Files

Introduction and/or Background

Linux has a whole suite of file tools available

Display

- df - We have explored this one before so we won't elaborate any further.
- du - Short for disk utilization. It reports the number of blocks used and
- the directory using it.
- ls - List structure. Displays the contents of a directory in various
- formats.
- cat - Displays the contents of a given file.
- head - Displays the top contents of a file.
- tail - Displays the bottom contents of a file. The -f option leaves the
- command live when new contents is appended to the file
- less - Like cat only the file can be navigated up or down.
- more - Like less only you can only navigate down the length of the file.

Manipulation

- cp - Copies a file to another named file or another directory.
- rm - Deletes a file from the system.
- mv - Moves a file and also can rename an existing file.
- mkdir - Creates a directory.
- rmdir - Removes a directory. At the command line all contents must be
- empty.
- touch - Create an empty file.

  Keep in mind, rm cannot remove a directory, must use rmdir. Also keep in mind that if you do not have the right permissions you cannot rm a file.

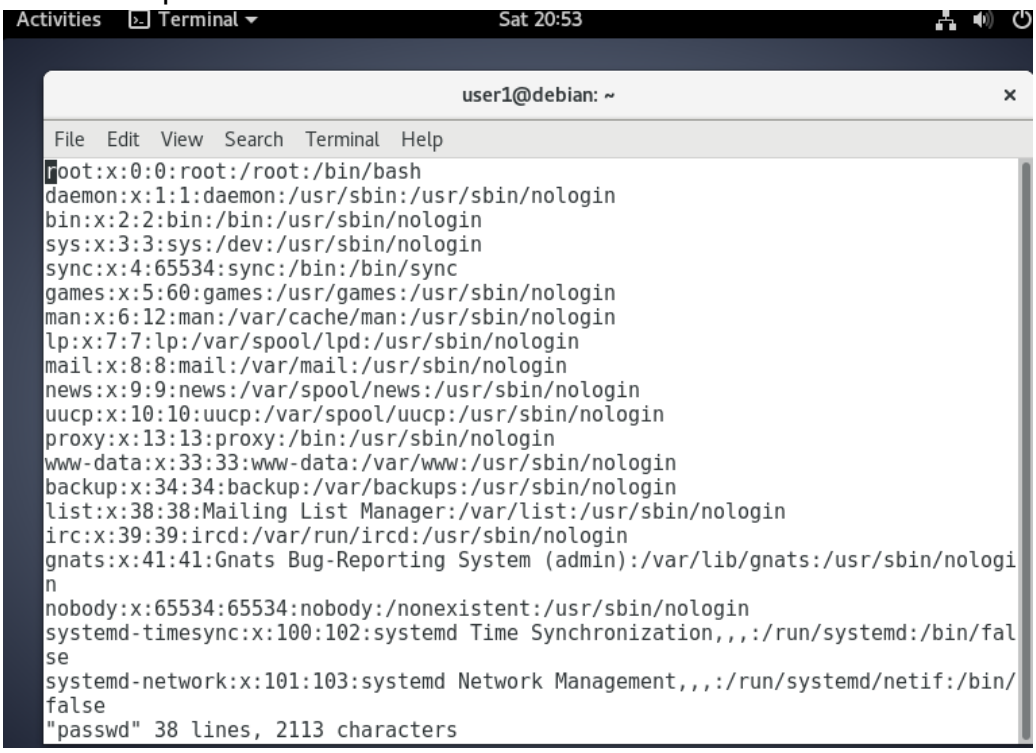
Editing

- vim - The default line editor in Linux. Every Linux OS has it.
- nano - An enhanced text editor.
- Emacs - Full blown text editor, still command line. Many programmers prefer it.
- gedit - Graphical based text editor.
- Libre Office Writer - Graphical based editor as a full replacement for MS Office Word.

