***Lab Week 3 – Chapter 4***

**1. Open a Terminal from the Activities Button**

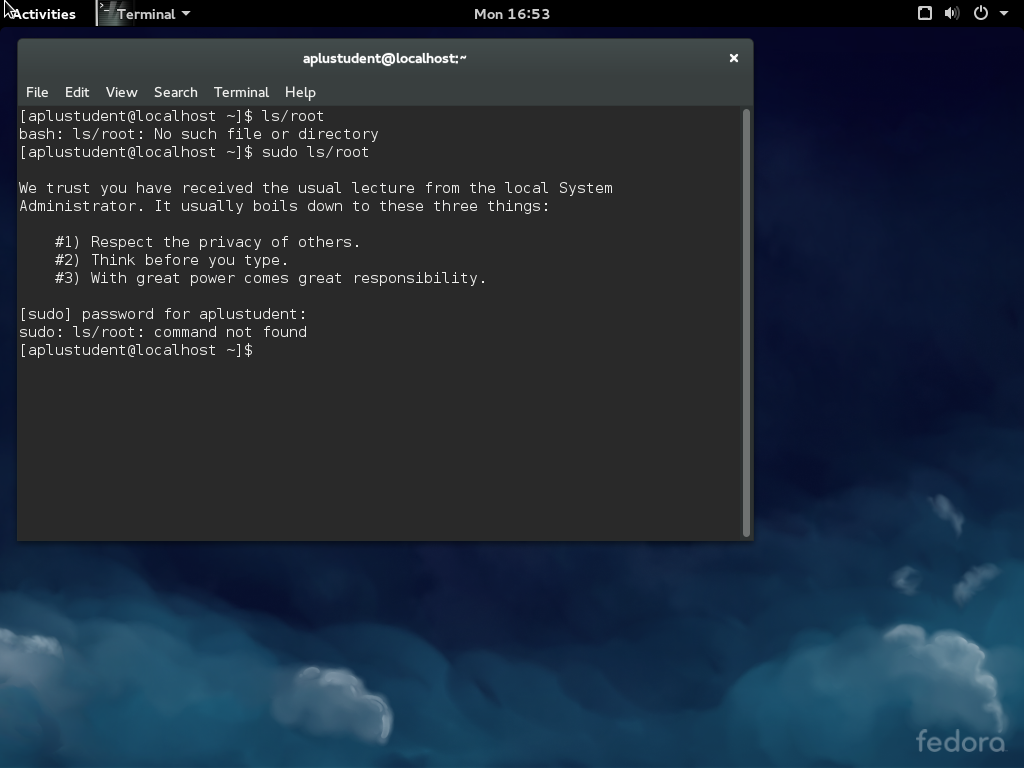
**type ls /root**

**What is the error message?**

**What command can be added in order to give us the correct permissions?**

**Add this command and take a screenshot of the output**

**A**: The error message is “bash: ls/root: No such file or directory” and we should add “sudo”.



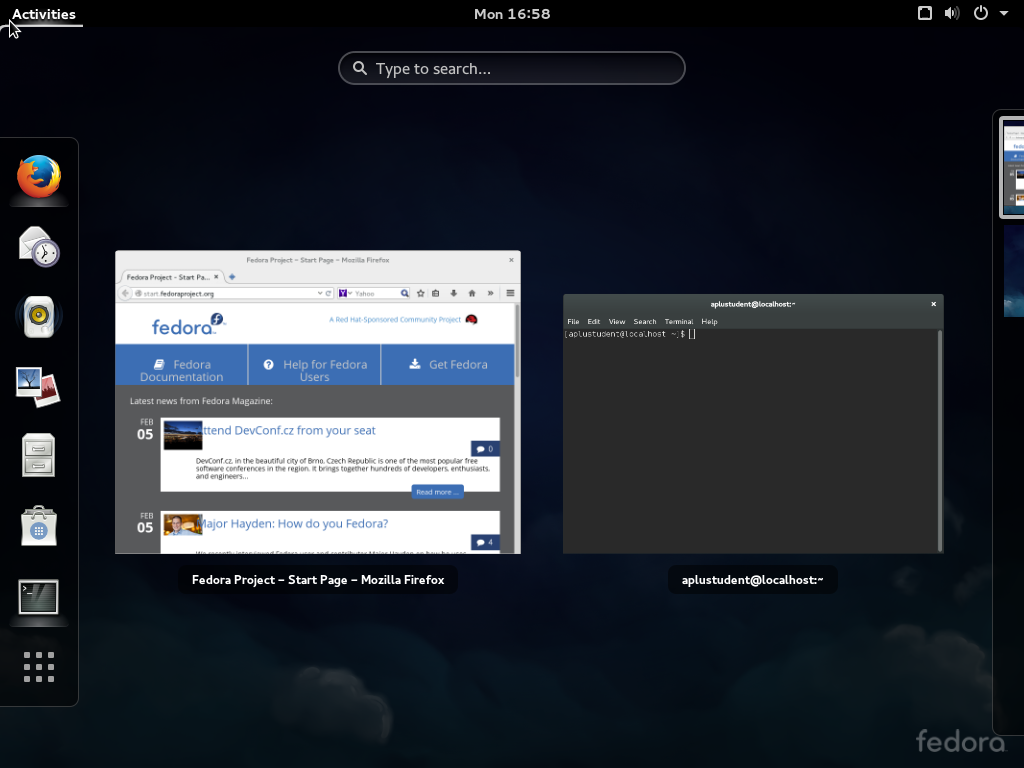
**2. Launch Firefox from the Activites button**

**then click activities**

**select a second virtual desktop**

**Launch a terminal on the virtual desktop**

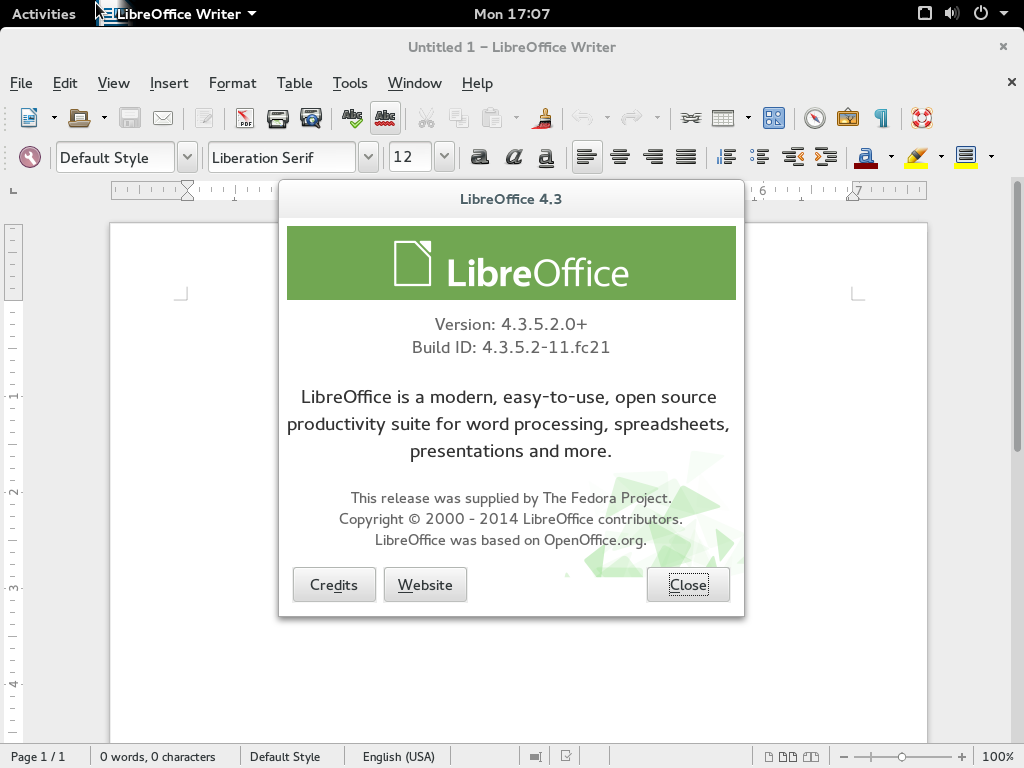
**Click the Activities button and hover over the virtual desktops and take a screenshot**



**3. Launch Libre Office Writer from the Activities menu**

**Click on the white triangle on the Title Bar (near the name of the application) select "About"**

**Submit a screenshot**



**4. Visit http://vivaldi.com download the rpm for Fedora Linux to downloads**

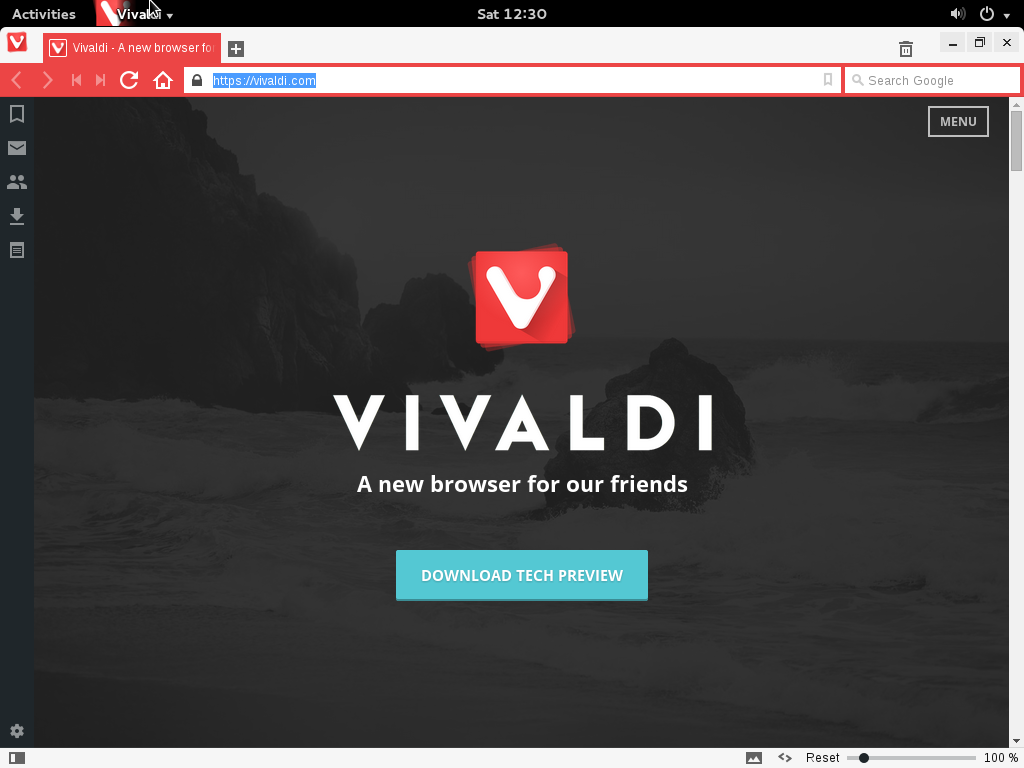
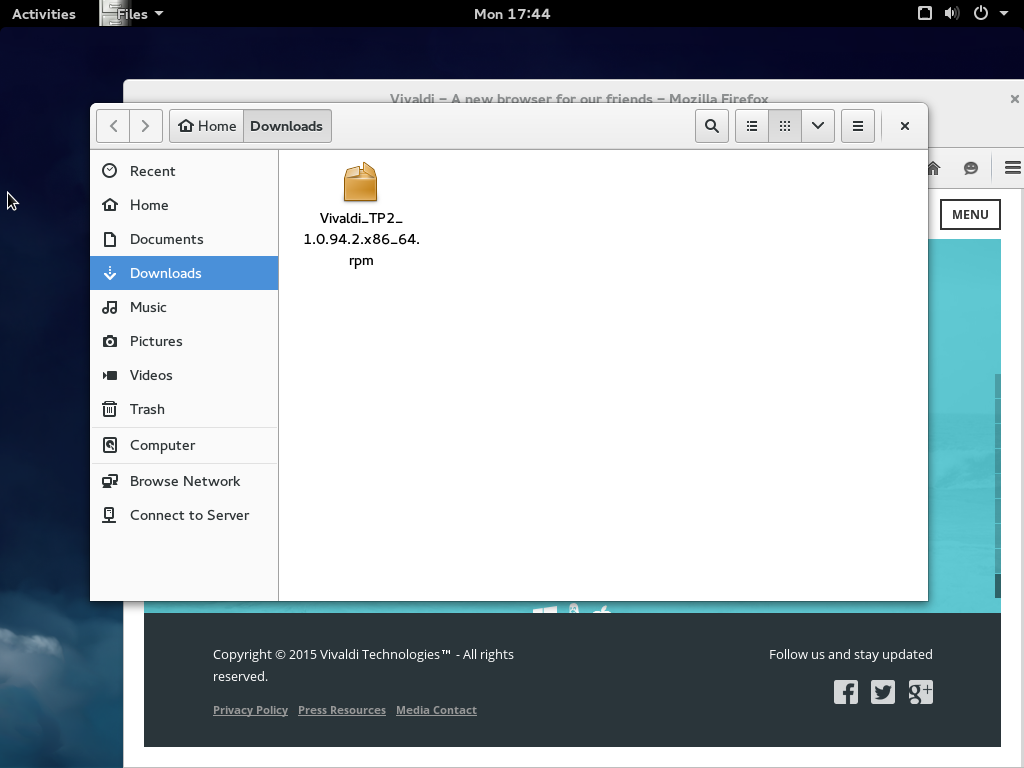
**Open Nautilus File Manager**

