**05 – Basix Unix System Administration**

**Activities**

COMP190 – Tools and Techniques for Software Development

Dickinson College

Fall 2021

**Name:**

Over the last several classes you have learned to navigate the Unix/Linux command line interface and to use a number of the common tools and filters. In this set of activities, you will again use all of those skills and add to them some new ones that allow you to manage a Linux installation. You will be learning about administrative privileges, creating new users and groups, installing software and learning a few new commands along the way. You’ll even install and setup a web server before you are done.

**VirtualBox Snaphots:**

Virtual box provides a mechanism for taking snapshots of the current machine state. A snapshot will allow you to return to a known machine configuration at any time. This can be extremely helpful when you are updating or reconfiguring a system or installing new software. Sometimes these tasks won’t go quite as planned. Your system may become unstable, completely unusable or maybe just littered with the remnants of everything you tried that didn’t work. Snapshots provide a way to protect yourself from these situations.

Before doing any significant work on the system you should always take a snapshot. That way if things do go sideways, you won’t have to reinstall everything from scratch.

1. Clean up any of the temporary file and directories that we created in the earlier activities (e.g. Test1, File3, ect…) and then create a snapshot of your machine configuration. Note: You do not have to shutdown the virtual machine to take a snapshot. Name the snapshot using your name. Then include a screen shot of the VirtualBox Manager window showing the snapshot that you created as your answer to this question.

2. Experiment with the snapshot feature a little. Make some changes to your virtual machine by creating a few new directories and files. Shutdown your virtual machine, restore the snapshot part 1 and then restart the virtual machine.

a. Briefly explain the steps you took to restore the snapshot?

b. Do the new directories and files you created exist in the snapshot? Why or why not?

**The Super User:**

Because Unix/Linux is a multiuser operating system it is necessary to ensure that not every user is able to modify the system configuration. In Unix, the user named root is the *super user* and has full administrative privileges to the machine. That is root may perform any operation on any file anywhere on the system. However, even when doing tasks that require administrative privileges you will not typically operate as root. You will instead use the sudo command that grants you temporary administrative privileges on a command by command basis.

To see some examples of the sudo command and to get a feel for how it is used you can watch the video *Linux Commands for Beginners (Old Version): 07 – Sudo* by Jay from LearnLinuxTV:

* <https://www.youtube.com/watch?v=Xi-jwoxoTKg> (5:36)

3. Let’s first confirm that sudo does in fact run commands as the super user.

a. What does the whoami command do? The description on the man page is a good source.

b. What is the output of the whoami command when you run it?

c. Now run whoami using sudo. Give the command that you used. What is the output now? Why?

4. Because sudo runs commands as root it will allow you to do anything to any file anywhere on the system. This comes with awesome power that should be used sparingly. So, let’s walk through an example seeing when we need to use sudo. We’ll do more useful things with it shortly.

The /etc/group file lists all of the user groups on the machine and identifies which users are in which groups.

a. What permissions do you have to the /etc/group file? Hint: Use ls -l /etc/group. Then recall that permissions (e.g. rw-r--r--) have three parts, owner, group and other. Notice in the ls output that root is both the owner and group for this file. So it is the “other” permissions apply to your user.

b. Because you have read permission to the /etc/group file you can look at its contents as your normal user. Give a grep command that will display all lines from the file /etc/group that contain your username.

c. Run your command from b. To which groups do you belong?

The /etc/sudoers file contains information about which users have sudo privileges.

d. Try to use cat to display the contents of the /etc/sudoers file. What happens?

e. What permissions do you have to the /etc/sudoers file? How did you determine this?

f. Use sudo to display the /etc/sudoers file. What is the full command you used?

g. Given your above answers, briefly explain why you are allowed to use the sudo command while others may not be permitted to?

It is not required viewing, but if you are interested in a little more background on why it is preferred to use sudo instead of working as the root user check out the video *Why “sudo”?* from the BeginLinux Guru:

* <https://www.youtube.com/watch?v=Q5qMY1elbyE> (8:37)

**Managing Users and Groups:**

One of the common tasks, at least on multiuser Linux systems, is creating new users. Even on your own system it can sometimes be useful to have multiple accounts (i.e. different logins). For example, you might want to have one that doesn’t have sudo privileges for everyday tasks and another that does for doing system administration work. With larger software projects it is also not unusual to have specific accounts just for running the associated programs (e.g. did you notice that there is a user named vboxsf?) In the next few questions you’ll create and manage a new user.