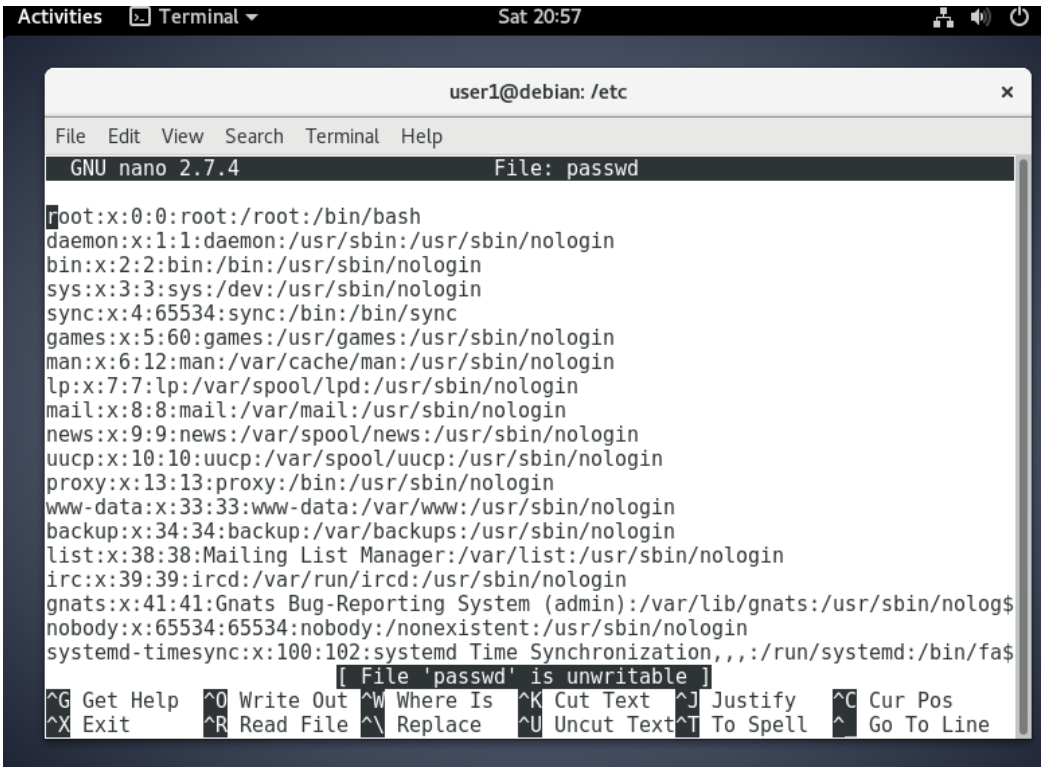
  If you are editing a plain text file or a configuration file, use only vim, nano, or

gedit. Emacs and Libre Office Writer embed formatting marks that can be misunderstood by the OS.

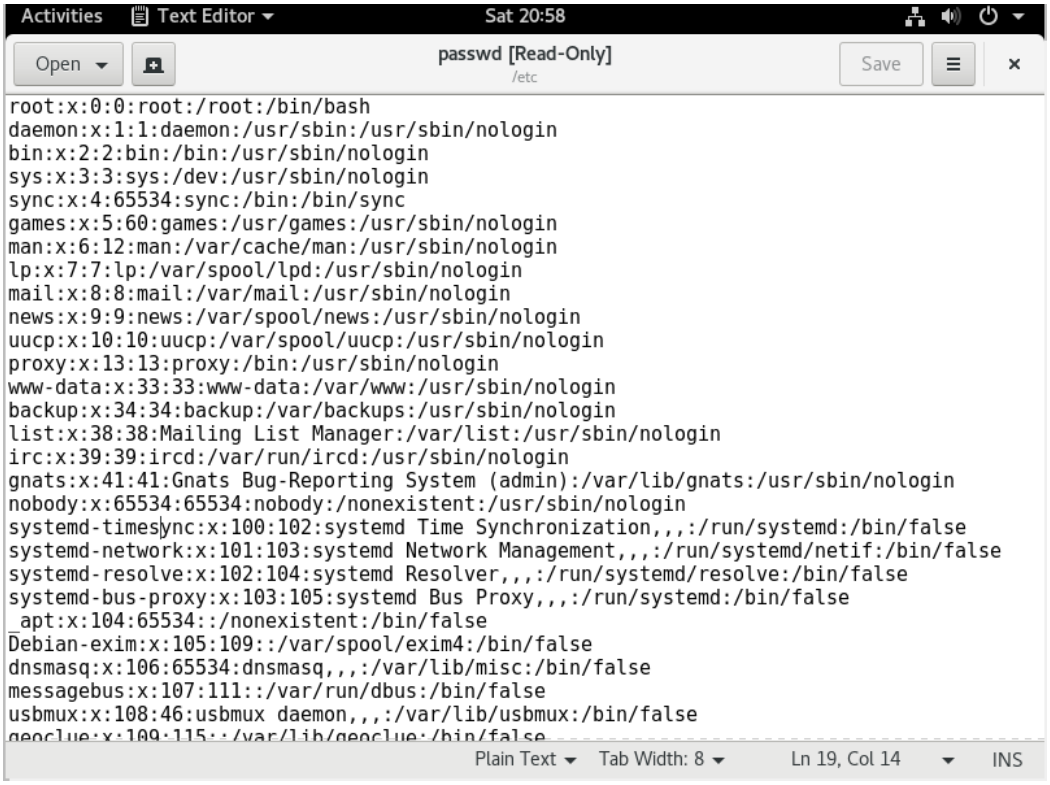
vim example:



