text option.

       -B NUM, --before-context=NUM
:
```

*Using **man grep** to show the man page for grep*

Research and select some options you want to try out … for this example we will use:

-R, -r, --recursive
Read all files under each directory, recursively; this is equivalent to the -d recurse option.

-w, --word-regexp
Select only those lines containing matches that form whole words. The test is that the matching substring must either be at the beginning of the line, or preceded by a non-word constituent character. Similarly, it must be either at the end of the line or followed by a non-word constituent character. Word-constituent characters are letters, digits, and the underscore.

Next, decide what you want to do with the command you selected. For this example we will:

1. Find the word love in the user's home directory

2. Count how many times it was found

3. Find the word love in the system /home directory (this directory holds all user home directories)

4. Count how many times it was found

Start coding!

Customize the menu options for Task 1

simmsben@opus:~

```
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
clear
echo -n "
CIS 90 Final Project
1) Task 1
2) Task 2
3) Task 3
4) Task 4
5) Task 5
6) Exit
Enter Your Choice: "
read RESPONSE
case $RESPONSE in
1) # Commands for Task 1
;;
2) # Commands for Task 2
;;
3) # Commands for Task 3
;;
4) # Commands for Task 4
"/home/cis90/depot/myscript"
```

simmsben@opus:~

```
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
clear
echo -n "
CIS 90 Final Project
1) Looking for love in all the wrong places
2) Task 2
3) Task 3
4) Task 4
5) Task 5
6) Exit
Enter Your Choice: "
read RESPONSE
case $RESPONSE in
1) # Commands for Task 1
;;
2) # Commands for Task 2
;;
3) # Commands for Task 3
;;
4) # Commands for Task 4
-- INSERT -- W10: Warning: Changing a readonly file          10,44          Top
```

simmsben@opus:~

```
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
clear
echo -n "
CIS 90 Final Project
1) Looking for love in all the wrong places
2) Task 2
3) Task 3
4) Task 4
5) Task 5
6) Exit
Enter Your Choice: "
read RESPONSE
case $RESPONSE in
1) # Commands for Task 1
;;
2) # Commands for Task 2
;;
3) # Commands for Task 3
;;
4) # Commands for Task 4
-- INSERT -- W10: Warning: Changing a readonly file          10,44          Top
```

Add your script commands for Task 1 here

Example code to highlight grep command

Task 1 case statement starts here

Task 1 script commands were added here

Task 1 case statement ends here

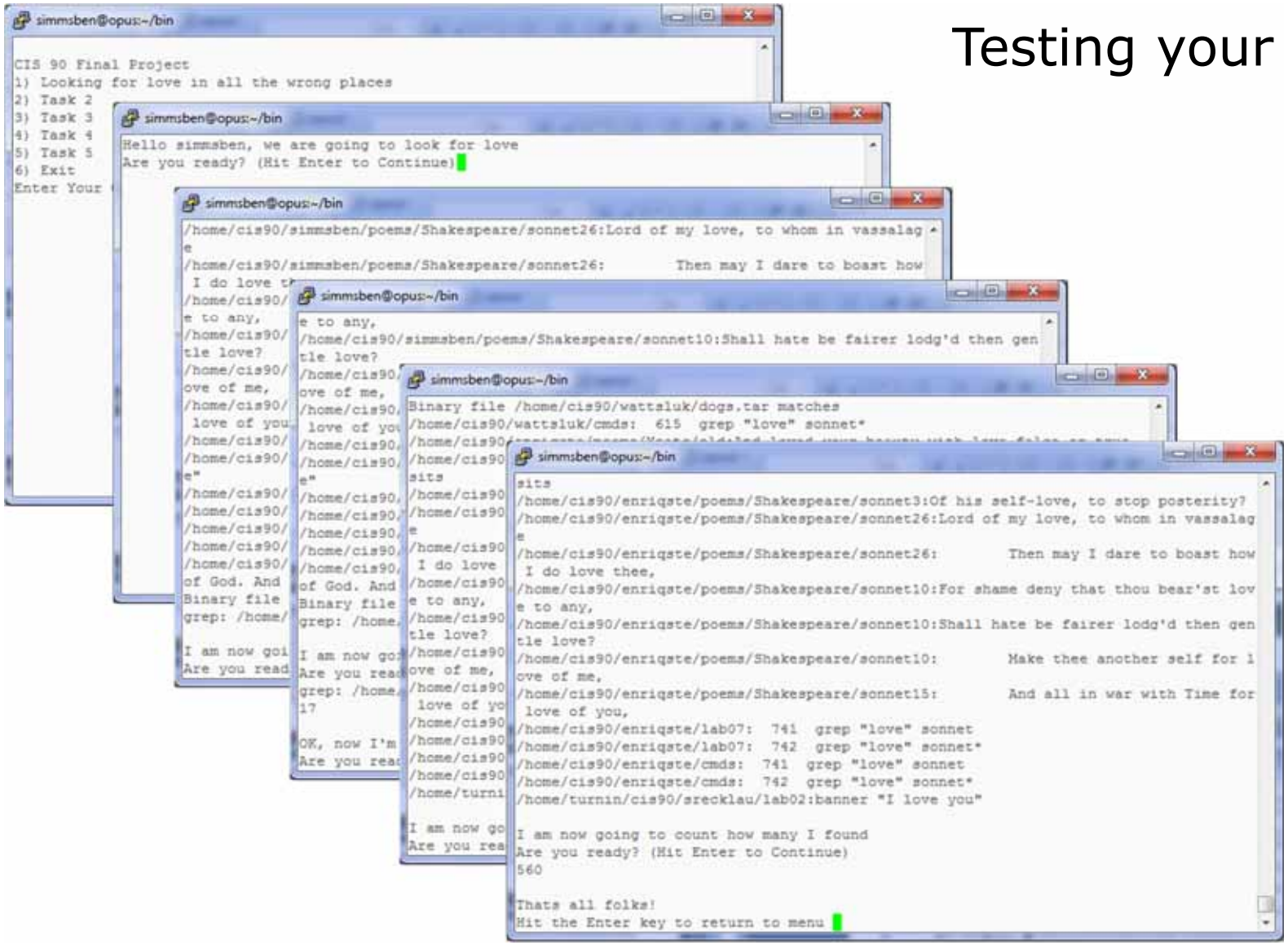
simmsben@opus:~/bin