5. The adduser command can be used to add a new user to the system. It is pretty easy to use in its basic form, which is: adduser <username>

a. Give a command (not using sudo) for adding a new user to your system. Your new user can have whatever name you want.

b. What happens when you run the command? Why?

c. Now give a version of your command using sudo.

d. Run your command from c and respond to the prompts provided by the adduser command. If all went well this will have created a new user with the name that you specified.

i. List the contents of the /home directory. What do you notice?

ii. What is the last line in the /etc/passwd file?

iii. Use the command su <username>, enter the password you assigned for the new user. Then, run the whoami command. What does the su command do?

iv. Does the new user have sudo privileges? How did you check?

v. If you su to the new user and then use the command exit, what happens?

6. The addgroup command can be used both to create new groups and to add users to existing groups. Use the command addgroup --help to answer the following questions.

a. Give a command that creates a new group named student. Run it and check that it worked by looking at the /etc/group file.

b. Give a command that adds your new user from above to the student group. Run it and check that it worked by looking at the /etc/group file. Note: There is also a groups command that you can use to check this. Use man groups if you want to learn more.

c. Explain how you would give this new user administrative privileges (i.e. allow them to use sudo) if you wanted to.

**Software installation:**

While LinuxLite comes with a lot of good software, anytime you are doing any serious development work you are going to need to regularly install new software and update old software. This section looks at package managers, which are the primary mechanisms by which software is installed and updated in Linux.

*A GUI Package Manager*

The Synaptic Package Manger is a graphical user interface for the Advanced Package Tool (apt) that is used by many Linux distributions for installing and updating software. In this section, you’ll use the Synaptic Package Manager to install some software and then in the next, you’ll get some experience with apt directly from the command line.

Start by watching the video *Using the Synaptic Package Manager* by Joe Collins. You can stop after about three minutes when he starts talking about the Software Center, that is Ubuntu specific and does not apply to LinuxLite or other distros.

* <https://www.youtube.com/watch?v=A2beSmnO_lk> (3:00)

7. Launch the Synaptic Package Manager in LinuxLite.

a. Where did you find it?

b. Did it ask you to enter your password? Why?

8. Find the htop program that Joe mentioned in Synaptic.

a. Is it already installed? If so, what version is installed? If not install it.

b. What does htop do? Hint: you can read about it right in the package manager.

c. The htop program can be pretty useful. It will tell you how busy your CPU is, how much memory is being used and which processes are using the most resources. It will even let you kill processes without first finding their PID as we did before. Run the htop program from the command line and use it to answer the following questions.

i. How are the processes sorted by default? (i.e. which processes are shown at the top of the list?) Hint: Look at the light blue column header.

ii. Imagine instead we want to see which processes are using the most memory. Change htop so that the list is sorted by the percentage of the memory that a program is using. Include a screenshot of the terminal window running htop as the answer for this question.

9. Joe’s video shows you where to find the repositories used by the package manager.

a. How does he define repositories?

b. You may have noticed in the “Sections” panel that a lot of sections are listed multiple times (see below), usually once with just the name and once with (universe) beside it. Use the “Repositories” dialog that is pointed out by Jay (on the “Settings” menu) to answer the questions below.

A screenshot of a cell phone

Description automatically generated

i. The section listing without any ( ) beside it corresponds to (main). What does (main) mean?

ii. Who or what is “canonical” and what is their relationship with Ubuntu? Use your favorite search engine to answer this one.

iii. What does (universe) mean?

Note that there are also several other categories of repositories (restricted) and (multiverse). These are non-FOSS (not Free and Open Source) and thus are in conflict with a commitment to FOSS. The (restricted) category contains essential software that is needed by particular types of hardware. So even though this software is not FOSS, the manufacturers provide it for free, and these repositories are included by default. The (multiverse) is not included by default because these repositories contain non-essential software and excluding them is more consistent with a commitment to FOSS.

10. We know that LinuxLite comes with firefox installed but I’m sure some of us prefer Google Chrome. So, let’s install it.

a. Find the package for the Google Chrome Browser. What is the name of the package for this browser? Note: There are multiple packages with “chrome” in their names. Use the Description to be sure you have the one for the browser.

b. What dependencies does Chrome have that will also be installed?

c. Install the Chrome browser. Run the Chrome browser and the bring up the “About Chrome” window from the “Help” menu under the in the upper right corner. Make the window a reasonably small size where you can still see the version number and include a screenshot of that window as your answer for this question.

Congratulations! You have just installed your first piece of Linux software using a package manager.