**Select Downloads**

**Take screenshot of downloads directory**

**Install the Vivaldi RPM**

**Launch Vivaldi from Activities and take a screeshot of the Vivaldi web browser**



**5. Launch Settings from Activities**

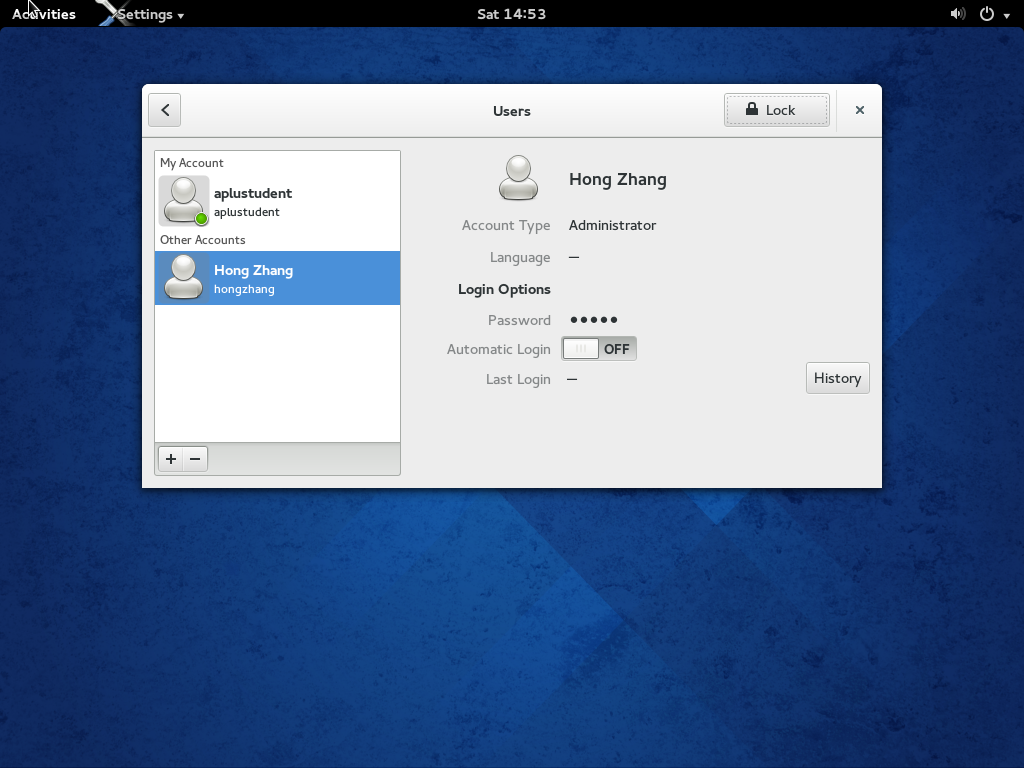
**Open Users Tab and add a new user account with your name**

**Take a screenshot**

**Open the Background Tab**

**Select a new background**

**Take a screenshot of the new desktop background**





**6. Launch Software from Activities**

**Click on the Development Software Tab**

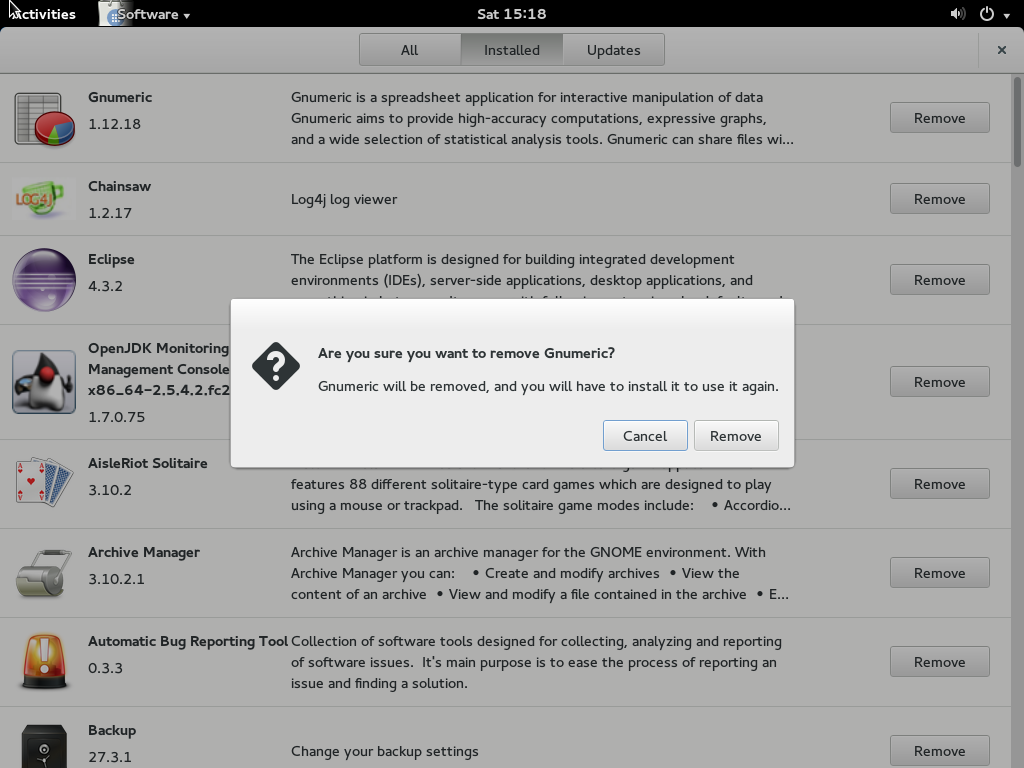
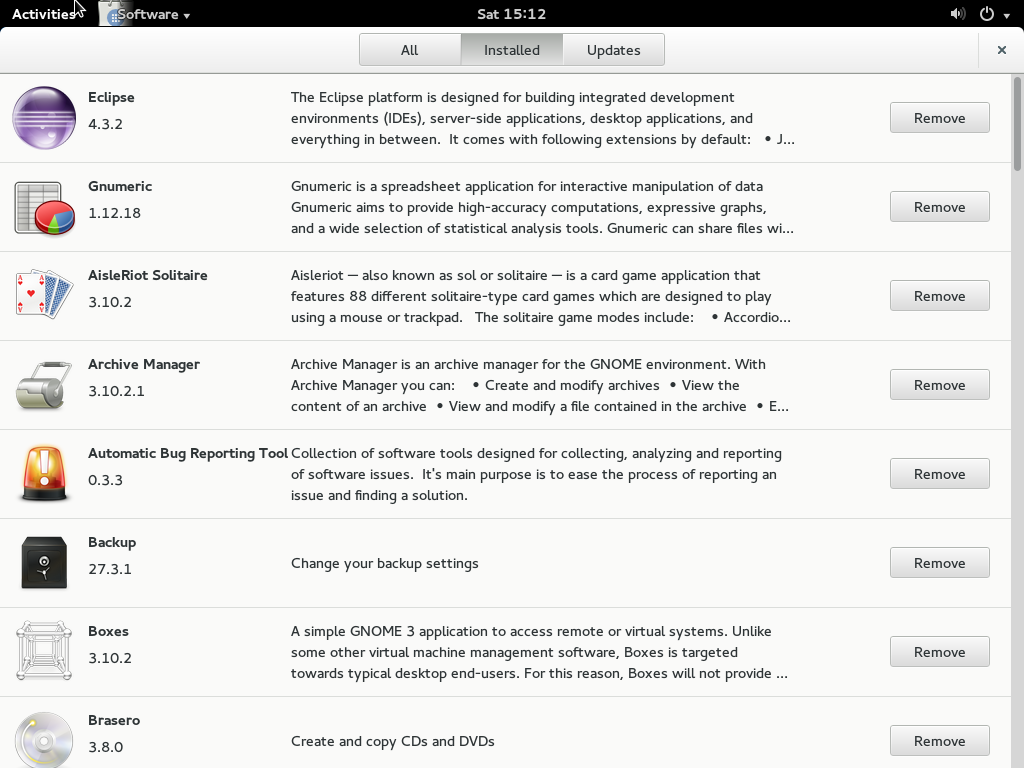
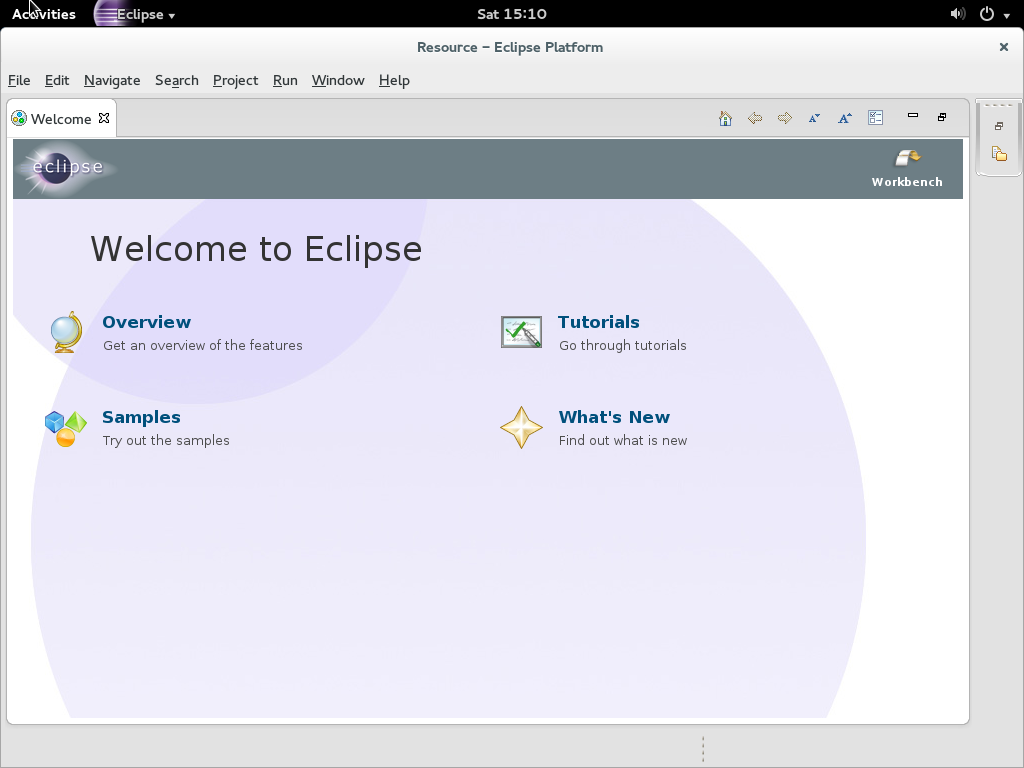
**Select Eclipse IDE for install**

**Launch Eclipse after install and take a screenshot**

**On the ALL tab select Office and install GNUMERIC**

**Take a screenshot of the INSTALLED TAB showing that GNUMERIC is present**

**Uninstall GNUMERIC and take a screenshot showing GNUMERIC uninstalling**



**7. Open a TERMINAL from Activities**

**type yum update - what happens? Why?**

**What command is needed to give yum enough install permissions?**

**Type sudo yum group install lxde-desktop**

**Upon completion in the upper right hand corner (white arrow) select log output**

**You will be presented with the GDM again (login but this time before you type your password - there will be a little gear icon next to your password box. Click it and change your desktop from GNOME to LXDE**

**Log in and take a screenshot of LXDE desktop**

**Don't forget to logout and change back to GNOME 3 if you want**

**A:** When I type “yum update”, I get an error message, “You need to be root to perform this command” because root privileges are needed for some operations. Sudo will give yum enough install permissions.