```
read RESPONSE
case $RESPONSE in
1) # Task 1 - Looking for Love, using grep (with recursive and full word options)
clear
echo "Hello $LOGNAME, we are going to look for love"
echo -n "Are you ready? (Hit Enter to Continue)"
read answer
echo "OK, trying your home directory first ..."
grep -rw love /home/cis90/$LOGNAME
echo; echo "I am now going to count how many I found"
echo -n "Are you ready? (Hit Enter to Continue)"
read answer
grep -rw love /home/cis90/$LOGNAME | wc -l
echo; echo "OK, now I'm going to try the entire /home directory ..."
echo -n "Are you ready? (Hit Enter to Continue)"
read answer
grep -rw love /home 2> /dev/null
echo; echo "I am now going to count how many I found"
echo -n "Are you ready? (Hit Enter to Continue)"
read answer
grep -rw love /home 2> /dev/null | wc -l
echo; echo "Thats all folks!"
;;
2) # Commands for Task 2
;;
-- INSERT --                                          20,6          53%
```

simmsben@opus:~/bin

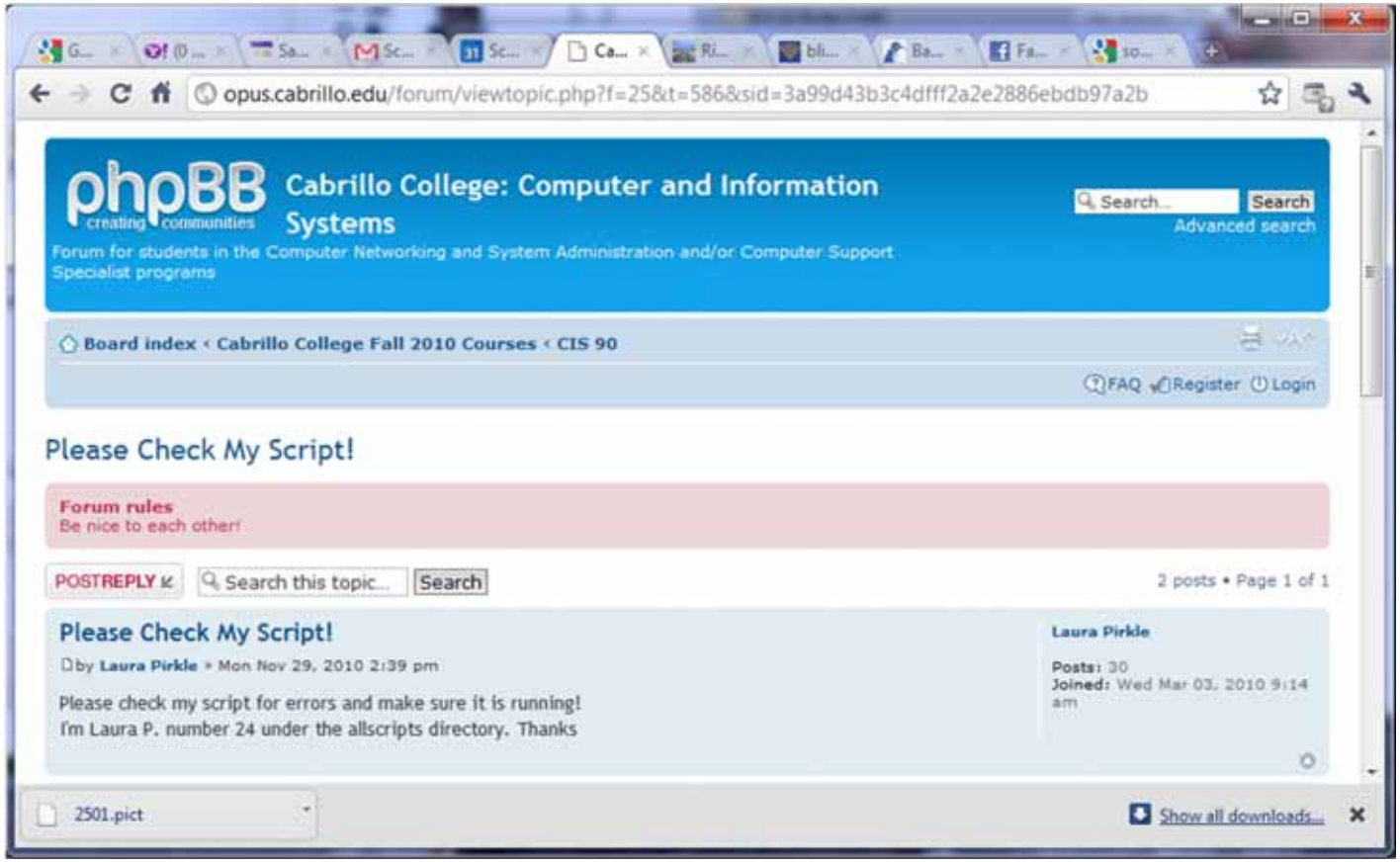
```
read RESPONSE
case $RESPONSE in
1) # Task 1 - Looking for Love, using grep (with recursive and full word options)
clear
echo "Hello $LOGNAME, we are going to look for love"
echo -n "Are you ready? (Hit Enter to Continue)"
read answer
echo "OK, trying your home directory first ..."
grep -rw love /home/cis90/$LOGNAME
echo; echo "I am now going to count how many I found"
echo -n "Are you ready? (Hit Enter to Continue)"
read answer
grep -rw love /home/cis90/$LOGNAME | wc -l
echo; echo "OK, now I'm going to try the entire /home directory ..."
echo -n "Are you ready? (Hit Enter to Continue)"
read answer
grep -rw love /home 2> /dev/null
echo; echo "I am now going to count how many I found"
echo -n "Are you ready? (Hit Enter to Continue)"
read answer
grep -rw love /home 2> /dev/null | wc -l
echo; echo "Thats all folks!"
;;
2) # Commands for Task 2
;;
-- INSERT --                                    20,6          53%
```

Testing your script



Test your script and make changes till its just the way you want it

Testing your script



The ask others on the forum to check your script and give you feedback

simmsben@opus:~/bin