*The APT Command Line Interface (CLI):*

The GUI interface provided by the Synaptic Package Manager is nice, but sometimes it is going to be more convenient or efficient to work from the command line. In addition, having a command line interface makes it possible to automate common installation or update operations using other tools or scripts. In this section, you’ll use the command line apt tool to install some software.

Start by watching the video *Package Management Basics* (6:59) from LinkedIn Learning to see the basics of how a package manager works.

<https://www.linkedin.com/learning/linux-tips-weekly/package-management-basics>

You can refer back to this video as you complete the exercises below.

11. First a few gotcha’s that you’ll want to be aware of so that they don’t trip you up later.

a. What happens if you try to run apt install without using sudo?

b. Open the Synaptic Package Manager and then try using apt install (with sudo). What happens?

Both of these gotcha’s say something about a “lock” file. The package manager uses a lock file so that it knows that only one process at a time is trying to change the configuration. Otherwise they may make incompatible changes. So, if you receive a message that says something about a “lock” file when using apt it is likely one of these two gotchas. Either you forgot to use sudo, in which case the messages will contain “Permission denied.” Or, the Synaptic Package Manager is also running, in which case it will say something about the lock file being held by another process. Both are easy to fix once you recognize them.

12. The first step in using apt should always be to run an update.

a. What does apt update do? (Hint: it is discussed in the video).

b. Give a screenshot to showing the list of all of the repositories that are checked when you do the update.

13. LinuxLite comes with a Java Runtime Environment (JRE) installed but without a Java Development Kit (JDK) installed. This means that it is possible to run Java programs on LinuxLite, but it is not possible to compile them. At least not yet…

a. Run the command java -version. Which version of the Java JRE is installed?

b. The Java compiler is a part of the JDK and is usually run with the command javac. What happens when you run the command javac -version?

c. Notice when you used the command in part b you were given a number of suggestions for how to install the JDK. We could just use one of those but which one? Let’s be a little careful in how we choose and also gain some practice with other apt commands.

i. Use the search feature of apt to find a list of packages for the “OpenJDK Development Kit”. What command did you use?

ii. The last part of the filename of the packages will tell you what specifically the packages contain. For example, those ending in -doc contain documentation and those ending in -source contain the source code (yep the OpenJDK is FOSS). You should also see a bunch of names like openjdk-??-jdk. These are the ones that contain the standard Java Development Kit. The ?? indicates the version of the JDK. What is the package name for the newest (i.e. highest version number) OpenJDK?

iii. Use the show feature of apt to learn more about the package you identified in part ii. On what other packages does the OpenJDK package you identified depend?

iv. Use apt to install the most recent OpenJDK. Give the full command you used to install this package.

d. Now what happens when you run the command javac -version?

e. Finally, the list feature of apt can be used to check if a specific package is installed. Show the output an apt command that shows that the OpenJDK package is installed.

14. The apt is also good for keeping your software up to date with more recent versions.

a. What command is used to check and keep your software up to date? (Hint: It’s near the end of the video).

b. When you run the command from part a, how many packages need to be upgraded? Whoa! Thank goodness for package managers… I wouldn’t want to do that manually! If you have a little time (maybe 15 minutes) go ahead and perform the upgrade. Otherwise, you might do it later.

It is worth noting that while it seems like doing an upgrade often might be a good idea, it can be a mixed experience when working on a big project with lots of parts. Sometimes, the upgraded software and libraries are no longer compatible with other ones that are used by project and thus can cause issues that have to be resolved. So sometimes, when you are in the middle of a project is it best to let things be stable until the project is over. Then upgrade. Or at a minimum, take a snapshot before doing the upgrade so that if something goes wrong you can revert.

**Putting it All Together:**

Let’s imagine we are setting up a Linux box (what techies often call a computer running Linux) for a web development project involving a number of developers who will be creating web pages. For that we’ll need to have a web server installed, create accounts for the developers and make it possible for them to create web pages that are served up by the server locally for testing.

15. Currently there is no web server running on your LinuxLite. Confirm this by opening a web browser and entering http://localhost into the address bar. Like dickinson.edu is the name of the web server for Dickinson College, localhost is the name of the web server that is running on the local machine (i.e. the same one the browser is running on). Thus, using localhost tells the browser to look on your machine for a web server rather than going to the internet. What happens in the browser when you visit the URL http://localhost?