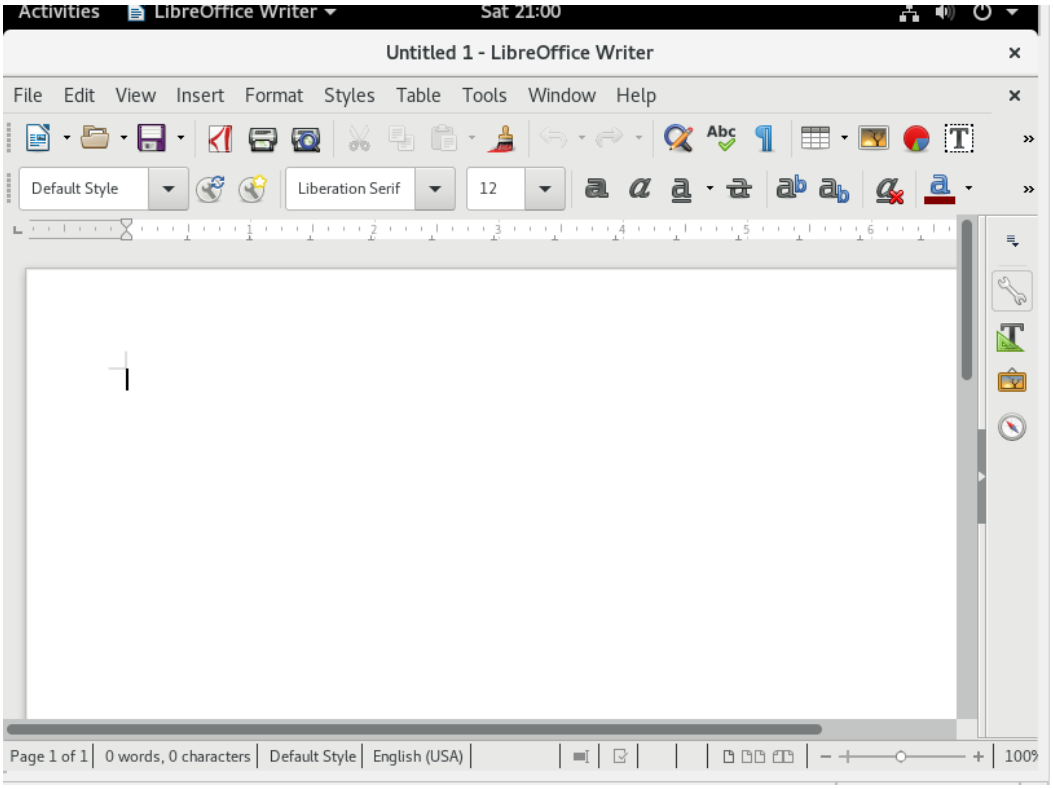
nano example:



gedit Example:



Libre Office Writer Example:



Objectives

In this project/lab the student will:

- Gain familiarity with Linux file tools.

Equipment/Supplies Needed

- As specified in Lab 0.0.1.

Procedure

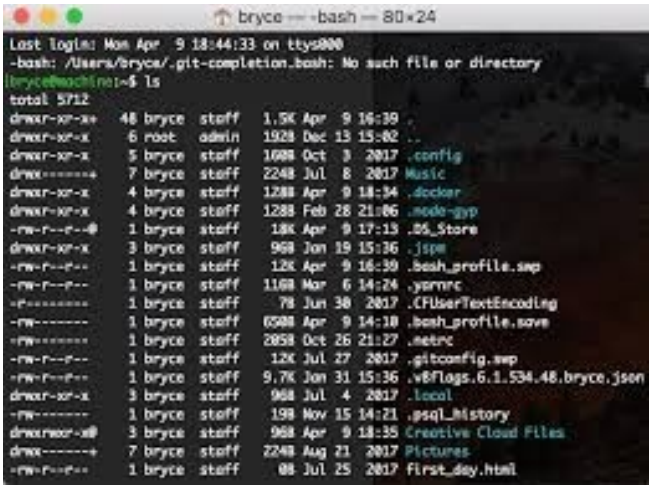
Perform the steps in this lab in the order they are presented to you. Answer all questions and record the requested information. Use the Linux Virtual Machine to perform lab activities as directed. Unless otherwise stated, all tasks done as a non-root user. If root access is needed use the sudo command.

Assignment

Launch Debian.

