Lab 5

More Bash plus Regular Expressions

CSCI 344 Shell Programming

Due Friday February 24, 11:59pm on Canvas
30 points

For each of the following problems, where applicable, show a screen shot of your code and output.

Before beginning this lab, create a file named *file1* with variety of data such as colors, animals, and numbers. Capitalize some of the letters. For example:

```
Dog
6 blue
Cat donkey
cat
pIG CAT yellow
23
145 pink
9 horse
65
horse
White
Black BLUE horse
grEEn
purple
cow
5 dog
12
…
```

1. **What does the following command do?**

history | grep cd | wc

2. **Is there a difference between the following 2 commands? If yes, what?**

cat myfile | grep horse
grep horse myfile

3. **Apply the grep command to *file1* by using the options -i, -c, and -v. Explain what each option does.**

4. **Write a script to calculate the average number of characters per line in *file1* using the *wc* command.**

5. **Create another file named *file2* "very similar" to *file1*. What does the following command do? Be specific to what the < character represents.**

diff file1 file2 | grep '<'

6. **What does the following command do?**

grep horse file1 >> file3

7. **Given file named *file1*, what *chmod* command provides the file owner full access (read, write, execute), the group users only read and execute access, and everyone else execute access only?**

8. **Recall Lab 2 how we applied the exit status $? variable. Provide an example that sets this exit status variable and then write an *if-statement* that checks it if the previous command was successful.**

9. **Write the code that parses out “only” the $HOME directory of username equal to “jim” (hint: refer to the /etc/passwd file)?**

10. **Write a script that sorts *file1* alphabetically and saves the results to *file2*. Then, sort the same *file1* file numerically and save it to *file3*. Ensure you can explain the results.**

11. **Provide an example of creating a tar ball file and then show how it is extracted.**

12. **Copy *file1* to the /tmp directory and have a colleague retrieve it. Make sure to re-name it because not all students can have the same filename of *file1* (i.e. file1_Horacio).**

13. **Use the UNIX *find* command to scan our AWS server and *count* how many files exist that are greater than 10MB. The correct answer is ~111 files (or a bit higher) on our AWS server have a size greater than 10MB. To execute, my suggestion is to send standard error to /dev/null and then pipe the output to the *wc* command. You may have to leave the command running for ~5 minutes to scan the entire server.**

14. **Given the RegExp \D\d+ and input text 7987dasf9d83rwd**
    a) **Name all the matching patterns**
    b) **Suppose we replace the plus + metacharacter with the asterisk * metcharacter. Are there any changes with the matching patterns? If yes, what are they?**

15. **What RegExp can you match complete numbers containing the digit 5. The values to compare with are provided as a list in a file. The matching number should encompass the entire line.**

16. **Provide the RegExp to validate a time value with a format of 2-digits, a colon, 2 digits and either am/AM/pm/PM. Ex. 10:35PM.**

17. **Describe the following regular expression and provide 3 matching patterns. (\d)[3](\d{2})(\d)\3?$**

18. **Fully explain each expression/metacharacter(s) and provide 3 distinct matching patterns given the following regular expression: (\D)[a-z]{2}(\d)\1[4]**

19. **Circle any match(es) found by using the following regular expression and text. Ignore the embedded spaces within the text since they are only included to make it easier to circle any match(es) found: regular expression = ([4])(\d){2}\1\1? and text = 4 1 5 3 4 1 2 4 4 4 a 4 4 6 4 A 4**

20. **Write a regular expression that matches all numbers between 5-32, (including 5 and 32).**

SUBMITTING WORK

If you are only able to complete the lab partially, clearly state it. Explain where you time was spent and any hurdles that were or were not met. Given the steps mentioned above, your deliverables are the following:

1. Start by pasting as many screenshots that describe your work for each step into one continuous document.

2. Spend much time on adding your personal comments within your lab submission. Refer to syllabus and example as shown in class. Anyone should be able to read your lab write-up and understand what it is trying to present.
3. Turn in a copy of all your RegExp code, and include any screen shots of your output, all within your **one** PDF/Word formatted document. DO NOT SIMPLY TAKE SCREEN SHOTS. Include highlights and arrows to describe specific points.
4. Add a conclusion section to your write-up. Elaborate on at any aspect of your lab. Some examples are: what was challenging, how were you able to overcome any hurdle(s), and/or what did you learn.
5. *** Please refer to the required format specified in the syllabus (i.e. turn in one PDF/Word/OpenOffice document). No individual image files. And no zipped, no compressed and no rtf-formatted files.

GRADING

To achieve a maximum score, students will need to clearly prove that they completed the goal. A clear description of the steps taken, and screen shots are essential. Partial credit is given if students can clearly state what was done and what is not working. If the instructor is required to decipher incompleteness, much less partial credit will be given.

Points lost for incompleteness, sloppiness, lateness, or failure to follow instructions.

Late policy: refer to syllabus