```
read RESPONSE
case $RESPONSE in
1) # Task 1 - Looking for Love, using grep (with recursive and full word options)
clear
echo "Hello $LOGNAME, we are going to look for love"
echo -n "Are you ready? (Hit Enter to Continue)"
read answer
echo "OK, trying your home directory first ..."
grep -rw love /home/cis90/$LOGNAME
echo; echo "I am now going to count how many I found"
echo -n "Are you ready? (Hit Enter to Continue)"
read answer
grep -rw love /home/cis90/$LOGNAME | wc -l
echo; echo "OK, now I'm going to try the entire /home directory ..."
echo -n "Are you ready? (Hit Enter to Continue)"
read answer
grep -rw love /home 2> /dev/null
echo; echo "I am now going to count how many I found"
echo -n "Are you ready? (Hit Enter to Continue)"
read answer
grep -rw love /home 2> /dev/null | wc -l
echo; echo "Thats all folks!"
;;
2) # Commands for Task 2
;;
-- INSERT --                                          20,6          53%
```

Finally, check your script against the grading rubric to see how many points you have earned

- Requirements for each task:
  - ✓ Minimum of 10 script command lines
  - ✓ Has comments to explain what it does
  - ✓ Has user interaction

simmsben@opus:~/bin

```
read RESPONSE
case $RESPONSE in
1) # Task 1 - Looking for Love, using grep (with recursive and full word options)
clear
echo "Hello $LOGNAME, we are going to look for love"
echo -n "Are you ready? (Hit Enter to Continue)"
read answer
echo "OK, trying your home directory first ..."
grep -rw love /home/cis90/$LOGNAME
echo; echo "I am now going to count how many I found"
echo -n "Are you ready? (Hit Enter to Continue)"
read answer
grep -rw love /home/cis90/$LOGNAME | wc -l
echo; echo "OK, now I'm going to try the entire /home direc
echo -n "Are you ready? (Hit Enter to Continue)"
read answer
grep -rw love /home 2> /dev/null
echo; echo "I am now going to count how many I found"
echo -n "Are you ready? (Hit Enter to Continue)"
read answer
grep -rw love /home 2> /dev/null | wc -l
echo; echo "Thats all folks!"
;;
2) # Commands for Task 2
;;
-- INSERT --
```

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 a GID or group (5 points)
- Use of a UID or user (5 points)
- Use of a signal (5 points)
- Use of piping (5 points) ✓
- Use of an environment variable (5 points) ✓
- Use of a comment (5 points)
- Use of /bin/mail (5 points)
- Use of a conditional (5 points)

The maximum for this section are 25 points.

Finally, check your script against the grading rubric to see how many points you have earned

Plan extra time for:

- Figuring our how to do what you really want to do!
- Removing sytax errors
- Removing logic errors
- Posting script code on the forum and asking others to view it and suggest how to fix it

Don't wait till the last minute to start your project!

Scripting Tips
vi

Global search and replace with vi

:%s /oldstring/newstring/g

```
rsimms@opus:/home/cis192/depot
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xht
ml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>Arwen's CIS 192 Lab 10</title>
</head>
<body>
<h1>Arwen's CIS 192 Lab 10</h1>
<h2>Internet Services</h2>
<div>
<img src="hwy50.jpg" alt="Highway 50" />
</div>

<p>Spring 2009</p>

<div>
<a href="http://validator.w3.org/check/referer"
style="background-color: transparent">
<img style="border-style:none" width="88" height="31"
src="http://www.w3.org/Icons/valid-xhtml10" alt="Valid
  
<a href="http://jigsaw.w3.org/css-validator/check/refer
style="background-color: transparent">
<img style="border-style:none" width="88" height="31"
src="http://jigsaw.w3.org/css-validator/images/vcss" al
</div>

</body>
</html>
~
:%s /Arwen/Elrond/g
```

```
rsimms@opus:/home/cis192/depot
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xht
ml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>Elrond's CIS 192 Lab 10</title>
</head>
<body>
<h1>Elrond's CIS 192 Lab 10</h1>
<h2>Internet Services</h2>
<div>
<img src="hwy50.jpg" alt="Highway 50" />
</div>

<p>Spring 2009</p>

<div>
<a href="http://validator.w3.org/check/referer"
style="background-color: transparent">
<img style="border-style:none" width="88" height="31"
src="http://www.w3.org/Icons/valid-xhtml10" alt="Valid XHTML 1.0 Strict" /></a>
  
<a href="http://jigsaw.w3.org/css-validator/check/referer"
style="background-color: transparent">
<img style="border-style:none" width="88" height="31"
src="http://jigsaw.w3.org/css-validator/images/vcss" alt="Valid CSS!" /></a>
</div>

</body>
</html>
~
:%s /Arwen/Elrond/g                                    7,1          All
```

Scripting Tips

$(cmd) or `cmd`

Shell Scripts

We can save the **output of a command to a file**
- Use > and >> to send command output to a file
  e.g. **echo "Almost There" > myfile**

We can send the **output of a command to another command**
- Use | to send output from one command to another
  e.g. **spell poems/Shakespeare/* | wc -l**

Note, we have used $ to mean "the value of"
- **echo $PATH**
- **echo $LOGNAME**

We can use $(command) to use the **output of a command**:
- **to assign to a variable**,
  e.g. **students=$(ls /home/cis90)**
- **as an argument to another command**,
  e.g. **banner $(date)**

$(command)

```
/home/cis90/roddyduk $ ls /home/cis90
answers   cervajai  donohlin  hussabil  langlcra  ortegedg  rossjun   tumawil
balesgeo  conydan   ferrajoe  keevejos  leekan    pantogab  senantal  weilmat
bin       delacedg  georgmic  kittldou  mambulie  pragejoe  simmsben  woodsben
caseypat  depot     guest     krametod  matligle  roddyduk  tumajan