**8. Shell commands short answer:**

**Which command option for ls will show Long Listing?**

A: -l is the option which will give you a long listing format

**Which command option for cp will show verbosely what is being copied?**

A: -v — verbose shows the progress of the files being copied.

**Which command for rm will ask for interactive permission before completing the operations?**

**A:** -I (-i) –interactive prompt before any removal

**9. Shell commands short answer:**

**What does the command ls -Fcla show?**

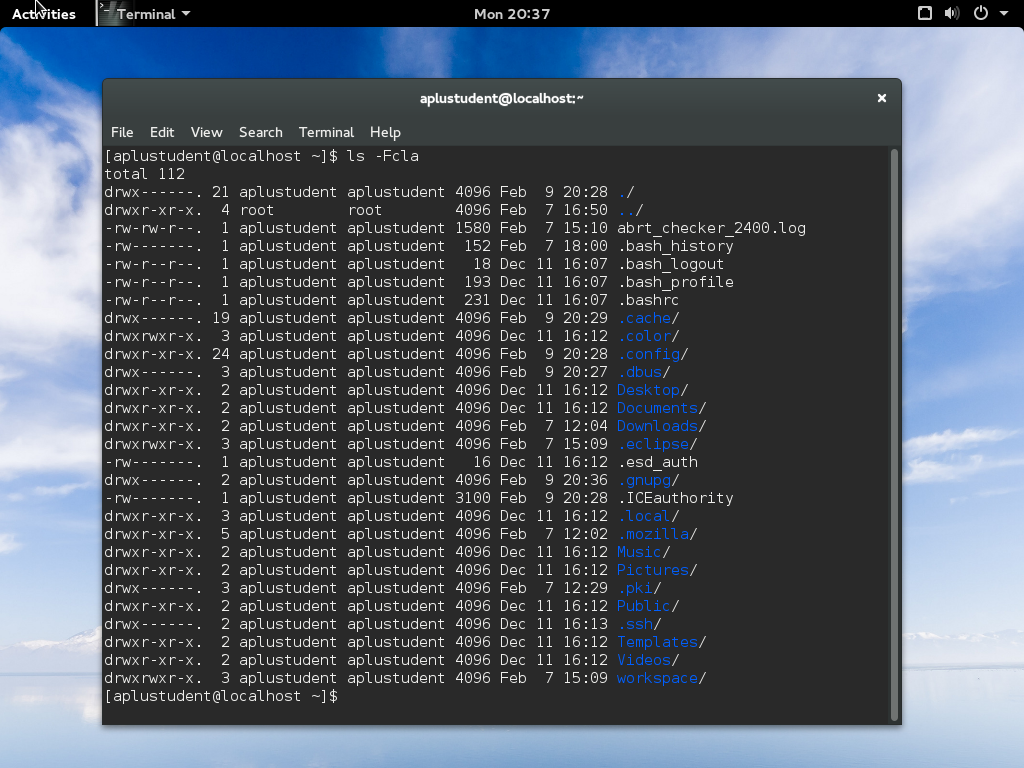
A: The ls command is used to list the contents of a directory and –Fcla is many options which include –F, –c (–etc), –l and –a.

–F (file type) adds a symbol to the end of each listing. These symbols include /, to indicate a directory; @, to indicate a symbolic link to another file; and \*, to indicate an executable file.

–c(etc) Lists the files in the /etc directory in long format.

–l (long) lists details about contents, including permissions (modes), owner, group, size, creation date, whether the file is a link to somewhere else on the system and where its link points.

–a (all/archive) lists all files in the directory, including hidden files (.filename).



**Which command displays the content of a text file?**

A: cat

**How do you rename a file on the commandline?**

A: mv

**10. You are asked to run the top command but are not sure on its usage and purpose - what would be an acceptable command to find out? (No google is not the right answer)**

**(also give a short explanation of what the top command does)**

A: Manual command or man pages can help us to find all options and features of a shell command. Type “man cp” in the terminal and we can find a short explanation for the top command: it provides a dynamic real-time view of a running system. It can display system summary information, as well as a list of processes or threads currently being managed by the kernel. The types of system summary information shown and the types, order and size of information displayed for tasks are all user-configurable.

**ITMO/IT-O 456 Laboratory Week 4 - chapter 6**

1. After your Linux system has been loaded, launch a terminal window

At the command prompt, type ls -F and press Enter.

At the command prompt, type mkdir mysamples and press Enter. Next, type ls -F at the command prompt and press Enter. How many files and subdirectories are there?

Three

At the command prompt, type cd mysamples and press Enter. Next, type ls -F at the command prompt and press Enter. What are the contents of the subdirectory mysamples?

Nothing.

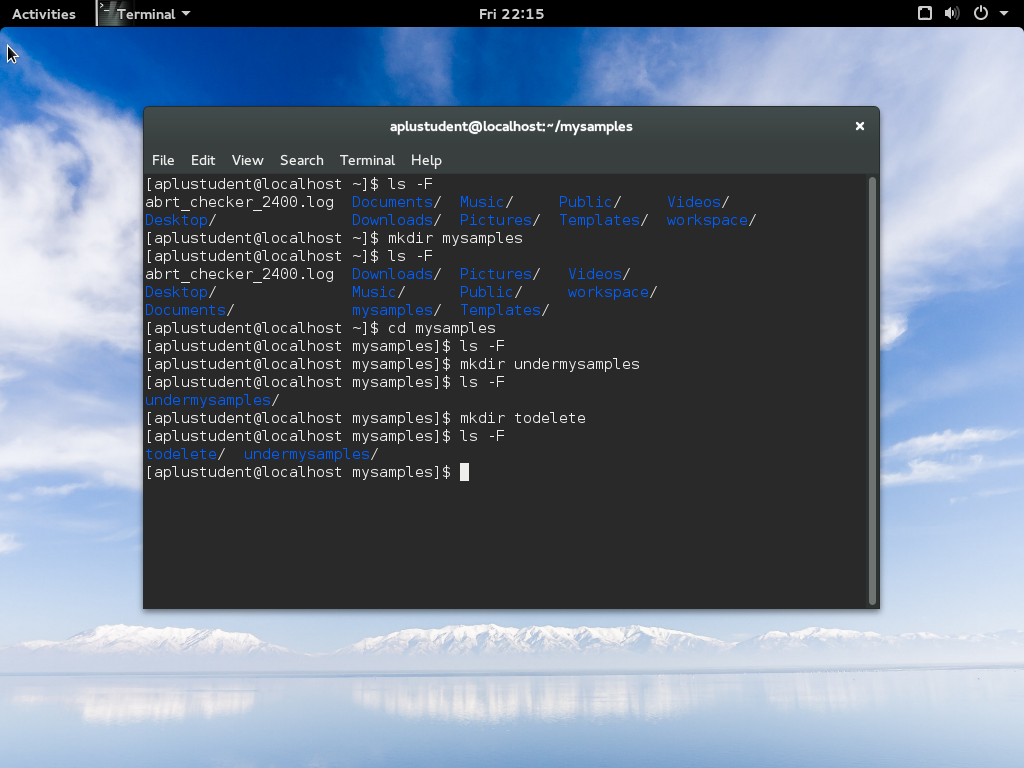
At the command prompt, type mkdir undermysamples and press Enter. Next, type ls -F at the command prompt and press Enter. What are the contents of the subdirectory mysamples?

undermysamples/

At the command prompt, type mkdir todelete and press Enter. Next, type ls -F at the command prompt and press Enter. Does the subdirectory todelete you just created appear listed in the display?

Yes.

Provide screen shot of the results



2. At the command prompt, type cd .. and press Enter. Next, type ls -R and press Enter. Notice that the subdirectory mysamples and its subdirectory undermysamples are both displayed. You have used the recursive option with the ls command.

At the command prompt, type cd .. and press Enter. At the command prompt, type pwd and press Enter. What is your current directory?

My current directory: /

At the command prompt, type sudo mkdir topsecret and press Enter. At the command prompt, type ls -F and press Enter. Does the subdirectory you just created appear listed in the display?

Yes.

Next, type ls -F at the command prompt and press Enter. Note the contents of your home folder.

boot/ home/ lost+found/ opt/ run/ sys/ topsecret/

3. At the command prompt, type touch sample1 and press Enter to create the file sample1.

At the command prompt, type cp sample1 and press Enter. What error message was displayed and why?

Error message: cp: missing destination file operand after ‘sample1’

Try ‘cp --help’ for more information.

Reason: cp is the command which makes a copy of your files or directories. Hence, you should have offer original file which you want to copy and the destination file which you will copy to.

At the command prompt, type cp sample1 sample1A and press Enter. Next, type ls -F at the command prompt and press Enter. How many files are there, and what are their names? Why?

They are two files, sample1 and sample1A because sample1 is copied.

At the command prompt, type cp sample1 mysamples/sample1B and press Enter. Next, type ls -F at the command prompt and press Enter. How many files are there, and what are their names? Why?

They are two files, sample1 and sample1A because the current directory does not show the file, sample1B and we need to change the directory to mysamples if we want to see this file.

At the command prompt, type cd mysamples and press Enter. Next, type ls -F at the command prompt and press Enter. Was sample1B copied successfully?

Yes.

4. At the command prompt, type cp /home/aplusstudent/sample2 . and press Enter. Next, type ls -F at the command prompt and press Enter. How many files are there, and what are their names? Why?

I create a file named sample2 in home directory. Then, I change the directory to Music/. After I run the command cp /home/aplusstudent/sample2 . and ls -F, I find the file of sample2 because I move this file from home directory to Music/.

At the command prompt, type cp sample1B .. and press Enter. Next, type cd .. at the command prompt and press Enter. At the command prompt, type ls -F and press Enter. Was the sample1B file copied successfully?

I run the command of cpsample1B .. in mysmaples/. After I run the command of ls -F, the sample1B file was copied successfully and it is in the home directory.

At the command prompt, type cp sample1 sample2 sample3 mysamples and press Enter. What message do you get and why? Choose y and press Enter. Next, type cd mysamples at the command prompt and press Enter. At the command prompt, type ls -F and press Enter. How many files are there, and what are their names? Why?

When I enter the directory, mysamples and use ls –F, there are three files, sample1 sample2 and sample3. The reason is I copy these three files to mysamples which is the destination.

At the command prompt, type cd .. and press Enter. Next, type cp mysamples mysamples2 at the command prompt and press Enter. What error message did you receive? Why?

Error message: cp: cannot stat ‘mysamples’: No such file or directory

The reason is the command, cd .. will take you up to 1 level. When you change the directory, you will not find the files in new directory.

At the command prompt, type cp -R mysamples mysamples2 and press Enter. Next, type ls -F at the command prompt and press Enter. Was the directory copied successfully? Type ls -F mysamples2 at the command prompt and press Enter. Were the contents of mysamples successfully copied to mysamples2?

Yes, the directory was copied successfully and the contents of mysamples were successfully copied to mysamples2.

Type ls -F at the command prompt and press Enter. Note the contents of your home folder.

At the command prompt, type mv sample1 and press Enter. What error message was displayed and why?

Error message: mv: missing destination file operand after ‘sample1’

Try ‘mv --help’ for more information.

Reason: this command only offer the source of the copy operation and miss the destination.

