Name	Laura Adam Feb 22,2019

CTS-120-841-Lab Module 4

- In this lab you will enter different Linux commands and answer questions about the results.
- Include a screen print of **just the area of the screen with the desired result** (not the whole screen) in the table cell below the question, unless otherwise instructed.
    - *Reminder: Use the **Shift-Ctrl-Prtscr shortcut** & select just the area that you want.*
- The lab is worth a total of 10 points – some questions have multiple sections

Module 4 – Section 1

Open a terminal window type ***gedit foo.txt*** – then enter the following data:

Save the file and close it.

Use this foo.txt file to complete the questions below. - use cat, less, head, tail, sort and/or uniq

	1. Use the command to output the first 10 lines of 'foo.txt'	1 Pt
Command:	$ HEAD + 10 FOO.TXT |CAT -N	
Screenprint:	1 ==> foo.txt <== 2 foo 3 foo 4 woo 5 noo 6 toocat foo.txt \ca 7 foo 8 coo 9 boo 10 coo 11 woo	

	2. Use cat and sort to output 'foo.txt' in sorted order.	1 Pt
Command:	CAT FOO.TXT | SORT	

Screenprint:	student@localhost ~]$ cat foo.txt | sort 1 2 3 4 5 5 8 8 9 9 9 boo coo coo foo foo foo moo noo soo soo too too too woo woo woo zoo

sort

	3. Use sort (only) to output 'foo.txt' in sorted order.	1 Pt
Command:	SORT FOO.TXT	
Screenprint:	[student@localhost ~]$ sort foo.txt 1 2 3	

	4 5 5 8 8 9 9 9 boo coo coo foo foo foo moo noo soo soo too too too woo woo woo zoo

4. Use cat and pipe to wc –l to get a count of the number of entries in 'foo.txt'	1 Pt
Command:	$ CAT FOO.TXT | WC
Screenprint:	[student@localhost ~]$ cat foo.txt | wc 29 28 91

5. Use sort and pipe to uniq with the -c switch to output a list of duplicate lines preceded by the number of times the line occurs. Your output should look something like this: . . . 2 coo 3 foo 1 moo 1 noo . . . But show me the whole list.		1 Pt
Command:	CAT FOO.TXT | SORT | UNIQ -C	
Screenprint:	[student@localhost ~]$ cat foo.txt | sort | uniq -c 1 1 1 1 2 1 3 1 4 2 5 2 8 3 9 1 boo 2 coo 3 foo 1 moo 1 noo 2 soo 3 too 3 woo 1 zoo	

Module 4 – Section 2

1. Using gedit**,** create a file called **softkitty** on your system.
2. Enter the following text in the softkitty file:

   Soft kitty,
   Warm kitty,
   Little ball of fur.
   Happy kitty,
   Sleepy kitty,
   Purr Purr Purr.
3. Save the file and close it.
4. Use cat to view the softkitty file on standard output. Familiarize yourself with the text of the file. It is a 6-line song. Make sure you have not made any spelling errors. If you did, edit your file

```
[student@localhost ~]$ cat softkitty
Soft kitty,
Warm kitty,
Little ball of fur.
Happy kitty,
Sleepy kitty,
Purr Purr Purr.
[student@localhost ~]$
```

6. Just read the file to see how many times do the following words appear? Do not use any technology.	4 Pts
Soft	1
kitty	4
of	1
Purr	3

It was easy to count the instances of the words since our file is so small. Unfortunately in the "real-world" you files will be much larger and you will have to be able to use tools to help you analyze them efficiently. Let's use some of those tools on this small file to familiarize ourselves with their capabilities.

	7. Use the command **grep kitty softkitty** • How many times did the word kitty appear? • What made it easier to count?	2 Pts
Answer:	**[student@localhost ~]$ grep kitty softkitty** **Soft kitty,** **Warm kitty,** **Happy kitty,** **Sleepy kitty,** **[student@localhost ~]$** **Kitty appeared four times – highlighted in red on my terminal**	

	8. Use the command **grep –c soft softkitty** • How many times does the word "soft" appear? • Why?	2 Pts
Answer:	**First grep found zero because grep is case sensitive (Looking for an exact pattern match - When I changed the case – grep found one instance of "Soft"** **[student@localhost ~]$ grep -c soft softkitty** **0** **[student@localhost ~]$ grep -c Soft softkitty** **1** **[student@localhost ~]$**	

	9. Use the command **grep –ic soft softkitty** • Now, how many times does the word "soft" appear? • What is the –i switch used for?	2 Pts
Answer:	**The switch -i makes the pattern independent of case – Now it found one instance of soft although the exact pattern in the text is "Soft"** **student@localhost ~]$ grep -ic soft softkitty** **1**	

	10. Use the command **grep –c of softkitty** • How many times does grep indicate "of" appears? • Why do you think grep counted the number of times "of" appears differently than what you counted?	2 Pts
Answer:	**Grep found two instances of the pattern "of". Once inside of Soft and once as the word "of"** **[student@localhost ~]$ grep -c of softkitty** **2**	

	11. Remove the –c option in the last command to see the difference • What was happening to alter the count of "of"?	1 Pts
Answer:	**It shows you the two instances of the pattern OF highlighted in red** **[student@localhost ~]$ grep of softkitty** **Soft kitty,** **Little ball of fur.** **[student@localhost ~]$**	

	12. Use the command **grep –w of softkitty** What is the –w switch used for?	1 Pts
Answer:	**It finds the pattern of "OF" being used as a word and not just a pattern. That is to say it has a space in front and back of "OF"meaking it a word** **The word is highlighted in red on the terminal output**	
Screenprint:	**[student@localhost ~]$ grep -w of softkitty** **Little ball of fur.**	

	13. Use the command **grep –wc of softkitty** • How many times does grep indicate the **word** "of" appear? • Is the count correct now?	2 Pts
Answer:	**It finds the pattern of "OF" being used as a word and not just a pattern and counts the number of times it occurs and returns the number of counts** [student@localhost ~]$ grep -wc of softkitty 1	

	14. Use the command **grep -n ball softkitty** • On what line number does the word "ball" appear? • What is –n switch used for?	2 Pts
Answer:	**It finds where the pattern occurs and returns the line number where that pattern is found. In this case the pattern Ball is found on line 3** **[student@localhost ~]$ grep -n ball softkitty**	

	3:Little ball of fur.

	1. Use the command **grep -v kitty softkitty** & Analyze the results. • What is –v switch used for?	2 Pts
Answer:	**Grep -v finds the lines with the desired patter nad excludes them from the standard output. There are two lines without the word kitty and these are the two lines that are returned.** **[student@localhost ~]$ grep -v kitty softkitty** **Little ball of fur.** **Purr Purr Purr.**	