16. Apache (<https://httpd.apache.org/>) is a FOSS web server that has been the most popular server on the web for over 20 years. You invariably visit pages and apps served up by an Apache server many many times per day. We definitely want to go with the best, so let’s install Apache.

a. Install the Apache web server from the apache2 package. What is the full command that you used?

b. Let’s check that it worked… reload the page at localhost. You should see a different result. As it says, it is the default page served from Apache on your LinuxLite. Paste a small screenshot of your browser window showing this page as your answer for this question.

17. Let’s now create an account for a developer and set things up so that they can create a web page.

a. Pick a name for your developer and create a new account. What command did you use? Note: You created an account earlier!

b. There are different ways to configure a web server, but by default all of the web pages to be served must be contained within the /var/www/html directory. What permissions does your new user have to that directory? Hint: Use ls -l as we did earlier.

c. Clearly without write permission the developer will not be able to create pages there. We could give them write permission to this directory, but then if there are multiple developers they would all be able to change each other’s pages, which may be undesirable. Instead let’s create a new directory inside of /var/www/html and make your new user the owner of that directory.

a. Create a directory with the developer’s username inside the /var/www/html directory. What command did you use?

b. Who owns the directory that you created in part a? Why?

c. To allow the developer to control their own directory you need to make them the owner of the directory. The chown command is used to change owner of a file as follows:

chown <path to file>

Give a command that uses chown to make the developer the owner of their own directory in the /var/www/html.

d. Give the output of an ls -l command showing that the developer is now the owner of their own directory as your answer for this question.

18. Now let’s pretend you are the developer and see if it all works.

a. Logout of your account. Use either the “Log Out” or “Switch User” options under the “Menu” at the bottom left in LinuxLite.

A screenshot of a cell phone

Description automatically generated

b. Log in using the account you created for the developer.

c. Create a very simple web page:

i. Open an editor and enter the following HTML code – replacing Username with the name of your new developer:

<html>

<head>

<title>Test Page</title>

</head>

<body>

<h1>Hi I’m Username. It works!</h1>

</body>

</html>

ii. Save this page as /var/www/html/<username>/index.html

d. Open a browser and enter localhost/<username> into the address bar. Recall the < > are dropped when replacing <username>. Ta da! Paste a screen shot of this web page displayed in the browser as your answer to this question.

**Taking it to the Next Level:**

The following are optional extensions that you can try if you are interested.

19. If our web development project is a success there may be new developers coming on board all the time. That process of creating a new user, adding a sub-directory in /var/www/html, setting it owner and creating a www link in the developer’s home directory will get a little tedious. Write a bash script to do it! Notes:

* Your script should take the name of the new developer as a command line argument.
* You’ll want to be sure that your script checks that the user does not already exist before proceeding.
* You’ll need to learn about the flags of the adduser command so that you can fully script its operation.
* You’ll need to run the script using sudo.

20. Sometimes it is more convincing to see the web page load up not in the guest OS but in the host OS. That way it feels a little like it is actually coming from a web server.

a. Change the network adapter for your virtual machine to be a “bridged adapter” in the VirtualBox Manager (see below).

A screenshot of a cell phone

Description automatically generated

b. In the terminal on LinuxLite run the command ifconfig. This will give you information about the network interfaces on the machine. On the second line of the output will be the IP number for your guest machine. It will look something like: ﻿192.168.0.65

What is the IP address of your virtual machine?

c. In a browser on the host machine enter <IP Address>/<username> where the IP address is the IP address of your virtual machine and the username is the username of the developer. Ta Da! You should now see the page served up from Apache on your virtual machine to your host machine. Give a screenshot of the browser window showing the page using the IP address in the URL bar.

21. It is a little inconvenient for the developer to have to work in a sub-directory of /var/www/html and to remember that path all of the time. Like Windows and MacOS have shortcuts and aliases, Unix/Linux has something called a *link*. The ln command is used to create links. Read up on ln and create a symbolic (or soft) link named www in the developer’s home directory that points to /var/www/html/<username>. What command did you use?

22. Extend your bash script from question 19 to also create the linked www subdirectory from question 21. What commands did you add?

**Optional:** To help us improve and scope these activities for future semesters please consider providing the following feedback.

a. Approximately how much time did you spend on this activity outside of class time?

b. Please comment on any particular challenges you faced in completing this activity.

**Acknowledgements:**

Some materials, questions and resources have been adapted from activities posted on foss2serve.org.

* <http://foss2serve.org/index.php/Linux_Package_Management_(Distribute_Your_App)>
* <http://foss2serve.org/index.php/Fedora_Install_Activity>