At the command prompt, type mv sample1 sample4 and press Enter. Next, type ls -F at the command prompt and press Enter. How many files are listed, and what are their names? What happened to sample1?

I only see sample4 because sample1 moved to sample4.

5. At the command prompt, type mv sample4 mysamples and press Enter. Next, type ls -F at the command prompt and press Enter. How many files are there, and what are their names? Where did sample4 go?

I only see mysamples and sample4 is gone because I moved sample4 to the directory, mysamples. Then, sample4 is under mysamples.

At the command prompt, type cd mysamples and press Enter. Next, type ls -F at the command prompt and press Enter. Notice that the sample4 file you moved in Step 5 was moved here.

At the command prompt, type mv sample4 .. and press Enter. Next, type ls -F at the command prompt and press Enter. How many files are there, and what are their names? Where did the sample4 file go?

I find sample4 is gone and it goes to the first level.

At the command prompt, type cd .. and press Enter. Next, type ls -F at the command prompt, and press Enter to view the new location of sample4.

At the command prompt, type mv sample4 mysamples/sample2 and press Enter. What message appeared on the screen and why?

Nothing. It means it works well.

Type y and press Enter to confirm you want to overwrite the file in the destination folder.

6. At the command prompt, type mv sample? mysamples and press Enter. Type y and press Enter to confirm you want to overwrite the file sample3 in the destination folder.

At the command prompt, type ls -F and press Enter. How many files are there and why?

The file, sample2, sample3 are gone because the command, mv sample? Mysamples just move these files to mysamples.

At the command prompt, type mv sample1\* mysamples and press Enter. Type y and press Enter to confirm you want to overwrite the file sample1B in the destination directory.

At the command prompt, type ls -F and press Enter. Notice that there are no sample files in the /root directory.

At the command prompt, type cd mysamples and press Enter. Next, type ls -F at the command prompt and press Enter. Notice that all files originally in /root have been moved to this directory.

At the command prompt, type cd .. and press Enter. Next, type ls -F at the command prompt and press Enter. Type mv mysamples samples and press Enter. Next, type ls -F at the command prompt and press Enter. Why did you not need to specify the recursive option to the mv command to rename the mysamples directory to samples?

Because they are in the same directory.

At the command prompt, type cd samples and press Enter. At the command prompt, type ls -R and press Enter. Note the two empty directories todelete and undermysamples.

7. At the command prompt, type rmdir undermysamples todelete and press Enter. Did the command work? Why? Next, type ls -F at the command prompt and press Enter. Were both directories deleted successfully?

Yes, it works because this command deletes two subdirectories, undermysamples and todelete. When I use ls –F to check , they were deleted successfully.

At the command prompt, type rm sample1\* and press Enter. What message is displayed? Answer n to all three questions.

At the command prompt, type rm -f sample1\* and press Enter. Why were you not prompted to continue? Next, type ls -F at the command prompt and press Enter. Were all three files deleted successfully?

This command will delete all files which start with sample1.

At the command prompt, type cd .. and press Enter. Next, type rmdir samples at the command prompt and press Enter. What error message do you receive and why?

Error message: rmdir: failed to remove ‘samples’: Directory not empty.

Reason: this directory, samples has some files and we can only delete empty directory.

At the command prompt, type rm -Rf samples and press Enter. Next, type ls -F at the command prompt and press Enter. Were the samples directory and all the files within it deleted successfully?

Yes. The option –Rf will delete anything.

At the command prompt, type touch permsample and press Enter. Next, type chmod 777 permsample at the command prompt and press Enter.

At the command prompt, type ls -l and press Enter. Who has permissions to this file?

Owner, group and others have permissions to this file.

At the command prompt, type chmod 000 permsample and press Enter. Next, type ls -1 at the command prompt and press Enter. Who has permissions to this file?

Nobody has permission to this file. (Root alone has permissions at 000)

8. At the command prompt, type rm -f permsample and press Enter. Were you able to delete this file? Why?

Yes, I can delete this file because –f option means force and it can delete anything.

At the command prompt, type cd / and press Enter. Next, type pwd at the command prompt and press Enter. What directory are you in? Type ls -F at the command prompt and press Enter. What directories do you see?

The directory is /. I can see etc/, mnt/, root/, srv/, tmp/, var/, boot/, home/, lost+found/, opt/, run/, sys/, topsecret/, dev/, media/, proc/, sysroot/ and usr/.

At the command prompt, type ls -1 and press Enter to view the owner, group owner, and permissions on the topsecret directory created earlier. Who is the owner and group owner? If you were logged in as the user "aplusstudent", in which category would you be placed (user, group, other)? What permissions do you have as this category (read, write, execute)?

The owner and group owner is root. If I was logged in as the user "aplusstudent", I will be placed in user. As a user, I can read, write and execute.

9. At the command prompt, type cd topsecret and press enter to enter the directory. Next, type ls -F at the command prompt and press Enter. Are there any files in this directory? Type cp /etc/hosts . at the command prompt and press Enter. Next, type ls -F at the command prompt and press Enter to ensure that a copy of the hosts file was made in your current directory.

There is no file in topsecret/. After I run the command cp /etc/hosts ., the file of hosts moved to topsecret/ from etc/.

At the command prompt, type ls -l and press Enter. Who is the owner and group owner of this file? If you were logged in as the user user1, in which category would you be placed (user, group, other)? What permissions do you have as this category (read, write, execute)?

aplustudent is the onwner and group owner of this file. If I was logged in as the user user1, I will be placed in other. Only user has permissions to read, write and execute.

10. At the command prompt, type cd topsecret and press Enter to enter the topsecret directory. Type cat hosts at the command prompt and press Enter. Were you successful? Why? Next, type nano hosts at the command prompt to open the hosts file in the nano editor. Delete the first line of this file by placing the cursor in the line and pressing Ctrl+K, then press Ctrl+O and Enter to save your changes. Were you successful? Why? Exit the editor by pressing Ctrl+X and discard your changes.

Yes, I was successful to run first command because I l have the permission to read.

However, I was not successful to run second command because I have no permission to write.

At the command prompt, type chmod o+w topsecret/hosts and press Enter.

At the command prompt, type cd topsecret and press Enter to enter the topsecret directory. Type nano hosts at the command prompt to open the hosts file in the nano editor. Delete the first line of this file by placing the cursor in the line and pressing Ctrl+K, then press Ctrl+O and Enter to save your changes. Why were you successful this time? Exit the editor by pressing Ctrl+X.

I use the command **chmod** to change file’s permission, **o** means others and **w** means write.Now it can be wrote.

At the command prompt, type ls -l and press Enter. Do you have permission to execute the hosts file? Should you make this file executable? Why?

When I run this command of **ls -l**, I see **-rw-r--rw**- which means I have no permission to execute the hosts file. Hence, I cannot make this file executable.

Next, type ls -1 /bin at the command prompt and press Enter. Note how many of these files to which you have execute permission. Type file /bin/\* at the command prompt and press Enter to view the file types of the files in the /bin directory. Should these files have the execute permission?

I have execute permission as user, group or others (**lrwxrwxrwx.**) When I type file /bin/\*, I found these files have the execute permission.

Lab 5:

1. What does the shell ordinarily do while a command is executing? What should you do if you do not want to wait for a command to finish before running another command?

A: While the command is executing, the shell warts for the process to finish and it sleeps. When the program finishes execution, the shell returns to an active state (wakes up), issues a prompt, and waits for another command. When you run a command in the background, you don’t have to wait for the command to finish before running another command.

2. Using sort as a filter, rewrite the following sequence of commands:

**$ sort list > temp**

**$ lpr temp**

**$ rm temp**

A: command: $ cat list | sort | lpr

3. What is a PID number? Why are these numbers useful when you run processes in the background? Which utility displays the PID numbers of the commands you are running?

A: PID number (process identification number) is a larger number assigned by the operating system. These numbers are useful because each of them identifies the command running in the background. PS (process status) utility displays the PID numbers of the commands you are running.

4. Assume the following files are in the working directory:

**$ ls**

intro notesb ref2 section1 section3 section4b

notesa ref1 ref3 section2 section4a sentrev

Give commands for each of the following, using wildcards to express filenames with as few characters as possible.

a. List all files that begin with **section**.

A: $ **ls section\***

b. List the **section1**, **section2**, and **section3** files only.

A: $ **ls section[1-3]**

c. List the **intro** file only.

A: $ **ls i\***

d. List the **section1**, **section3**, **ref1**, and **ref3** files.

A: $ **ls \*[13]**

5. Refer to the info or man pages to determine which command will

(hint look at the wc command)

a. Display the number of lines in its standard input that contain the word **a** or **A**.

A: $ **command | grep –wci a**

b. Display only the names of the files in the working directory that contain the pattern **$(**.

A: **$ls \*$\(\***

c. List the files in the working directory in reverse alphabetical order.

A: **$ls | sort –r or $ ls -R**

d. Send a list of files in the working directory to the printer, sorted by size.

A: **$ ls -S | lpr**

6. Give a command to

a. Redirect standard output from a sort command to a file named **phone\_list**. Assume the input file is named **numbers**.

A: $ **sort numbers > phone\_list**

b. Translate all occurrences of the characters **[** and **{** to the character **(**, and all occurrences of the characters **]** and **}** to the character **),** in the file **permdemos.c**. (*Hint:* Refer to the tr man page.)

A: $ **cat permdemos.c | tr '[{}]' '(())'**

c. Create a file named **book** that contains the contents of two other files:

**part1** and **part2**.

A: $ **cat part[12] > book**

7. The lpr and sort utilities accept input either from a file named on the command line or from standard input.

a. Name two other utilities that function in a similar manner.

A: grep, cat

b. Name a utility that accepts its input only from standard input.

A: tr

8. Give an example of a command that uses **grep**

a. With both input and output redirected.

A: $ **grep \$Name < \*.c > name\_list**

b. With only input redirected.

A: $ **grep -i aplustudent < demo**

c. With only output redirected.

A: $ **grep asd hong.txt > hong2.txt**

d. Within a pipeline.

A: $ **file /usr/bin/\* | grep "Again shell script" | sort -r**

In which of the preceding cases is grep used as a filter?

A: Part **d** uses grep as a filter.

9. Explain the following error message. Which filenames would a subsequent ls command display?

$ **ls**

abc abd abe abf abg abh

**$ rm abc ab \***

rm: cannot remove 'abc': No such file or directory

A: It is duplicated. “ab\*” means any files which starts ab. Hence, rm receives a list of files that includes abc twice. After rm removes abc, it will get an error message when it is asked to

remove abc again. After giving the preceding rm command, ls does not list any files.

10. When you use the redirect output symbol (**>**) on a command line, the shell creates the output file immediately, before the command is executed. Demonstrate that this is true.

A: I will some commands as follows:

$ **ls hong**

ls: hong: No such file or directory

$ **ls id > hong**

ls: id: No such file or directory

$ **ls hong**

hong

The first command shows the file **hong** does not exist in the working directory. The second command uses **ls** to attempt to list a nonexistent file (**id**) and sends standard output to **hong**. The **ls** command fails and sends an error message to standard error. Even though the **ls** command failed, the empty file named **hong** exists. Because the **ls** command failed, it did not create the file; the shell created it before calling **ls**.

11. In experimenting with variables, Max accidentally deletes his **PATH** variable. He decides he does not need the **PATH** variable. Discuss some of the problems he could soon encounter and explain the reasons for these problems. How could he *easily* return **PATH** to its original value?

A: When he deletes it, there is no way to connect users with commands or utilities. Hence, it is very hard to run these commands because Max does not know where the directories or files are. A simple way to return PATH to its original value is to log out and then log back in.

12. Assume permissions on a file allow you to write to the file but not to delete it.

a. Give a command to empty the file without invoking an editor.

A: $ **cat /dev/null > hong\_file**

b. Explain how you might have permission to modify a file that you cannot delete.

A: When you have write permission only for this file and execute permission only for the directory, you can modify it but you cannot delete it.

13. If you accidentally create a filename that contains a nonprinting character, such as a CONTROL character, how can you remove the file?