Open a Terminal. Enter:

1. `ls -la`



given : -rwxrw-r-- 1 root root 2048 Jan 13 07:11 afile.exe

where -

col 1 -file permissions,
col2 - number of links,
col3 - owner name,
col4 - owner group,
col5 - file size,
col 6 - time of last modification, and
col7 - file/directory name

File permissions is displayed as following:

first character is - or l or d, d indicates a directory, a line represents a file, l

is a symlink (or soft link) - special type of file

three sets of characters, three times, indicating permissions for owner, group and other: r = readable w = writable x = executable

In the provided afile.exe above, what are the group permissions for the file and the file size? Place the answers in a Word or Writer document.

Execute:

2. cd /etc/
3. more passwd

How many lines are displayed? Place the answers in a Word or Writer document.

Execute:

4. less passwd

Is the display similar? How is it different from more?

While we are in here try head and tail -

5. head passwd
6. tail passwd

Still in the Terminal, return to your home directory,

7. cd ~

Now create a directory -

8. cd Desktop
9. mkdir test
10. cd test

Make a file:

11. touch myfile.txt

What is the file size just created? Place that answer in the Word or Writer document.

copy the file,

12. cp myfile.txt ourfile.txt

Execute:

13. ls -la

Who is listed as the owner of the files? Are they similar?

Rename ourfile.txt:

14. mv ourfile.txt dump.txt

Now clean up:

15. rm myfile.txt dump.txt
16. cd ..
17. rmdir test
18. pwd

What directory are you currently in? Does the directory test exist? Can you determine what pwd means?

Open Terminal again. Execute:

19. nano .bashrc

Using the cursor keys arrow down till you see

HISTSIZE=1000
Change it to

20. HISTSIZE=500

Now save the changes by the following, Ctrl+x, then Y to save to disk. You should now be at the BASH prompt again. Take a screenshot, place that image in a Word or Writer document.

Repeat the above steps using gedit as the editor. Change the HISTSIZE to 400. Save the file by selecting File –> Save. Close by selecting File –> Quit.

Now that you can list permissions, you surely want to mess up with them .

Chown is the first command we're going through. It is used to change the owner (or user subject). Its syntax is:

chown OWNER FILE

Lets create a new file:

21. touch file1

Perform:

22. ls -la

Who is the owner of file1? Place that answer in the Word or Writer document.

Practice is better than words in this case; let's take a look at what happens when I change file1's owner from mark to root.

23. # chown root file1
24. $ ls -l

Who is now the owner of the file? Did the group ID change?

Note: I had to use a privileged user to use chown on someone else. For now you can access the privileged mode using sudo su or sudo COMMAND where COMMAND = the command you're issuing.

Chmod is probably the most difficult command among the three presented in this post. It is used to change the permissions of the three subjects and its syntax is:

chmod PERMISSION FILE

Now the problem is what goes into the permission field? In the precedent post I mentioned you can use two forms to represent permissions: r w x or 4 2 1, in this case we'll be using the numerical form. When you set the permission for one subject you will have to set them for the other two too! So be careful now: let's suppose we want to assign read+write+execute to owner, read to the group and none to others. It's time to calculate:

read	write	execute
4	2	1

The sum is 7. So for the owner of file1 the value is set to 7. For the group it is 4 for read only access. For everyone else, aka other, would be '-' or 0. Hence, the command would look like this:

$chmod 740 file1

The readout using ls -l would be similar to:

-rwxr----- 1 owner group 0 Jan 24 14:11 afile

Notice how it changed. Now suppose we want to give full permissions to everyone! Can you guess which number I will use? Try:

25. $ chmod 777 file1
26. $ ls -l

What must you do to have the above command work correctly? Why might you not want to do that?

Lab Submissions Proof: Provide screenshots as indicated in the lab; upload your proof to Canvas for grading.

Rubric

Checklist/Single Point Mastery

Concerns Working Towards Proficiency	Criteria Standards for This Competency	Accomplished Evidence of Mastering Competency
	Criteria #1: In the provided afile.exe above, what are the group permissions for the file and the file size? (10 points)	
	Criteria #2: How many lines are displayed?(10 points)	
	Criteria #3: Is the display similar? How is it different from more? (10 points)	
	Criteria #4: What is the file size just created? (10 points)	
	Criteria #5: Who is listed as the owner of the files? Are they similar?	

	(10 points)	
	Criteria #6: What directory are you currently in? Does the directory test exist? Can you determine what pwd means? (10 points)	
	Criteria #7: Who is the owner of file1? (10 points)	
	Criteria #8: Who is now the owner of the file? Did the group ID change? (10 points)	
	Criteria #9: What changed? (10 points)	
	Criteria #10: What must you do to have the above command work correctly? Why might you not want to do that?(10 points)	