/home/cis90/roddyduk $ echo $(ls /home/cis90)
answers balesgeo bin caseypat cervajai conydan delacedg depot donohlin ferrajoe
georgmic guest hussabil keevejos kittldou krametod langlcra leekan mambulie
matligle ortegedg pantogab pragejoe roddyduk rossjun senantal simmsben tumajan
tumawil weilmat woodsben

/home/cis90/roddyduk $ students=$(ls /home/cis90)

/home/cis90/roddyduk $ echo $students
answers balesgeo bin caseypat cervajai conydan delacedg depot donohlin ferrajoe
georgmic guest hussabil keevejos kittldou krametod langlcra leekan mambulie
matligle ortegedg pantogab pragejoe roddyduk rossjun senantal simmsben tumajan
tumawil weilmat woodsben

/home/cis90/roddyduk $ echo Have a great day | mail -s "SPAM" $students
```

$(command) can be used wherever you need to assign the output of a command to a variable

$(command)

```
/home/cis90/roddyduk $ ls /home/cis90
answers   cervajai  donohlin  hussabil  langlcra  ortegedg  rossjun   tumawil
balesgeo  conydan   ferrajoe  keevejos  leekan    pantogab  senantal  weilmat
bin       delacedg  georgmic  kittldou  mambulie  pragejoe  simmsben  woodsben
caseypat  depot     guest     krametod  matligle  roddyduk  tumajan

/home/cis90/roddyduk $ echo `ls /home/cis90`
answers balesgeo bin caseypat cervajai conydan delacedg depot donohlin ferrajoe
georgmic guest hussabil keevejos kittldou krametod langlcra leekan mambulie
matligle ortegedg pantogab pragejoe roddyduk rossjun senantal simmsben tumajan
tumawil weilmat woodsben

/home/cis90/roddyduk $ students=`ls /home/cis90`

/home/cis90/roddyduk $ echo $students
answers balesgeo bin caseypat cervajai conydan delacedg depot donohlin ferrajoe
georgmic guest hussabil keevejos kittldou krametod langlcra leekan mambulie
matligle ortegedg pantogab pragejoe roddyduk rossjun senantal simmsben tumajan
tumawil weilmat woodsben

/home/cis90/roddyduk $ echo Have a great day | mail -s "SPAM" $students
```

$(command) and `command` (using back ticks) are equivalent

Class Exercise

Scripting

Try these commands:

```
echo ls
echo $(ls)
banner ls
banner $(ls)

banner `date`
banner $(date)

mybinfiles=$(ls ~/bin)
echo $mybinfiles
```

Scripting Tips

getting a field from a table

/etc/passwd

```
[rsimms@opus ~]$ cat /etc/passwd
< snipped >
tumajan:x:1020:103:Janet Tuma:/home/cis90/tumajan:/bin/bash
tumawil:x:1021:103:William Tuma:/home/cis90/tumawil:/bin/bash
kittldou:x:1022:103:Douglas Kittle:/home/cis90/kittldou:/bin/bash
hussabil:x:1023:103:Bilal Hussain:/home/cis90/hussabil:/bin/bash
senantal:x:1024:103:Talley Senanayake:/home/cis90/senantal:/bin/bash
ferrajoe:x:1025:103:Joe Ferrante:/home/cis90/ferrajoe:/bin/bash
balesgeo:x:1026:103:George Bales:/home/cis90/balesgeo:/bin/bash
matligle:x:1029:103:Glen Matlin:/home/cis90/matligle:/bin/bash
krametod:x:1030:103:Todd Kramer:/home/cis90/krametod:/bin/bash
ortegedg:x:1031:103:Edgar Ortega:/home/cis90/ortegedg:/bin/bash
pantogab:x:1032:103:Gabriel Pantoja:/home/cis90/pantogab:/bin/bash
cervajai:x:1033:103:Jaime Cervantes:/home/cis90/cervajai:/bin/bash
donohlin:x:1034:103:Linda Donohue:/home/cis90/donohlin:/bin/bash
mambulie:x:1035:103:Lieven Mambulu:/home/cis90/mambulie:/bin/bash
caseypat:x:1036:103:Patrick Casey:/home/cis90/caseypat:/bin/bash
pragejoe:x:1037:103:Joe Prager:/home/cis90/pragejoe:/bin/bash
rossjun:x:1038:103:Junious Ross:/home/cis90/rossjun:/bin/bash
conydan:x:1039:103:Dan Cony:/home/cis90/conydan:/bin/bash
georgmic:x:1040:103:Michael George:/home/cis90/georgmic:/bin/bash
langlcra:x:1041:103:Craig Langlo:/home/cis90/langlcra:/bin/bash
leekan:x:1042:103:Kang Lee:/home/cis90/leekan:/bin/bash
```

*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
                ;;
```

Lets start with something simple

```
                Duke'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
        ;;
```

Lets add a Hello with the users's logname

```
                Duke'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 roddyduk
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 show the info for the user in /etc/passwd

```
                 Duke'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 roddyduk
roddyduk:x:1156:103:Duke Roddy:/home/cis90/roddyduk:/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
              ;;
```

*Now cut out the 5th field from the /etc/passwd row using the **cut** command. The **-d** option specifies the delimiter to use.*

```
                Duke'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 roddyduk
Duke Roddy
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

```
                Duke'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 roddyduk
Hello Duke Roddy
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

```
                Duke'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 Duke Roddy
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":" | cut -f1 -d" ")
        echo "Hello $NAME"
        date
        ;;
```

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

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

        Enter Your Choice: 8
Hello Duke
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 to your example401 script 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

      ;;
```

Style 1

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

Style 2

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

*Coding styles will vary. Some folks like the **then** on the same line as the **if** followed by a block of indented commands . Some will put **then** on the next line.*

myscript

```
                Duke'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

```
                Duke'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 50) Duke
- select Task 5 and enter d (for date)
- select Task 5 and enter c (for calendar)

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

- **vi /home/cis90/roddyduk/bin/myscript**

Scripting Tips

if statement with "or"

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

```
                Duke'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

```
                Duke'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
```

date is run because user typed a D

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

Class Exercise

Make a new script in your bin directory

```
cd bin
vi example654
```

In vi add these lines to your script 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"

myscript

```
6)    # 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 smokey 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

```
 Duke'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? 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