A: To remove it, you should use some wildcard characters to find it. Firstly, use echo to confirm it. Secondly, use rm to delete it. For example, assume we want to remove the file named hongCONTROL-u. The commands are:

$ echo hong?u

hongu

$ rm hong?u

14. Why does the noclobber variable *not* protect you from overwriting an existing file with cp or mv?

A: **noclobber** prevents overwriting a file using redirection (>) but it does work with commands or utilities, such as cp and mv.

15. Why do command names and filenames usually not have embedded SPACEs? How would you create a filename containing a *SPACE*? How would you remove it? (This is a thought exercise, not recommended practice. If you want to experiment, create a file and work in a directory that contains only your experimental file.)

A: SPACE has special meaning to the shell. It separate tokens (elements) on the command line. If you want to create or remove a filename containing a SPACE, you should quote the SAPCE. For instance : touch hong\ zhang.txt; rm hong\ zhang.txt.

16. Create a file named **answer** and give the following command:

$ **> answers.0102 < answer cat**

Explain what the command does and why. What is a more conventional way of expressing this command?

A: this command asks shell to redirect standard output to **answers.0102**, to redirect standard input from **answer**, and to execute the cat utility. More conventionally, the same command is expressed as **cp answer answers.0102**

**Lab Chapter 7**

============================

**1. At the command prompt, type touch newfile and press Enter.**

**At the command prompt, type find / -name "newfile" and press Enter. Did the find command find the file? Why not?**

No. I cannot find this file because this command searches the root to find this file and this command search entire file system including mounted devices and network storage devices. It totally lost.

**At the command prompt, type find /home/aplustudent -name "newfile" and press Enter. Did the find command find the file? How quickly did it find it? Why? (NOTE - /home/aplustudent can be replaced with ~)**

Yes. It is very quickly to find it because this command offers the directory to find this file.

**2.** **At the command prompt, type which newfile and press Enter.**

**Did the which command find the file? Why or why not?**

No. The message shows "/usr/bin/which: no newfile in (/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/bin:/sbin:/home/aplustudent/ .local/bin:/home/aplustudent/bin)"

which locates a utility. newfile is created by the user and it is not a utility.

**Type echo $PATH at the command prompt and press Enter. Is the /home/aplustudent directory listed in the PATH variable?**

Yes. the /home/aplustudent directory is listed in the PATH variable. (No. the /home/aplustudent directory is not listed in the PATH variable. Is /home/aplustudent/.local/bin)

**Is the /bin directory listed in the PATH variable?**

Yes.

**3. At the command prompt, type which grep and press Enter. Did the which command find the file? Why?**

Yes, the path is "usr/bin/grep". Because grep is a utility, which command can find it through a system path.

**At the command prompt, type find /home/aplustudent -type l and press Enter. (Note that the last character in this command is the lowercase letter "l" and not the number one.) What files are listed? Why?**

The file is /home/aplustudent/ .local/share/systemd/user because "-type l" searches for symbolic links.

**At the command prompt, type find /home/aplustudent -size 0 and press Enter. What types of files are listed? Type find /home/aplustudent -size 0 | more to see all of the files listed.**

This command finds all files the size is 0. If we add the command more and we will see these files one screen at a time.

**4. At the command prompt, type cat /etc/sysconfig/network-scripts/ifcfg-p2p1 and press Enter to view the contents of the file hosts, which reside in the directory /etc.**

p2p1

**Next, type cat -n /etc/hosts and press Enter. How many lines does the file have?**

Two lines. It offers local DNS.

**At the command prompt, type tac /etc/sysconfig/network-scripts/ifcfg-eth0 and press Enter to view the same file in reverse order. The output of both commands should be visible on the same screen. Compare them.**

**To see the contents of the same file in octal format instead of ASCII text, type od /etc/hosts at the command prompt and press Enter.**

**At the command prompt, type cat /etc/services and press Enter.**

**At the command prompt, type head /etc/services and press Enter. What is displayed on the screen? How many lines are displayed, which ones are they, and why?**

It shows the beginning of the file. There are ten lines. Because we use head, this file of services just display the beginning which includes ten lines of context.

**At the command prompt, type head -5 /etc/services and press Enter. How many lines are displayed and why?**

Because we use -5 which present 5 lines, it shows the beginning of the file and only shows 5 lines.

**Next, type head -3 /etc/inittab and press Enter. How many lines are displayed and why?**

Because we use -3 which present 3 lines, it shows the beginning of the file and only shows 3 lines.

**At the command prompt, type tail /etc/services and press Enter. What is displayed on the screen? How many lines are displayed, which ones are they, and why?**

The utility, tail shows the end of a file. Hence, it shows the last ten lines for /ect/services.

**At the command prompt, type tail -5 /etc/services and press Enter. How many lines are displayed and why?**

Because we use -5 which present 5 lines, it shows the end of the file and only shows 5 lines.

**Type the cat -n /etc/services command at a command prompt and press Enter to justify your answer.**

**5. At the command prompt, type cd ~ and press Enter to ensure we are in the /home/aplustudent directory;**

**Connect to Blackboard and under the CONTENT section Lab Week 6 there is a file to download called lab6.tar.gz Lab-week-6.zip download them both into your virtualmachine**

**Unzip the contents of the zip file into a directory called lab6-zips (hint use -d option)**

**Extraxct the tar ball using -xvzf options as well**

**cd to Lab6 directory**

$ unzip Lab-week6.zip –d lab6-zips

**6. At the command prompt, type grep "http" itmo456-spring2014-syllabus.md and press Enter. What is displayed and why?**

The utility, grep will search for a pattern in files. This command searches the pattern “http” in the file of itmo456-spring2014-syllabus.md and displays them.

**At the command prompt, type grep -v "http" itmo456-spring2014-syllabus.md and press Enter. What is displayed and why? How does this compare with the results from previous step?**

-v option prints inverts the match. That means that it only displays those lines that do not contain the pattern “http”.

**At the command prompt, type grep "consecrate" gettysburgaddress.txt and press Enter. What is displayed and why?**

This command searches the pattern “consecrate” in the file of gettysburgaddress.txtand displays them.

**At the command prompt, type grep -i "consecrate" gettysburgaddress.txt and press Enter. What is displayed and why? How does this compare with the results from Steps 22 and 24?**

-i option will ignore word case i.e match consecrate, CONSECRATE or Consecrate.

**At the command prompt, type grep "W" gettysburgaddress.txt and press Enter. What is displayed and why?**

This command searches the pattern “W” in the file of gettysburgaddress.txtand displays them.

**At the command prompt, type grep "c..e" gettysburgaddress.txt and press Enter. What is displayed and why?**

This command searches the pattern “c..e” in the file of gettysburgaddress.txtand displays them.

By the way, the pattern “c..e” means the word includes “c”, “e” and between them are any two characters.

**At the command prompt, type grep " ^N" gettysburgaddress.txt and press Enter. What is displayed and why?**

There is nothing because this pattern starts a space.

**At the command prompt, type grep "^N" gettysburgaddress.txt and press Enter. What is displayed and why? How does this differ from the results in Step 29 and why?**

It displays a paragraph which starts with “N” because ^ will match at the start of a line. So “^N” matches at the beginning of a line followed by N. this question differ from last one because last one had a space before “^N”.

**At the command prompt, type grep ".$" gettysburgaddress.txt and press Enter. Is anything displayed? (Hint: Be certain to look closely!) Can you explain the output?**

In REGEX language, $ means EOL (end of line). “.$”matches any line ending in “.”. hence, there is nothing because in the file, there is nothing after “.”.

**7. What is the command to CUT the second column of IP Addresses from the file hosts.deny and then pipe "|" them to the sort command?**

$ cat hosts.deny

$ cut –d‘:’ –f2 hosts.deny | sort

**8. Using the diff command print out the difference between hosts.deny and hosts.deny.smaller files**

$ diff hosts.deny hosts.deny.smaller

**9. Using the file command - what type of file is Lab6.tar.gz**

It shows the type of this file is gzip.

**10. Use the date command to print out an America style date with formatting like this: mm/dd/yyyy hr:mn:ss then give the command to print a rest of the world style date dd/mm/yyyy hr:mn:ss**

America style date: date +“%m%d%y %T”

World style date : date +“%d%m%y %T”

===============================================================================

= W e l c o m e t o t h e V I M T u t o r - Version 1.7 =

===============================================================================

Vim is a very powerful editor that has many commands, too many to

explain in a tutor such as this. This tutor is designed to describe

enough of the commands that you will be able to easily use Vim as

an all-purpose editor.

The approximate time required to complete the tutor is 25-30 minutes,

depending upon how much time is spent with experimentation.

ATTENTION:

The commands in the lessons will modify the text. Make a copy of this

file to practise on (if you started "vimtutor" this is already a copy).

It is important to remember that this tutor is set up to teach by

use. That means that you need to execute the commands to learn them

properly. If you only read the text, you will forget the commands!

Now, make sure that your Shift-Lock key is NOT depressed and press

the j key enough times to move the cursor so that Lesson 1.1

completely fills the screen.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 1.1: MOVING THE CURSOR

\*\* To move the cursor, press the h,j,k,l keys as indicated. \*\*

^

k Hint: The h key is at the left and moves left.

< h l > The l key is at the right and moves right.

j The j key looks like a down arrow.

v

1. Move the cursor around the screen until you are comfortable.

2. Hold down the down key (j) until it repeats.

Now you know how to move to the next lesson.

3. Using the down key, move to Lesson 1.2.

NOTE: If you are ever unsure about something you typed, press <ESC> to place

you in Normal mode. Then retype the command you wanted.

NOTE: The cursor keys should also work. But using hjkl you will be able to

move around much faster, once you get used to it. Really!

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 1.2: EXITING VIM

!! NOTE: Before executing any of the steps below, read this entire lesson!!

1. Press the <ESC> key (to make sure you are in Normal mode).

2. Type: :q! <ENTER>.

This exits the editor, DISCARDING any changes you have made.

3. When you see the shell prompt, type the command that got you into this

tutor. That would be: vimtutor <ENTER>

4. If you have these steps memorized and are confident, execute steps

1 through 3 to exit and re-enter the editor.

NOTE: :q! <ENTER> discards any changes you made. In a few lessons you

will learn how to save the changes to a file.

5. Move the cursor down to Lesson 1.3.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 1.3: TEXT EDITING - DELETION

\*\* Press x to delete the character under the cursor. \*\*

1. Move the cursor to the line below marked --->.

2. To fix the errors, move the cursor until it is on top of the

character to be deleted.

3. Press the x key to delete the unwanted character.

4. Repeat steps 2 through 4 until the sentence is correct.

---> The cow jumped over the moon.

5. Now that the line is correct, go on to Lesson 1.4.

NOTE: As you go through this tutor, do not try to memorize, learn by usage.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 1.4: TEXT EDITING - INSERTION

\*\* Press i to insert text. \*\*

1. Move the cursor to the first line below marked --->.

2. To make the first line the same as the second, move the cursor on top

of the first character AFTER where the text is to be inserted.

3. Press i and type in the necessary additions.

4. As each error is fixed press <ESC> to return to Normal mode.

Repeat steps 2 through 4 to correct the sentence.

---> There is some text missing from this line.

---> There is some text missing from this line.

5. When you are comfortable inserting text move to lesson 1.5.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 1.5: TEXT EDITING - APPENDING

\*\* Press A to append text. \*\*

1. Move the cursor to the first line below marked --->.

It does not matter on what character the cursor is in that line.

2. Press A and type in the necessary additions.

3. As the text has been appended press <ESC> to return to Normal mode.

4. Move the cursor to the second line marked ---> and repeat

steps 2 and 3 to correct this sentence.

---> There is some text missing from this line.

There is some text missing from this line.

---> There is also some text missing here.

There is also some text missing here.

5. When you are comfortable appending text move to lesson 1.6.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 1.6: EDITING A FILE

\*\* Use :wq to save a file and exit. \*\*

!! NOTE: Before executing any of the steps below, read this entire lesson!!

1. Exit this tutor as you did in lesson 1.2: :q!

Or, if you have access to another terminal, do the following there.

