GREP

Introduction and/or Background
Most files in Linux are text based. Text editors are used for editing text files, writing code, updating instruction files, etc., as part of administering the system. They are one of the most frequently used applications on a linux-based system. Nano is a common text editor that mostly comes pre-installed on several distributions. Some other popular editors used in Linux are the vi, vim, pico, gedit.

The need to view, manipulate and change text data without "editing" files requires some additional tools to help filter/view their contents. The grep utility is widely used to help filter through text files based on patterns.

Objectives
In this project/lab the student will:

- Use a text editor and text filtering tool.

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

nano Text Editor

Most system configurations are done by editing text files in Linux, mainly using command-line text editors. One of the most popular and versatile of these is nano. Some Linux distros may already come with the nano text editor pre-installed. Some other popular editors used in Linux are the vi, vim, pico, gedit

Check if nano is installed, use the *package_name* –-version command.

1. `nano --version`

If not installed, execute the apt-get install command, as root.

Enter the root password when prompted.

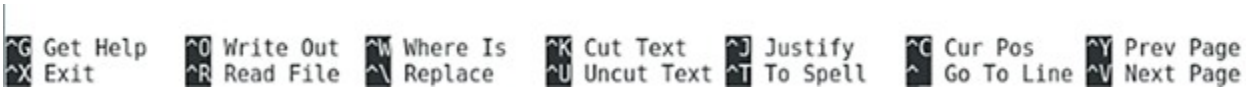
2. sudo apt-get install nano

Confirm the install with nano –version command again.

To open/edit an existing file in nano, use the **nano** command [**nano *filename***, or **nano** *path filename* (where *path* is the file location, if file is not in the present directory).

To create a new file use the **nano *filename*** command. If no name is provided the file will be created and nano will prompt for a name when saving/exiting the editor. The nano window editor should appear with this command, ready for editing.

A shortcut 'menu' will be located at the bottom of the editor window. These are the most commonly used shortcuts in the editor.



Note: "^" (caret) = **CTRL** key on keyboard

Within the editor:
Use the arrow keys to move the cursor within the nano editor
Use CTRL + O to save changes/Continue editing
Use CTRL + X to exit editor [prompt for saving Y or N if changes made; otherwise editor closes]
Use CTRL + Y (page down), and CTRL + V (page up) to move up and down the screen.

To select text, go to the beginning of the desired text and press ALT + A. This will set a mark for selecting. Then move over the text with arrow keys. Press ALT + 6 to copy the selected text to the clipboard. To cut the highlighted text, [[place on 'clipboard'] press CTRL + K. Use CTRL + U to paste the text into a selected location. *For a more detailed explanation of the nano editor and features:* *https://www.howtoforge.com/linux-nano-command/*

From the home directory, create a file, *seuss.txt*

3. nano *seuss.txt*

Enter the following into the editor:

I know some good games we could play, said the cat.

I know some new tricks, said the cat in the hat.

a lot of good tricks.

I will show them to you.

Your mother will not mind at all if i do.

Use ***^X*** to exit the editor and save the file.

Create a second file, *seuss2.txt*. Enter the following into the editor:

should we tell her the things that went on there that day?

should we tell her about it?

now, what SHOULD we do?

well... what would YOU do

if your mother asked YOU?

Use ***^X*** to exit the editor and save the file.

Check to make sure the files were created/saved using the ***ls*** command. Record the output.

Global regular expression print (grep)

grep searches files for lines containing a pattern (word or string of characters). It is case sensitive.

Syntax: grep *options* '*pattern*' *filename(s)*

Common Basic Options for grep

-w	Searches for the entire pattern or word
-i	ignore case (search for upper and lower case matches)
-n	displays the line number in the file of the given search string
–c	Count Number of Matches
-r	search for the string/word in the current directory along with all of the subdirectories
--color	forces grep to display output in colors
–A	displays the matched line and number of lines either that come after the search string
-B	displays the matched line and number of lines either that come before the search string

View the file ***Seuss.txt*** using the cat command:

4. *cat Seuss.txt*

   Record a screenshot of the output.

   View the file ***Seuss2.txt*** using the cat command:

5. *cat Seuss2.txt*

   Record a screenshot of the output.

   Use grep to search any instance of the word/pattern **at** in the ***Seuss.txt*** file.

6. grep '*at*' *Seuss.txt*

Record a screenshot of the output.

Use grep to search just the word/pattern **at.**

7. grep –w '*at*' Seuss*.txt*

Record a screenshot of the output.

Use grep to search for the pattern/word ***you*** in the two files created (***Seuss.txt, Seuss2.txt***).

8. grep *–w 'you' Seuss.txt Seuss2.txt*

Record a screenshot of the output.

Use the –i option to check for the word ***YOU*** (all capital letters) in the Seuss2.txt file.

9. grep *–i 'YOU' Seuss2.txt*

Record a screenshot of the output.

Count the number of times the word ***I*** appears in the Seuss.txt file.

10. grep *–c 'I' Seuss2.txt*

Record a screenshot of the output.

Show the line numbers for instances where the word ***what*** appears in the ***Seuss2.txt*** file.

11. grep *–n 'what' Seuss2.txt*

Record a screenshot of the output.

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: Record output showing seuss1.txt and seuss2.txt using the ls command (10 points)	
	Criteria #2: Record the output of cat Seuss.txt (10 points)	

	Criteria #3: Record the output of cat Seuss2.txt (10 points)	
	Criteria #4: Record the output of grep ‘at’ Seuss.txt (10 points)	
	Criteria #5: Record the output of grep -w ‘at’ Seuss.txt (12 points)	
	Criteria #6: Record the output of grep -w ‘you’ Seuss.txt Seuss2.txt (12 points)	
	Criteria #7: Record the output of grep –i ‘YOU’ Seuss2.txt (12 points)	
	Criteria #8: Record the output of grep –c ‘I’ Seuss2.txt (12 points)	
	Criteria #9: Record the output of grep –n ‘what’ Seuss2.txt (12 points)	