```
                Duke'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 50) Duke
- select Task 7 several times with different answers

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

- **vi /home/cis90/roddyduk/bin/myscript**

Scripting Tips

if

file types

myscript

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

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

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

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

tests to see if it's a regular file

tests to see if it's a directory

myscript

```
                  Duke's CIS 90 Final Project
        1) Color
        2) My Find Command
        3) More practice
        4) Task 4
        5) Task 5
        6) Exit

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

myscript

```
 Duke's CIS 90 Final Project
        1) Color
        2) My Find Command
        3) More practice
        4) Task 4
        5) Task 5
        6) 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 roddyduk cis90 4096 Apr  8 10:01 Anon
Hit the Enter key to return to menu
```

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

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

Class Exercise

Run the previous example task

- run **allscripts**
- select 50) Duke
- select Task 4

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

- **vi /home/cis90/roddyduk/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'
        ;;
```

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

        Enter Your Choice: 3
Hello Duke Roddy
Wednesday
Wednesday, December 03, 2008
Hit the Enter key to return to menu
```

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
        ;;
```

```
Enter Your Choice: 3
Hello Duke Roddy
Duke Roddy, 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 Duke Roddy
Duke Roddy, 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.roddyduk myscript.v1 newscript old program quiet quiet.bak
script treed tryme typescript zoom

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

[rsimms@opus bin]$ echo $3
banner

[rsimms@opus bin]$ echo $7
I

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

[rsimms@opus bin]$ echo $#
20

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

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 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
```

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.

Class Exercise

Make a new script in your bin directory

```
cd bin
vi example777
```

In vi add these lines to your script then save:

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

Prepare and run your script

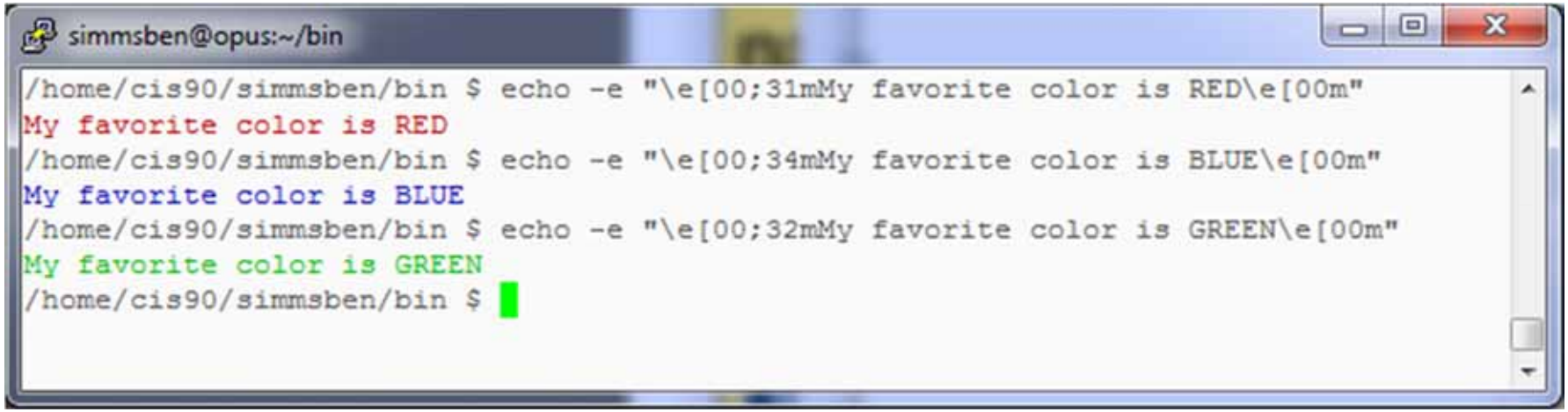
```
chmod +x example777
example777
```

Scripting Tips

color

Using Color

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



Make sure to use echo -e to enable interpretation of backslash escapes

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

Using Color

```
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 $
```

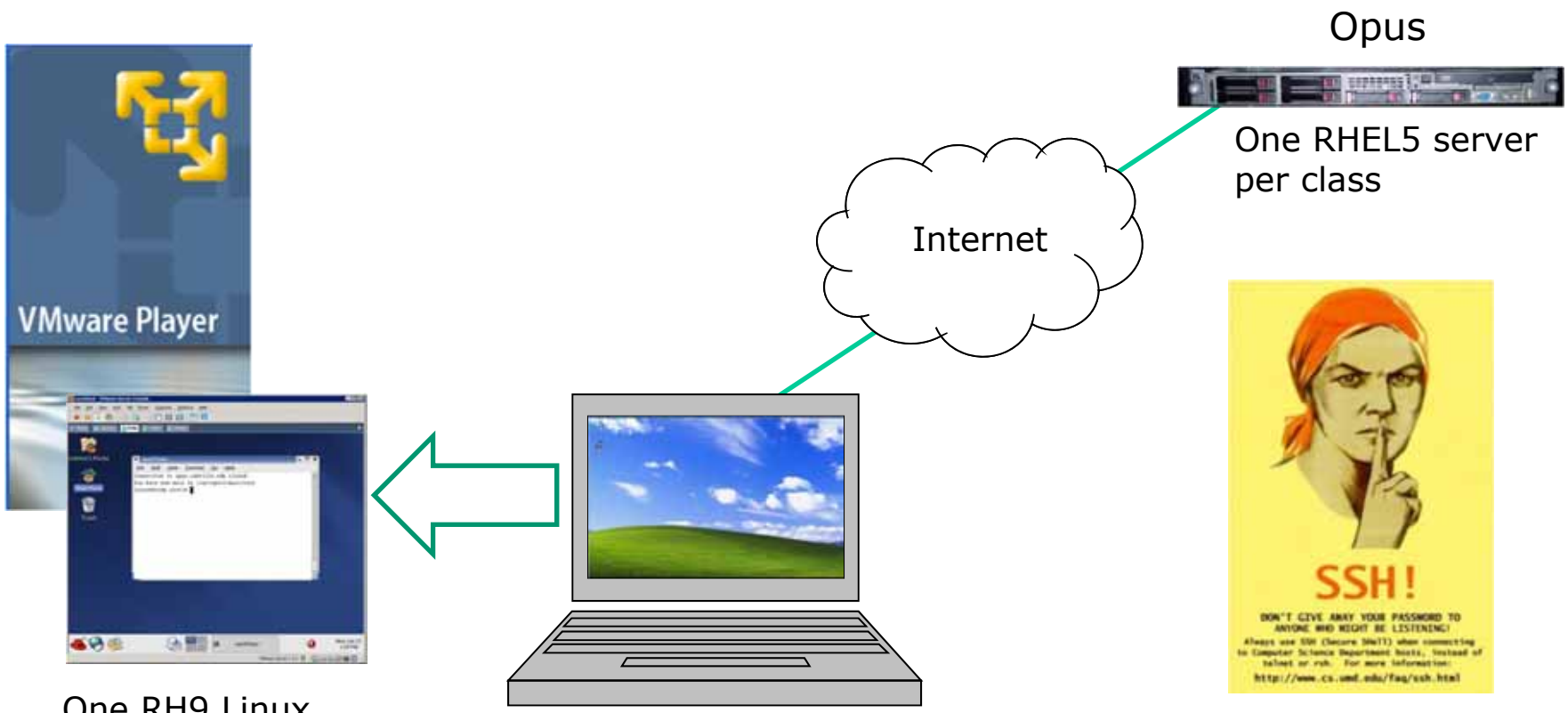
Use echo -e '\e[00m' to revert to normal