2. At the shell prompt type this command: vim tutor <ENTER>

'vim' is the command to start the Vim editor, 'tutor' is the name of the

file you wish to edit. Use a file that may be changed.

3. Insert and delete text as you learned in the previous lessons.

4. Save the file with changes and exit Vim with: :wq <ENTER>

5. If you have quit vimtutor in step 1 restart the vimtutor and move down to

the following summary.

6. After reading the above steps and understanding them: do it.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 1 SUMMARY

1. The cursor is moved using either the arrow keys or the hjkl keys.

h (left) j (down) k (up) l (right)

2. To start Vim from the shell prompt type: vim FILENAME <ENTER>

3. To exit Vim type: <ESC> :q! <ENTER> to trash all changes.

OR type: <ESC> :wq <ENTER> to save the changes.

4. To delete the character at the cursor type: x

5. To insert or append text type:

i type inserted text <ESC> insert before the cursor

A type appended text <ESC> append after the line

NOTE: Pressing <ESC> will place you in Normal mode or will cancel

an unwanted and partially completed command.

Now continue with Lesson 2.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 2.1: DELETION COMMANDS

\*\* Type dw to delete a word. \*\*

1. Press <ESC> to make sure you are in Normal mode.

2. Move the cursor to the line below marked --->.

3. Move the cursor to the beginning of a word that needs to be deleted.

4. Type dw to make the word disappear.

NOTE: The letter d will appear on the last line of the screen as you type

it. Vim is waiting for you to type w . If you see another character

than d you typed something wrong; press <ESC> and start over.

---> There are some words that don't belong in this sentence.

5. Repeat steps 3 and 4 until the sentence is correct and go to Lesson 2.2.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 2.2: MORE DELETION COMMANDS

\*\* Type d$ to delete to the end of the line. \*\*

1. Press <ESC> to make sure you are in Normal mode.

2. Move the cursor to the line below marked --->.

3. Move the cursor to the end of the correct line (AFTER the first . ).

4. Type d$ to delete to the end of the line.

---> Somebody typed the end of this line twice.

5. Move on to Lesson 2.3 to understand what is happening.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 2.3: ON OPERATORS AND MOTIONS

Many commands that change text are made from an operator and a motion.

The format for a delete command with the d delete operator is as follows:

d motion

Where:

d - is the delete operator.

motion - is what the operator will operate on (listed below).

A short list of motions:

w - until the start of the next word, EXCLUDING its first character.

e - to the end of the current word, INCLUDING the last character.

$ - to the end of the line, INCLUDING the last character.

Thus typing de will delete from the cursor to the end of the word.

NOTE: Pressing just the motion while in Normal mode without an operator will

move the cursor as specified.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 2.4: USING A COUNT FOR A MOTION

\*\* Typing a number before a motion repeats it that many times. \*\*

1. Move the cursor to the start of the line marked ---> below.

2. Type 2w to move the cursor two words forward.

3. Type 3e to move the cursor to the end of the third word forward.

4. Type 0 (zero) to move to the start of the line.

5. Repeat steps 2 and 3 with different numbers.

---> This is just a line with words you can move around in.

6. Move on to Lesson 2.5.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 2.5: USING A COUNT TO DELETE MORE

\*\* Typing a number with an operator repeats it that many times. \*\*

In the combination of the delete operator and a motion mentioned above you

insert a count before the motion to delete more:

d number motion

1. Move the cursor to the first UPPER CASE word in the line marked --->.

2. Type d2w to delete the two UPPER CASE words

3. Repeat steps 1 and 2 with a different count to delete the consecutive

UPPER CASE words with one command

---> this line of words is cleaned up.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 2.6: OPERATING ON LINES

\*\* Type dd to delete a whole line. \*\*

Due to the frequency of whole line deletion, the designers of Vi decided

it would be easier to simply type two d's to delete a line.

1. Move the cursor to the second line in the phrase below.

2. Type dd to delete the line.

3. Now move to the fourth line.

4. Type 2dd to delete two lines.

---> 1) Roses are red,

---> 3) Violets are blue,

---> 4) I have a car,

---> 7) And so are you.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 2.7: THE UNDO COMMAND

\*\* Press u to undo the last commands, U to fix a whole line. \*\*

1. Move the cursor to the line below marked ---> and place it on the

first error.

2. Type x to delete the first unwanted character.

3. Now type u to undo the last command executed.

4. This time fix all the errors on the line using the x command.

5. Now type a capital U to return the line to its original state.

6. Now type u a few times to undo the U and preceding commands.

7. Now type CTRL-R (keeping CTRL key pressed while hitting R) a few times

to redo the commands (undo the undo's).

---> Fix the errors on this line and replace them with undo.

8. These are very useful commands. Now move on to the Lesson 2 Summary.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 2 SUMMARY

1. To delete from the cursor up to the next word type: dw

2. To delete from the cursor to the end of a line type: d$

3. To delete a whole line type: dd

4. To repeat a motion prepend it with a number: 2w

5. The format for a change command is:

operator [number] motion

where:

operator - is what to do, such as d for delete

[number] - is an optional count to repeat the motion

motion - moves over the text to operate on, such as w (word),

$ (to the end of line), etc.

6. To move to the start of the line use a zero: 0

7. To undo previous actions, type: u (lowercase u)

To undo all the changes on a line, type: U (capital U)

To undo the undo's, type: CTRL-R

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 3.1: THE PUT COMMAND

\*\* Type p to put previously deleted text after the cursor. \*\*

1. Move the cursor to the first ---> line below.

2. Type dd to delete the line and store it in a Vim register.

3. Move the cursor to the c) line, ABOVE where the deleted line should go.

4. Type p to put the line below the cursor.

5. Repeat steps 2 through 4 to put all the lines in correct order.

---> a) Roses are red,

---> b) Violets are blue,

---> c) Intelligence is learned,

---> d) Can you learn too?

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 3.2: THE REPLACE COMMAND

\*\* Type rx to replace the character at the cursor with x . \*\*

1. Move the cursor to the first line below marked --->.

2. Move the cursor so that it is on top of the first error.

3. Type r and then the character which should be there.

4. Repeat steps 2 and 3 until the first line is equal to the second one.

---> When this line was typed in, someone pressed some wrong keys!

---> When this line was typed in, someone pressed some wrong keys!

5. Now move on to Lesson 3.3.

NOTE: Remember that you should be learning by doing, not memorization.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 3.3: THE CHANGE OPERATOR

\*\* To change until the end of a word, type ce . \*\*

1. Move the cursor to the first line below marked --->.

2. Place the cursor on the u in lubw.

3. Type ce and the correct word (in this case, type ine ).

4. Press <ESC> and move to the next character that needs to be changed.

5. Repeat steps 3 and 4 until the first sentence is the same as the second.

---> This line has a few words that need changing using the change operator.

---> This line has a few words that need changing using the change operator.

Notice that ce deletes the word and places you in Insert mode.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 3.4: MORE CHANGES USING c

\*\* The change operator is used with the same motions as delete. \*\*

1. The change operator works in the same way as delete. The format is:

c [number] motion

2. The motions are the same, such as w (word) and $ (end of line).

3. Move to the first line below marked --->.

4. Move the cursor to the first error.

5. Type c$ and type the rest of the line like the second and press <ESC>.

---> The end of this line needs to be corrected using the c$ command.

---> The end of this line needs to be corrected using the c$ command.

NOTE: You can use the Backspace key to correct mistakes while typing.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 3 SUMMARY

1. To put back text that has just been deleted, type p . This puts the

deleted text AFTER the cursor (if a line was deleted it will go on the

line below the cursor).

2. To replace the character under the cursor, type r and then the

character you want to have there.

3. The change operator allows you to change from the cursor to where the

motion takes you. eg. Type ce to change from the cursor to the end of

the word, c$ to change to the end of a line.

4. The format for change is:

c [number] motion

Now go on to the next lesson.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 4.1: CURSOR LOCATION AND FILE STATUS

\*\* Type CTRL-G to show your location in the file and the file status.

Type G to move to a line in the file. \*\*

NOTE: Read this entire lesson before executing any of the steps!!

1. Hold down the Ctrl key and press g . We call this CTRL-G.

A message will appear at the bottom of the page with the filename and the

position in the file. Remember the line number for Step 3.

NOTE: You may see the cursor position in the lower right corner of the screen

This happens when the 'ruler' option is set (see :help 'ruler' )

2. Press G to move you to the bottom of the file.

Type gg to move you to the start of the file.

3. Type the number of the line you were on and then G . This will

return you to the line you were on when you first pressed CTRL-G.

4. If you feel confident to do this, execute steps 1 through 3.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 4.2: THE SEARCH COMMAND

\*\* Type / followed by a phrase to search for the phrase. \*\*

1. In Normal mode type the / character. Notice that it and the cursor

appear at the bottom of the screen as with the : command.

2. Now type 'errroor' <ENTER>. This is the word you want to search for.

3. To search for the same phrase again, simply type n .

To search for the same phrase in the opposite direction, type N .

4. To search for a phrase in the backward direction, use ? instead of / .

5. To go back to where you came from press CTRL-O (Keep Ctrl down while

pressing the letter o). Repeat to go back further. CTRL-I goes forward.

---> "errroor" is not the way to spell error; errroor is an error.

NOTE: When the search reaches the end of the file it will continue at the

start, unless the 'wrapscan' option has been reset.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 4.3: MATCHING PARENTHESES SEARCH

\*\* Type % to find a matching ),], or } . \*\*

1. Place the cursor on any (, [, or { in the line below marked --->.

2. Now type the % character.

3. The cursor will move to the matching parenthesis or bracket.

4. Type % to move the cursor to the other matching bracket.

5. Move the cursor to another (,),[,],{ or } and see what % does.

---> This ( is a test line with ('s, ['s ] and {'s } in it. ))

NOTE: This is very useful in debugging a program with unmatched parentheses!

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 4.4: THE SUBSTITUTE COMMAND

\*\* Type :s/old/new/g to substitute 'new' for 'old'. \*\*

1. Move the cursor to the line below marked --->.

2. Type :s/thee/the <ENTER> . Note that this command only changes the

first occurrence of "thee" in the line.

3. Now type :s/thee/the/g . Adding the g flag means to substitute

globally in the line, change all occurrences of "thee" in the line.

---> the best time to see the flowers is in the spring.

4. To change every occurrence of a character string between two lines,

type :#,#s/old/new/g where #,# are the line numbers of the range

of lines where the substitution is to be done.

Type :%s/old/new/g to change every occurrence in the whole file.

Type :%s/old/new/gc to find every occurrence in the whole file,

with a prompt whether to substitute or not.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Lesson 4 SUMMARY

1. CTRL-G displays your location in the file and the file status.

G moves to the end of the file.

number G moves to that line number.

gg moves to the first line.

2. Typing / followed by a phrase searches FORWARD for the phrase.

Typing ? followed by a phrase searches BACKWARD for the phrase.

After a search type n to find the next occurrence in the same direction

or N to search in the opposite direction.

CTRL-O takes you back to older positions, CTRL-I to newer positions.

3. Typing % while the cursor is on a (,),[,],{, or } goes to its match.

4. To substitute new for the first old in a line type :s/old/new

To substitute new for all 'old's on a line type :s/old/new/g

To substitute phrases between two line #'s type :#,#s/old/new/g

To substitute all occurrences in the file type :%s/old/new/g

To ask for confirmation each time add 'c' :%s/old/new/gc

Review CH2

1. **Briefly, what does the process of installing an operating system such as Linux involve?**

**A:** Installing Linux is the process of copying operating system files from an installation medium ( e.g., DVD, USB flash drive) or the network to hard disk(s) on a system and setting up configuration files so Linux runs properly on the hardware.

2. **What is an installer? What is the name of the Fedora/RHEL installer?**

**A:** The installer is a tool that automates the process of installing Linux and makes the installation process easier and friendlier. The name of Fedora/RHEL installer is Anaconda.

3. **Would you set up a GUI on a server system? Why or why not?**

**A:** No because server systems can save resources by not including a GUI.

4. **A system boots from the hard disk. To install Linux, you need it to boot from a DVD. How can you make the system boot from a DVD?**

**A:** The BIOS determines the order in which the system tries to boot from each device. Hence, as the system boots, go into the BIOS setup and change the order of the devices the system tries to boot from. Revise the order so that the system first tries to boot from the DVD and then tries to boot from the hard disk.

