Cabrillo College

Lab 2: Using Commands

The purpose of this lab is to explore command usage with the shell and miscellaneous UNIX commands.

Preparation

Everything you need to do this lab can be found in the Lesson 2 materials on the CIS 90 website: http://simms-teach.com/cis90calendar.php. Review carefully all Lesson 2 slides, even those that may not have been covered in class.

Check the forum at: http://oslab.cishawks.net/forum/ for any tips and updates related to this lab. The forum is also a good place to ask questions if you get stuck or help others.

If you would like some additional assistance come to the CIS Lab on campus where you can get help from instructors and student lab assistants: http://webhawks.org/~cislab/.

Procedure

This lab must be done on Opus to get credit

Please log into the Opus server using your personal account. You will need to use the following commands in the steps below.

banner	**clear**	**finger**	**passwd**	**whatis**
bash	**date**	**history**	**ps**	**who**
bc	**echo**	**id**	**type**	
cal	**exit**	**man**	**uname**	

For grading purposes your command history along with your answers to three questions will be submitted at the end of the lab. Your command history will be scanned to verify each step below was completed.

The Shell

1. What shell are you currently using? What command did you use to determine this? (Hint: We did this in Lab 1)

2. The **type** command takes the name of a command as an argument. It can be used to show where that command resides on the path. Enter the following commands and notice where the commands such as **man**, **uname,** etc. are located on the path.

   type man
   type uname
   type tryme
   type echo
   type type
   type bogus

   Can the **type** command take multiple arguments? Try:

   type man uname type

3. Use the **echo** command to show the value of several shell variables.

   echo $HOME
   echo $TERM
   echo $LOGNAME
   echo $PS1
   echo $SHELL
   echo $PATH

   Can you specify more than one variable as an argument? Try it.

   echo $TERM $HOME $LOGNAME

   Use the **echo** command again and notice why the **$** metacharacter is important.

   echo $LOGNAME
   echo LOGNAME

   What happens with a variable that does not exist? Try:

   echo $BOGUS

   Now try supplying both text and variables as arguments to the echo command:

   echo I am $LOGNAME and I like the $SHELL shell

4. Use the following to display your terminal type and compare it to your terminal device:

   echo $TERM
   tty

   Note that your terminal type ($TERM) and terminal device (output from **tty**) are two different things.

Set the TERM environment variable to "dumb", and execute the **clear** command. What happens?

```
TERM="dumb"
clear
```

Use **echo $TERM** to see the new setting. Set TERM to "vt100" or "ansi" What happens now with the clear command?

```
echo $TERM
TERM="ansi"
clear
```

Set the TERM environment variable back to "xterm" which is what it was when you logged in.

```
TERM=”xterm”
```

5. What happens when you enter the following commands? Why?

```
DATE
Date
date
```

6. What results do you get from the command: **who -g**
   What program outputs this message?

7. Enter each command below (exactly) and observe the results. How many arguments does each of the following command lines have?

```
echo one    two       threefour
echo "My TERM type is" $TERM
echo one.two.three
```

8. What is the difference in output between the following two commands? Note, the $ and > are part of the prompt, you don’t need to type them.

```
$ echo red 'white
> and blue'
```

and

```
$ echo red white \
> and blue
```

Note: the *Enter* key is pressed immediately after the last character of each line.

9. Use the shell metacharacter "**;**" to write out a one line command that will clear the screen, print out the date and the current month's calendar.
   $ _____

Commands

10. If you have not already done so, use the **passwd** command to change your password. Name three things you should never do with your password:

    1._____

    2._____

    3._____

11. Use a single **uname** command with the necessary options to display ONLY the *network node hostname,* the *kernel release* number and the *operating system.* Your command should produce the following output exactly:

    ```
    oslab.cishawks.net 2.6.32-220.23.1.el6.i686 GNU/Linux
    ```

    Hint: Use the **man uname** command, scroll up/down and use **q** to quit.

12. What is the difference in output between the following two commands?

    banner I am fine
    banner "I am fine"

13. guest90 is another user account on Opus. Use the **finger** command to find out what guest90's plan is. (Hint: Use guest90 as an argument to the **finger** command.)

    guest90's plan:

14. What is the UID (User ID) number for your own account? (Hint: we did this in Lab 1)

Using online documentation

15. Issue a **man bc** command. Scroll up and down then use **q** to quit.

16. What is the **whatis** command?
    Use the command with the argument: **bc**
    How does this compare to using the **man** command with **-f** option?
    man -f bc

17. Is **tryme** a UNIX command? Use the commands you know to find out.

18. Use the manual pages and the **who** command to output a count of the number of users logged on.

19. Run the command: **man -k boot**
    Use the manual pages to find out what the **-k** option does.
    What command is **man -k** equivalent to? Run the equivalent command and verify.

20. Run the command: **info bash**
    See if you can explore the hot links (marked with a *). Use the up and down arrows to select a link. Use Enter key to follow a link. Use L to go back to last page. Use **q** to quit.

21. Now use your web browser (outside of Opus) and google "linux bc command". If you find any interesting sites you can post them on the forum.

22. Here's a challenging task: Use the **man** command to discover how you can use the **bc** command to obtain the square root of 361. The **bc** command is an example of an interactive command, because you must enter the numbers to calculate from the keyboard while the program is running.

Submit your work

Now that you have finished this lab on Opus, you may submit your work using the following two commands:

history -a
submit

When the command asks you which assignment to submit, respond with **2** followed by the *Enter* key. Then answer the three questions that it asks of you.

You can submit as many times as you wish up to the deadline. Only your last submittal will be graded. You can use the verify command to check what will be graded.

Grading Rubric

27 points	For entering all the commands on Opus asked for in each step of the lab. The instructor will scan the commands in your history file and take off a point for any missing commands.
3 points	For correct answers to the three questions asked by the submit script (1 point each)

Remember, late work is not accepted. If you can't finish the lab before the deadline then submit what you have completed before the deadline for partial credit.

Appendix

Questions asked by the submit script:

1. Name a UNIX command that gets its input only from the command line?
2. Name an interactive command that reads its input from the keyboard?
3. Name a UNIX command that gets its input from the Operating System?