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

scp

Copying your files on Opus to Linux system at home

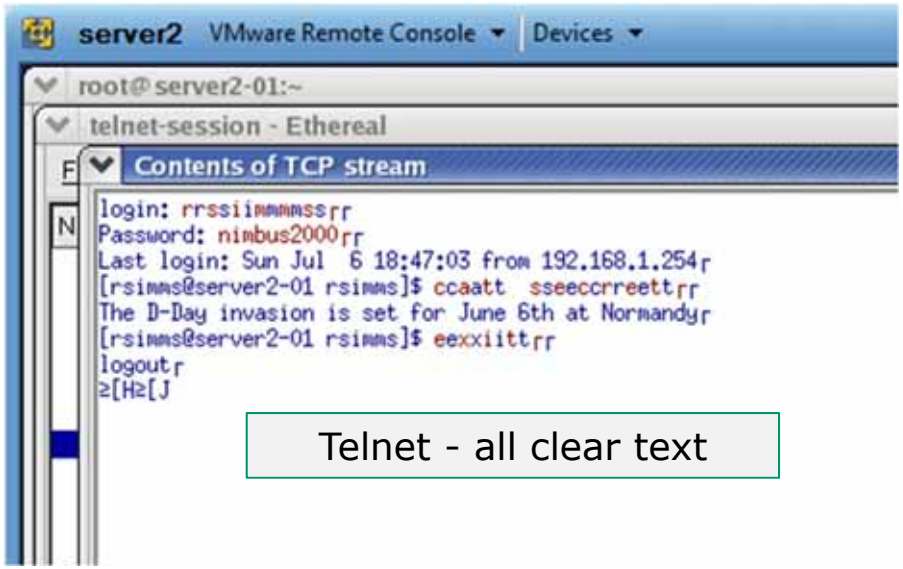
Classroom PC's, VMs and Remote Server



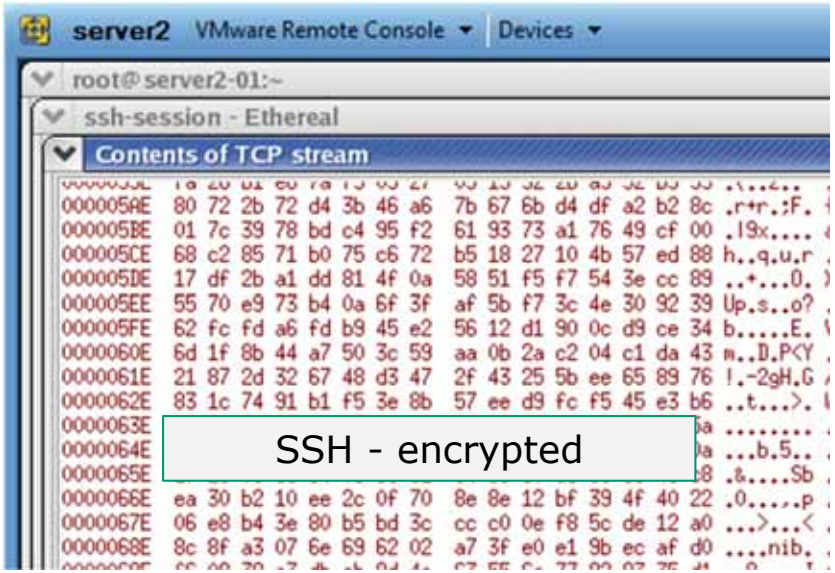
SSH is a network protocol that enables secure connections between computers

Telnet and SSH (Secure Shell)

Opus



Sniffer view of a Telnet session



Sniffer view of a SSH session

Local computer

ssh protocol

Secure Shell Protocol

- Allows secure (encrypted connections between computers)
  - ssh command – secure login and terminal sessions
  - scp command – secure file copies between computers

scp

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

- cp
  - copies files on the same computer
  - examples:

```
cp myscript myscript.v1
cp myscript.v1 backups/
cp /home/cis90/simmsben/bin/myscript benscript
```

- scp
  - copies files between computers:
  - examples:

```
scp roddyduk@opus.cabrillo.edu:myscript .
scp lab45 roddyduk@opus.cabrillo.edu:lab45
scp lab45 roddyduk@opus.cabrillo.edu:
```

scp

command	1st argument (from)	2nd argument (to)
scp	roddyduk@opus.cabrillo.edu:bin/myscript	.
scp	lab45	roddyduk@opus.cabrillo.edu:lab45
scp	lab45	roddyduk@opus.cabrillo.edu:

scp

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

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

scp

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

Path to file. Relative to user's home directory.