5. **What is free space on a hard disk? What is a filesystem?**

**A:** The area of a partitioned dis that is not occupied by partitions is called free space.

A filesystem is a way to organize and save data for a program after it is shut down. It needs to be written to the partition.

6. **What is an ISO image? How do you burn an ISO image to a DVD?**

**A:** An installation (ISO) image is an exact image of what needs to be on the DVD. When you burn an ISO image to a DVD, you must use a special command that is part of most DVD-writing software; you cannot copy an ISO image to a DVD the same way you copy other files. The special command has a label similar to Record CD from CD Image or Burn CD Image. Refer to the instructions for the software you are using for information on how to burn an ISO image file to a CD/DVD.

### Advanced Exercises

**7. Give two reasons why RAID cannot replace backups.**

**A:** Firstly, if the system experiences a catastrophic failure, RAID is useless. Earthquake, fire, theft, and other disasters might leave the entire system inaccessible (if the hard disks are destroyed or missing). Secondly, RAID also does not take care of the simple case of replacing a file when a user deletes it by accident.

**8. What are RAM disks? How are they used during installation?**

**A:** A RAM disk is random access (system) memory that uses in place of the hard disk used for a normal boot operation. The installer copies tools required for the installation or to bring up a system from a live CD or an install DVD to the RAM disks. The use of RAM disks allows the installation process to run through the specification and design phases without writing to the hard disk and enables you to opt out of the installation at any point before the last step of the installation. The RAM disks also allow a live session to leave the hard disk untouched.

**9. What is SHA2? How does it work to ensure that an ISO image file you download is correct?**

**A:** SHA2 is Secure Hash Algorithm 2. The third member of the SHA family, SHA2 is a set of four cryptographic hash functions named SHA-224, SHA-256, SHA-384, SHA-512 with digests that are 224, 256, 384, and 512 bits, respectively.

The CHECKSUM file contains the SHA2 sums for each of the available installation (ISO) image files. When you process a file using the sha256sum utility, sha256sum generates a number based on the file. If that number matches the corresponding number in the CHECKSUM file, the downloaded file is correct. With the –c option and the name of the CHECKSUM file, sha256sum checks each of the files listed in the CHECKSUM file.

**10. If the SHA256 checksums of an ISO don't match what could some reasons why?**

A: If the two codes don't match then it means that the ISO file has changed in some way, most likely due to being corrupted. Because sha256sum generates a number based on the file and ISO file also come from the file, these two codes should be the same.

Review CH3

1**. What is a live system? What advantages does it have over an installed system?**

**A:** A live session is a Linux session that you run on a computer without installing Linux on the computer. The advantage is it gives you a chance to preview this software without installing it. After you use it; the main computer is unaffected or changed by a live system.

**2. Describe the Anaconda installer.**

**A:** Anaconda is written in Python and C. It identifies the hardware, loads drivers, probes for the devices it will use during installation, builds the necessary filesystems, starts the X server, and installs the Fedora/RHEL operating system. Anaconda can run in graphical interactive mode (default), limited textual system or in automated mode (Kickstart).

**3. Where can the installation image used by Anaconda be located?**

**A:** Auto-detected installation media or a location (URL) on the network.

**4. Why is it important to test the installation medium? How can you do so?**

**A:** It is important to verify the integrity of a downloaded image to ensure that it will be functional and that it has not been tampered with. You can test the installation medium by selecting the Test this media in the Boot menu, by clicking Verify in the Software/Installation Source screen during installation, or by using manually using sha256sum before installation.

**5. What should you do if the graphical installer does not work?**

**A:** On some hardware, the installation might pause for as long as ten minutes. Before experimenting with other fixes, try waiting for a while. If the installation hangs, try booting with one or more of the boot parameters described in this section.

**6. When might you specify an ext2 filesystem instead of ext4?**

**A:** Use ext2 for partitions whose data does not change often, such as /boot. The added overhead of the ext4 journal offers no benefit on these filesystems.

**7. Which utilities can you use to partition a hard disk prior to installation?**

**A:** The GNOME disk utility.

**8. What do you need to do before you can install Fedora as the second operating system on a Windows machine (to create a dual-boot system)?**

**A:** You need to back up important data and find disk space, create free space on the disk to install Fedora. You can create free space by deleting or shrinking partitions.

**9. How does Anaconda set up a hard disk by default?**

**A:** Anaconda, by default, uses LVM to set up most of the hard disk, creating LVs (logical volumes) instead of partitions. It places /boot on the first partition on the drive, not under the control of LVM. LVM creates a VG (volume group) named fedora that occupies the rest of the disk space. Within this VG it creates two or three LVs: root (/,fedora\_root), swap (fedora\_swap), and if there is room, /home (fedora\_home).

**10. How would you turn off DMA (direct memory access) for all disk controllers when you install a new system?**

**A:** You need to specify the nodma boot parameter as you boot the system. To specify a boot parameter, you must interrupt the automatic boot process by pressing the SPACE bar while the system is counting down when you first boot the system. When you press the SPACE bar, Fedora displays the Fedora Boot menu. Use the ARROW keys to highlight the selection you want before proceeding. With the desired selection highlighted, press the TAB key to display the boot command-line parameters. Enter a SPACE followed by nodma and press RETURN to boot the system.

Review CH4

**1. The system displays the following message when you attempt to log in with an incorrect username or an incorrect password:**

**Login incorrect**

1. **This message does not indicate whether your username, your password, or both are invalid. Why does it not reveal this information?**

**A:** It does not differentiate between an unacceptable username and an unacceptable password — a strategy meant to discourage unauthorized people from guessing names and passwords to gain access to the system.

1. **Why does the system wait for a couple of seconds to respond after you supply an incorrect username or password?**

**A:** It isa security feature to stop you to run a script and guess the password.

**2. Give three examples of poor password choices. What is wrong with each?**

**A:** Examples of poor password choices follow:

1. **mother** a word from a dictionary
2. **hong** user’s name
3. **9u** too short

**3. Is fido an acceptable password? Give several reasons why or why not.**

**A:** No, fido is not an acceptable password.

Reasons:

a. a word from a dictionary;

b. less than 8 characters;

c. not contain any nonalphanumeric character;

d. no capital letters.

**4. What is a context menu? How does a context menu differ from other menus?**

**A:** A context menu has choices that apply specifically to the window or object you click. The choices differ from window to window and from object to object. Some windows do not have context menus. Frequently a right-click displays a context menu.

**5. What is a workspace? What is the relationship between a workspace and the desktop?**

**A:** A workspace is a screen that holds windows of one or more applications. The Activities Overview screen and the Application Switcher enable you to display any of the running applications and its workplace. The desktop, which is not displayed all at one, is the collection of all workspaces.

**6. How would you swap the effects of the right and left buttons on a mouse? What is the double-click speed? How would you change it?**

**A:** The Mouse & Touchpad window enables you to change a mouse from right-handed to left-handed, or vice versa. The double-click speed specifies the speed with which you must double-click a mouse button to have the system recognize your action as a double click rather than as two single clicks. You can change this characteristic by using the Mouse & Touchpad window.

**7. What is an Application menu? What does it allow you to do?**

**A:** Many applications have an application menu that allows you to set preferences and get help with the applications. While the application is active (has the focus), click the object (the name of the window) for the application on the Top panel; GNOME opens the Application menu for the application.

**8. What is Nautilus? What does it allow you to do?**

**A:** Nautilus is the GNOME file manager. You can use it to copy, move, open, and execute files.

**9. Describe three ways to**

**a. Change the size of a window.**

**A:** a. you can double-click the titlebar to maximize and restore a window.

b. Window Operations menu allows you to minimize, maximize or resize a window.

c. To resize a window, position the mouse pointer over an edge of the window; the pointer turns into an arrow pointing to a line. When the pointer is an arrow pointing to a line, you can click and drag the side of a window. When you position the mouse pointer over a corner of the window, you can resize both the height and the width of the window simultaneously.

**b. Delete a window.**

**A:** a. clicking the close button (X) closes the window and usually terminates the program running in it.

b. Window Operations menu allows you to close a window.

c. using Menubar to choose the option to close a window.

**10. What are the functions of a Window Operations menu? How do you display this menu?**

**A:** Right-clicking the window titlebar displays the Window Operations menu.

This menu allows you to move, resize, close, and otherwise manipulate a window.

**11. How would you set the system time manually?**

**A:** Click Date & Time in the setting window or click the time at the top of the screen and select Date & Time Settings.

**12. While working on the command line you get an error that includes the words missing destination file operand. What would be a good way to get help figuring out what the problem is?**

**A:** You can use the GNOME help window, the internet or help command to get help.

**13. How does the mouse pointer change when you move it to the edge of a window? What happens when you left-click and drag the mouse pointer when it looks like this? Repeat this experiment with the mouse pointer at the corner of a window.**

**A:** The mouse pointer changes to a double-headed arrow. When you drag this arrow, you can resize the window. When you position the pointer on an edge of the window, you can resize the window in one direction. When you position the pointer on a corner, you can resize in both directions at once. When I left-click and drag the mouse pointer, it appears a rectangle and choose everything in this rectangle.

**14. What is the Session menu? When and where does it appear? How do you use it?**

**A:** A session starts when you log in and ends when you log out or reset the session. With fully GNOME-compliant applications, GNOME can manage sessions. The Session menu is the Startup Applications Preferences window. When you start to run Linux and log in, it will appear. To open the Startup Applications Preferences window give the command gnome-session-properties from a terminal emulator or Run Application window (ALT-F2). You must give this command while logged in as yourself (not while working with root privileges). To save your sessions automatically when you log out, click the Options tab in the Startup Applications Preferences window and put a tick in the check box labeled Automatically remember running applications when logging out.

**15. What happens when you run nano from the Enter a Command window? Where does the output go? (2 points)**

**A:** When you run nano in this manner, the output is lost.

**16. The example on page 130 shows that the man pages for crontab appear in Sections 1 and 5 of the system manual. Explain how you can use man to determine which sections of the system manual contain a manual page with a given name. (2 points)**

**A:** Use man with the –f option (equivalent to whatis) to list man entries for the argument that follows this option. Alternatively, use man with the –a option to display all of the entries for the argument that follows this option. Different Sections present different file type and name.

**17. How many man pages are in the Devices subsection of the system manual? (Hint: Devices is a subsection of Special Files.) (2 points)**

**A:** Approximately 60. The following command shows exactly how many man pages are in the Devices subsection of the system manual:

$ ls /usr/share/man/man4 | wc –l

Review CH6

1. Is each of the following an absolute pathname, a relative pathname, or a simple filename?

**a. milk\_co**

**b. correspond/business/milk\_co**

**c. /home/max**

**d. /home/max/literature/promo**

**e. ..**

**f. letter.0210**

A: a = Simple Filename

b = relative Pathname

c = absolute Pathname

d = absolute Pathname

e = Hidden Filename (is a relative path name as it means go up 1 level.)

f = Simple Filename

2. List the commands you can use to perform these operations:

a. Make your home directory the working directory

b. Identify the working directory

A: a. cd; (cd~)

b. pwd

3. If the working directory is **/home/max** with a subdirectory named **literature**, give three sets of commands you can use to create a subdirectory named **classics** under **literature**. Also give several sets of commands you can use to remove the **classics** directory and its contents.

A: a = mkdir /home/max/literature/classics,

mkdir ~max/literature/classics,

or mkdir ~/literature/classics

b = rmdir /home/max/literature/classics,

rmdir ~max/literature/classics,

rmdir ~/literature/classics,

or rm -r /home/max/literature/classics

4. The **df** utility displays all mounted filesystems along with information about each. Use the df utility with the **-h** (human-readable) option to answer the following questions.

a. How many filesystems are mounted on your Linux system?

b. Which filesystem stores your home directory?

c. Assuming your answer to exercise 4a is two or more, attempt to create a hard link to a file on another filesystem. What error message do you get? What happens when you attempt to create a symbolic link to the file instead?