Either the IP address or hostname of the remote computer. Need to connect to system on the Internet

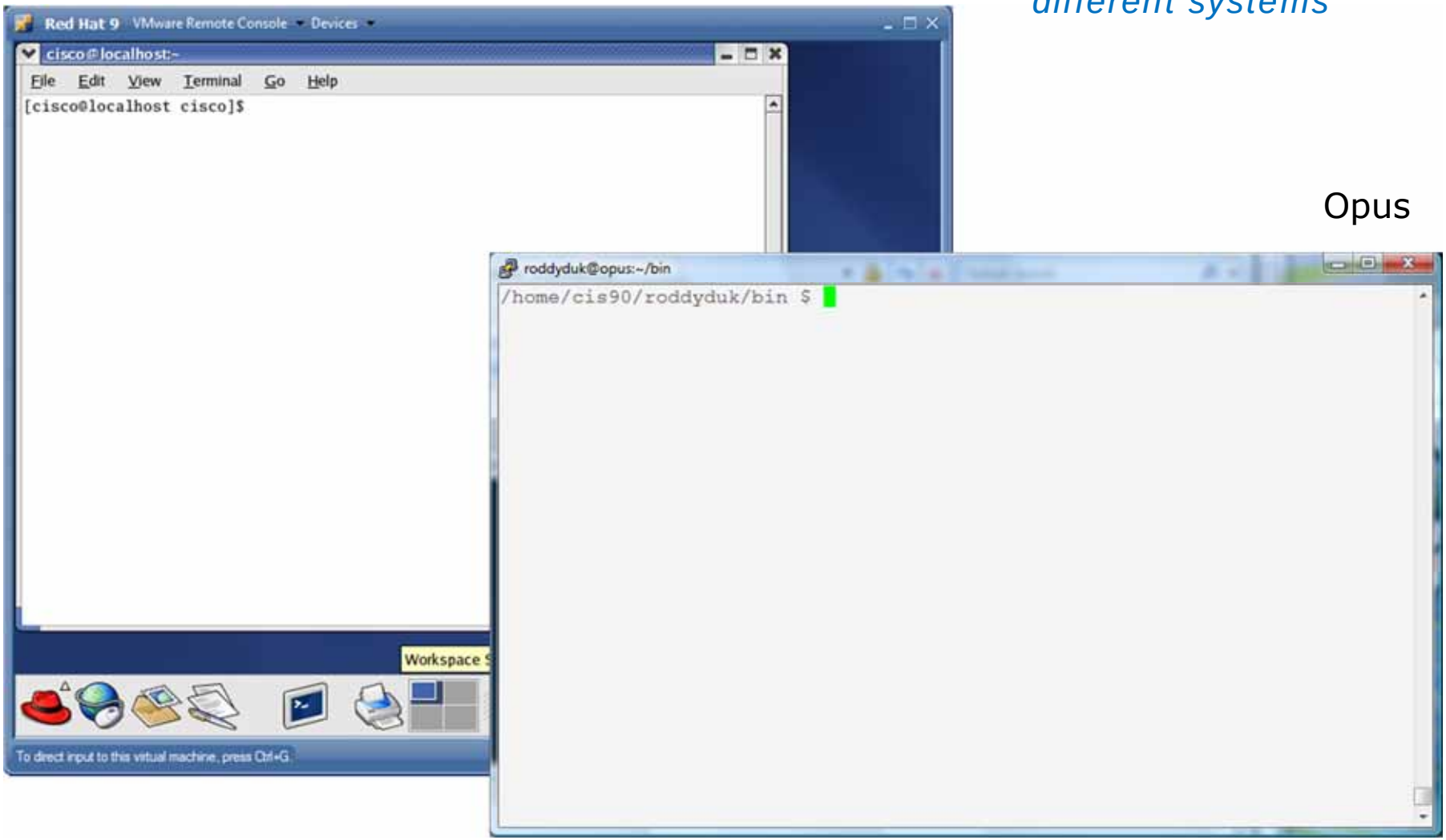
The username on the remote computer. Needed to login and this defines the home directory for the relative path to the file

scp

Local Linux System

Logged in to two different systems

Opus



scp

Local Linux System

Performing scp copy from Opus to local Red hat system

Opus

```
[cisco@localhost cisco]$ man scp
[cisco@localhost cisco]$ scp roddyduk@opus.cabrillo.edu:bin/myscript .
roddyduk@opus.cabrillo.edu's password:
myscript                  100% |****************************|  1410      00:00
[cisco@localhost cisco]$
```

```
/home/cis90/roddyduk/bin $ ls
app     datecal        hi    I         myscript.bak  treed  zoom
banner  enlightenment  home  myscript  treat5        tryme
/home/cis90/roddyduk/bin $
```

scp

Local Linux System

Catting files on both systems to verify the copy

Opus

```
[cisco@localhost cisco]$ man scp
[cisco@localhost cisco]$ scp roddyduk@opus.cabrillo.edu:bin/myscript .
roddyduk@opus.cabrillo.edu's password:
myscript                  100% |****************************|  1410      00:00
[cisco@localhost cisco]$ cat myscript
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
        clear
        echo -n "
                Duke's CIS 90 Final Projec
        1) Getting started
        2) My Find Command
        3) Task 3
        4) Task 4
        5) Task 5
        6) Exit

        Enter Your Choice: "
        read RESPONSE
        case $RESPONSE in
```

```
/home/cis90/roddyduk/bin $ cat myscript
#!/bin/bash
#
# menu: A simple menu template
#
while true
do
        clear
        echo -n "
                Duke's CIS 90 Final Project
        1) Getting started
        2) My Find Command
        3) Task 3
        4) Task 4
        5) Task 5
        6) Exit

        Enter Your Choice: "
        read RESPONSE
        case $RESPONSE in
          1)    # Getting started
                echo -n "What is your name? "
                read NAME
                echo -n "$NAME, are you excited for the upcoming new year? "
                echo " a) Yes i can't wait! 2009! "
```

```
[cisco@localhost bin]$ scp roddyduk@opus.cabrillo.edu:bin/* .
roddyduk@opus.cabrillo.edu's password:
app                   100% |*****************************|   220    00:00
scp: bin/backups: not a regular file
banner                100% |*****************************|  6160    00:00
benscript             100% |*****************************| 10433    00:00
datecal               100% |*****************************|   509    00:00
enlightenment         100% |*****************************|  3388    00:00
hi                    100% |*****************************|   107    00:00
home                  100% |*****************************|   104    00:00
I                     100% |*****************************|   375    00:00
myscript              100% |*****************************|  1652    00:00
myscript.bak          100% |*****************************|  1148    00:00
myscript.v1           100% |*****************************|  1410    00:00
treat5                100% |*****************************|   795    00:00
treed                 100% |*****************************|   190    00:00
tryme                 100% |*****************************|   174    00:00
zoom                  100% |*****************************|    74    00:00
[cisco@localhost bin]$ ls
app      banner     datecal        hi    I         myscript.bak  treat5  tryme
backups  benscript  enlightenment  home  myscript  myscript.v1   treed   zoom
[cisco@localhost bin]$
```

*To copy multiple files, use the * expansion character*

Note, this will copy files, but not directories

To recursively copy files and directories use the –r option

Red Hat 9 VMware Remote Console Devices

cisco@localhost:~/bin

File Edit View Terminal Go Help

```
[cisco@localhost bin]$ scp -r roddyduk@opus.cabrillo.edu:bin/* .
roddyduk@opus.cabrillo.edu's password:
app                   100% |*****************************|   220    00:00
myscript              100% |*****************************|  1410    00:00
banner                100% |*****************************|  6160    00:00
benscript             100% |*****************************| 10433    00:00
datecal               100% |*****************************|   509    00:00
enlightenment         100% |*****************************|  3388    00:00
hi                    100% |*****************************|   107    00:00
home                  100% |*****************************|   104    00:00
I                     100% |*****************************|   375    00:00
myscript              100% |*****************************|  1652    00:00
myscript.bak          100% |*****************************|  1148    00:00
myscript.v1           100% |*****************************|  1410    00:00
treat5                100% |*****************************|   795    00:00
treed                 100% |*****************************|   190    00:00
tryme                 100% |*****************************|   174    00:00
zoom                  100% |*****************************|    74    00:00
[cisco@localhost bin]$ ls
app      banner     datecal        hi    I         myscript.bak  treat5  tryme
backups  benscript  enlightenment  home  myscript  myscript.v1   treed   zoom
[cisco@localhost bin]$
```

Now all files and directories will be copied from your Opus home directory to your Linux computer at home

tar

tar command

tar *options tarfile files*

To simplify file transfers, Windows users typically "zip" multiple files together into a singe "zipfile".

Linux users use the **tar** command to do this and "archive" multiple files into a single "tarball".

tar command

pathname to the starting directory which will recursively include all files in any directories below

tar cvf *tarfile files*

will create a tarball named *tarfile* containing all the files specified

Example:

Creates a tarball named mytarball that archives all the files in your home directory (that you have permission to access and excluding the tarfile itself)

```
/home/cis90ol/simmsben $ cd
/home/cis90ol/simmsben $ tar cvf mytarball .
./
./olddir/
./island/
< snipped >
./sayid
./lab04.graded
tar: Error exit delayed from previous errors
/home/cis90ol/simmsben $
```

This means some files were not added to the archive

tar command

pathname to the starting directory which will recursively include all files in any directories below

tar tvf *tarfile [files]*

will view a tarball's "table of contents"

Example:

Views some files

```
/home/cis90ol/simmsben $ tar tvf mytarball ./poems/Yeats
drwxr-xr-x simmsben/cis90ol  0 2011-04-14 14:20:08 ./poems/Yeats/
-r--r--r-- simmsben/cis90ol 863 2001-07-20 15:04:39 ./poems/Yeats/whitebirds
-r--r--r-- simmsben/cis90ol 856 2011-02-17 10:15:18 ./poems/Yeats/mooncat
-r--r--r-- simmsben/cis90ol 520 2001-07-20 15:04:39 ./poems/Yeats/old
/home/cis90ol/simmsben $
/home/cis90ol/simmsben $ tar tvf mytarball
drwxr-xr-x simmsben/cis90ol  0 2011-05-19 09:26:39 ./
drwxrwxr-x simmsben/cis90ol  0 2011-04-14 14:20:08 ./olddir/
< snipped >
-rw-r--r-- simmsben/cis90ol      0 2011-04-19 11:03:27 ./sayid
-r-------- simmsben/staff      512 2011-03-11 14:18:29 ./lab04.graded
/home/cis90ol/simmsben $
```

Views all files

tar command

options
(no – needed)

archive file

files to backup

```
      c
tar   tvf    tarfile    file(s)
      x
```

***c**reate*
***t**able of contents (view)*
*e**x**tract*

v = verbose, double v (vv) provides more information

Note: The full path to each file is stored in the archive and these paths are used when restoring files

Class Exercise

tar

On Opus, tar up you entire home directory. Use your own logname to name the tarball:

```
cd
tar cvf logname.tar .
```

be sure and start in your home directory

On your local vm, create a directory with the same name as your Opus logname and change into it.

```
mkdir logname
cd logname/
scp logname@opus.cabrillo.edu:*.tar .
tar xvf logname.tar
```

Wrap up

Commands:

scp	- secure copy command
tar	- archive command
if then else	- conditionals in scripts
[ ]	- for logic tests in scripts

Next Class

Project is due next week!

Backup

Class Exercise

- Start up the local VM named Frodo and login as cis90
- Copy the files banner and myscript from your bin directory on Opus

```
scp logname@opus.cabrillo.edu:bin/myscript .
scp logname@opus.cabrillo.edu:bin/banner .
```

- Create a bin directory on your local VM (if needed)

```
mkdir bin
```

- Move banner and myscript to your new bin directory

```
mv banner myscript bin/
```

- Check permissions to make sure myscript and banner have execute permissions

```
cd bin
ls -l
```

- Run the scripts you copied from Opus:

```
./myscript
./banner "Hello"
```

Class Exercise

scp

- Copy all the files and directories in your Opus bin directory to your local bin directory

```
scp –r logname@opus.cabrillo.edu:bin/* .
```

- Now see if you can copy your entire poems directory on Opus to a new poems directory in your local cis90 home directory.