A: a. six;

b. /dev/shm; (/dev/mapper/fedora-root)

c. Failed to create hard link `/home/Destop' => `draft': Permission denied;

Failed to create symbolic link `/home/Destop': Permission denied

5. Suppose you have a file that is linked to a file owned by another user. How can you ensure that changes to the file are no longer shared?

A: Make a local copy of the file, not linked to the copy owned by the other user.

6. You should have read permission for the **/etc/passwd** file. To answer the following questions, use cat or less to display **/etc/passwd**. Look at the fields of information in **/etc/passwd** for the users on the local system.

a. Which character is used to separate fields in **/etc/passwd**?

b. How many fields are used to describe each user?

c. How many users are on the local system?

d. How many different login shells are in use on your system? (\_Hint:\_ Look at the last field.)

e. The second field of **/etc/passwd** stores user passwords in encoded form. If the password field contains an **x**, your system uses shadow passwords and stores the encoded passwords elsewhere. Does your system use shadow passwords?

A: a. colon (:);

b. seven;

c. three: root, aplustudent, hongzhang;

d. five;

e. yes.

7. If **/home/zach/draft** and **/home/max/letter** are links to the same file and the following sequence of events occurs, what will be the date in the opening of the letter?

a. Max gives the command **vim letter**.

b. Zach gives the command **vim draft**.

c. Zach changes the date in the opening of the letter to January 31, writes the file, and exits from vim.

d. Max changes the date to February 1, writes the file, and exits from vim.

A: The date will be February 1 because Max is the last person who write the file.

8. Suppose a user belongs to a group that has all permissions on a file named **jobs\_list**, but the user, as the owner of the file, has no permissions. Describe which operations, if any, the user/owner can perform on **jobs\_list**. Which command can the user/owner give that will grant the user/owner all permissions on the file?

A: Using **ls** command can list this file.

The **chmod** command lets you change the file permission.

9. Does the root directory have any subdirectories you cannot search as an ordinary user? Does the root directory have any subdirectories you cannot read as a regular user? Explain.

A: Yes, as an ordinary user, you cannot search or read some subdirectories in the root directory, such as the executable files that are setuid and owned by root because they have root privileges when they run, even if they are not run by root. This type of program is very powerful because it can do anything that root can do (and that the program is designed to do). Because of the power they hold and their potential for destruction, many sites minimize the use of these programs on their systems and ordinary users cannot search or read them.

10. Assume you are given the directory structure shown in Figure 6-2 on page 177 and the following directory permissions:

d--x--x--- 3 zach pubs 512 2010-03-10 15:16 business

drwxr-xr-x 2 zach pubs 512 2010-03-10 15:16 business/milk\_co

For each category of permissions--owner, group, and other--what happens when you run each of the following commands? Assume the working directory is the parent of correspond and that the file cheese\_co is readable by everyone.

a. **cd correspond/business/milk\_co**

b. **ls –l correspond/business**

c. **cat correspond/business/cheese\_co**

A: a. owner: OK; group: OK; other: Permission denied

b. owner, group, and other: Permission denied

c. owner and group: Is a directory; other: Permission denied

11. What is an inode? What happens to the inode when you move a file within a filesystem?

A: An inode is the control structure for a file. Each directory entry associates a filename with an inode. Although a single file might have several filenames (one for each link), it has only one inode. If the two filenames have the same inode number, they share the same control structure and are links to the same file. Hence, when you move a file within a filesyste, the inode will not change.

12. What does the .. entry in a directory point to? What does this entry point to in the root (/) directory?

A: The .. entry is a link to the parent directory. If it is in the root directory, there is no parent, and the .. entry is a link to the root directory itself.

13. How can you create a file named -i? Which techniques do not work, and why do they not work? How can you remove the file named -i?

A: the command to create a file: mkdir ./-i

Because this file likes a command line argument, it does not be easy to remove.

The command to remove this file: rm ./-i

14. Suppose the working directory contains a single file named **andor**. What error message do you get when you run the following command line?

**$ mv andor and\/or**

Under what circumstances is it possible to run the command without producing an error?

A: The error message: **mv: cannot move 'andor' to 'and/or': No such file or directory**

The solution is you should create a file named “and”. The command will be:

**$ mkdir and**

**$ mv andor and\/or**

**$ ls and**

**or**

15. The **ls -i** command displays a filename preceded by the inode number of the file (page 206). Write a command to output inode/filename pairs for the files in the working directory, sorted by inode number. (\_Hint:\_ Use a pipeline.)

A: Command: **ls -i | sort –i**

16. Do you think the system administrator has access to a program that can decode user passwords? Why or why not? (See exercise 6.)

A: Normally, the system administrator cannot decode user passwords. The administrator can assign a new password to a user. Passwords are generally encrypted by a one-way hash so the system can tell when the correct password is entered, but it cannot regenerate the cleartext password. The system applies the hash algorithm to the entered password and checks whether the result matches the stored, encrypted password. A match means the correct password was entered. However, if a user has a weak password, the system administrator can use a program such as crack or John the Ripper to decode a password. I can download either of these utilities using yum. The packages are named crack and john.

17. Is it possible to distinguish a file from a hard link to a file? That is, given a filename, can you tell whether it was created using an **ln** command? Explain.

A: No. All links to a file are of equal value: The operating system cannot distinguish the order in which multiple links were created. When a file has two links, you can remove either one and still access the file through the remaining link. You can remove the link used to create the file, for example, and as long as one link remains, still access the file through that link.

18. Explain the error messages displayed in the following sequence of commands:

**$ ls -l**

total 1

drwxrwxr-x. 2 max pubs 1024 03-02 17:57 dirtmp

**$ ls dirtmp**

**$ rmdir dirtmp**

rmdir: dirtmp: Directory not empty

**$ rm dirtmp/\***

rm: No match.

A: There is a file whose name begins with a period (a hidden file) in the

dirtmp directory. Use ls with the –a option to list the file. Remove the file, and then you will be able to remove the directory.

19. Name and describe the 3 P's of linux problems?

A: All problems in Linux can be described by the 3 P’s which is Path, Permission and dependencies.

Review Ch7

**1. Which commands can you use to determine who is logged in on a specific terminal?**

who or w

**Which command shows you which terminal you are giving the command from?**

w

**2. Give a command that displays a long listing of the files in /bin in reverse chronological order.**

$ ls –ltr /bin

**Give the command again but this time display the output one screen at a time.**

$ ls –ltr /bin | less

**3. What happens when you give the following commands if the file named done already exists?**

**$ cp to\_do done:** this command copy the file done to to\_do

**$ mv to\_do done:** this command rename the file done to to\_do

Either command overwrites done with the contents of to\_do.

**4. List the first 20 lines in /etc/services that describe TCP ports.**

$ grep tcp /etc/services | head -20

5. **How can you find the phone number for Ace Electronics in a file named phone that contains a list of names and phone numbers?**

$ grep "Ace Electronics" phone

**Which command can you use to display the entire file in alphabetical order?**

$ sort phone

**How can you display the file without any adjacent duplicate lines?**

$ uniq phone

**How can you display the file without any duplicate lines?**

$ sort –u phone

**6. What happens when you use diff to compare two binary files that are not identical? (You can use gzip to create the binary files.) Explain why the diff output for binary files is different from the diff output for ASCII files.**

When the files are different, diff shows a message that files differ; when the files are the same, there is no message. The diff utility compares ASCII files on a line-by-line basis; it is not designed to compare binary files on a byte-by-byte basis. Use cmp to compare binary files.

**7. What is the result of giving the which utility the name of a command that resides in a directory that is not in your search path?**

The which utility displays a message saying that the command you are looking for is not in the list of directories in your search path. For example,

$ which me

/usr/bin/which: no me in (/usr/lib64/qt -3.3/bin: /usr/local/sbin: /usr/bin: /usr/sbin: /bin: /sbin: /home/aplustudent/ .local/bin: /home/aplustudent/bin)

**8. Are any of the utilities discussed in this chapter located in more than one directory on the local system? If so, which ones?**

No. However, some commands which are built into a shell have counterparts that exits as executable file, such as echo.

**9. Experiment by calling the file utility with the names of files in /usr/bin. How many different types of files are there?**

Approximately 20.

10. **Which command can you use to look at the first few lines of a file named status.report?**

head ($ head status.report)

**Which command can you use to look at the end of the file?**

tail ($ tail status.report)

**11. Display only the names of the last ten services listed in /etc/services. (Hint: the services are listed in the first column.)**

$ cut –f1 /ect/services | tail

**12. Display a long listing of the files in the /etc/pam.d directory hierarchy that are links.**

$ find /etc/pam.d –type l | xargs ls -l

**13. Display a count of the number of words in the /etc/passwd file.**

$ wc –w /etc/passwd

**14. Display the /etc/passwd file, replacing all colons (:) with TABs.**

$ cat /etc/passwd | tr ‘:’ ‘\t’

**Display the /etc/services file, substituting one SPACE for each occurrence of multiple SPACEs.**

$ cat /etc/services | tr –s ‘ ’

15. **Change the modification time of /bin/bash to the system time. What happens? Why?**

$ touch /bin/bash

Touch: cannot touch ‘/bin/bash’: Permission denied because this file is created by root and cannot be changed.

**Copy /bin/bash to the working directory, display its modification time, and then change its modification time to the system time. Display the modification time of your copy of bash and the system time. Change its modification time to January 5, 2018 at 2:34 PM and display its modification time.**

$ cp /bin/bash bash1

$ ls –l bash1

$ touch bash1

$ ls –l bash1

$ touch –d ‘5 January 2018 14:34’ bash1

**16. Copy /bin/bash to the working directory and make two copies so you have three identical files: bash1, bash2, and bash3. Compress bash1 using gzip and bash2 using bzip2. Do not change bash3.**

$ gzip bash1

$ bzip2 bash2

$ ls -l bash\*

-rwxr-xr-x. 1 aplustudent aplustudent 491547 Jan 5 2018 bash1.gz

-rwxr-xr-x. 1 aplustudent aplustudent 466775 Feb 26 03:10 bash2.bz2

-rwxr-xr-x. 1 aplustudent aplustudent 1051464 Feb 26 03:10 bash2

**Which utility does the best job of compressing the file?**

bzip2

**Which does the worst?**

gzip

**How big is bash2.bz2 compared to bash3?**

Bash3 is bigger than bash2.bz2

**17. Re-create the colors.1 and colors.2 files used in the example on page 228. Test your files by running diff -u on them. Does diff display the same results as in the figure?**

-u option doesn't display any redundant information. It will show pink, purple and blue.

**18. Try giving these two commands:**

**$ echo cat**

**$ cat echo**

**Explain the differences between the output of each command.**

$ echo cat: cause echo to display the characters c, a and t.

$ cat echo: use cat to copy the contents of a file named echo. If there is no file named echo, it will display an error message.

**19. Repeat exercise 5 using the file phone.gz, a compressed version of the list of names and phone numbers. Consider more than one approach to answer each question and explain how you made your choices.**

First option: use gunzip to decompress the file. Then, use the same commands as exercise 5.

Second option: use zcat and a pipe to display the results without creating an intermediate file.

$ zcat phone.gz | grep "Ace Electronics"

$ zcat phone.gz | sort

$ zcat phone.gz | uniq

$ zcat phone.gz | sort -u

The different depends on the size of the file. If phone.gz is large, the first option will be faster because more size means you should use more CPU (processor) time. If it is small, the second one will use less hard disk storage, or vice versa.

**20. Find or create files that**

**a. gzip compresses by more than 80 percent.**

The gzip utility compresses most text files by more than 80 percent.

**b. gzip compresses by less than 10 percent.**

The gzip utility compresses most files that are already compressed. Such as jpeg files, by less than 10 percent.

**c. Get larger when compressed with gzip.**

The gzip utility expands a file that has already been compressed with gzip. to compress a gzipped file a second time, you shoud remove the .gz filename extension.

**d. Use ls -l to determine the sizes of the files in question. Can you characterize the files in a, b, and c?**

Files with repeated information or inefficiently stored information can be compressed the most. Files that have been compressed already store information efficiently and can be compressed only a small amount, not at all, or expanded.