Assignment 6

sed command

Name:

Download this document, complete the assignment and save as a .pdf. After you complete this assignment, transfer the file to the Linux web server, make it visible and send the URL to your instructor. Email your instructor if you have any questions judy.pino@sfcc.edu.

Student Learning Objectives:

After successful completion of this exercise a student should be able to:

- Log into a UNIX/Linux based server
- Perform and explain sed command
- Navigate a Linux file system
- Create a text file and perform operations on the file

Read/resources:

sed tutorial: https://linuxconfig.org/learning-linux-commands-sed
stdin and stdout http://www.learnlinux.org.za/courses/build/shell-scripting/ch01s04.html

- Locate a nursery rhyme on the WWW. Copy the nursery rhyme to a text editor (Notepad). Save the file on your PC in a folder called "filestotransfer". Remember when naming files, NO spaces or special characters in the file name.

Done

- Using Window command line or SSH on a MAC or Linux system transfer the file to the tmp folder on SFCC CS web server. Send me and image of proof showing the file was received and transferred 100% to the CS web server. You send an image embedded in a .pdf.

Sent as an email.

- Next, log into the CS web server. Create a folder in your public_html directory called nurseryrhyme. Next, navigate to the tmp folder. Locate another nursery rhyme document in the tmp folder. Who does that file belong to? Move that file to your nurseryrhyme directory. Could you move it?

The file 'itsybitsyspider.txt' belongs to 'twise'
I could not move the file. It said "Operation not permitted"

- **Copy** the file to your nurseryrhyme directory. Give the command you used to move the file:

```
cp itsyBitsySpider.txt ~/public_html/nurseryrhyme
```

- Once you've copied the file to your directory use the command ls – l to view a long listing of the file. What information are you given about the file? Who does it belong to? Why did the file permissions change? Explain.

"ls -l" gives whether it is a directory and permissions, hard link count, timestamp and the file name.

The copy belongs to me.

Copying involves writing and I'm the one doing the writing, so it has my permissions independent of where the bits in the file came from.

- Once you have moved the file use the ***cat*** command to view it and answer the following questions:
- What does line 5 say? What command can you use to add line numbers?

"and dried up all the rain"
I used "cat -n itsyBitsySpider.txt"

- Use the wc command, what information are you given? Does a **character** include a space? How can you replace a string that has both upper and lower case letters?

"wc" gives line count, word count and character count.
The character count includes spaces.

You can match in a sed substitute command using the "I" modifier

```
sed 's/the/_the_/gI' itsyBitsySpider.txt
```

Research the command sed. Preliminary:

1. Explain the utility sed.

"sed" stands for "Stream EDitor"
It lets you search for and edit patterns in files.
You can:
replace one word with another word
print out a range of lines in a file
print out all but a given range of lines
print out lines that match a pattern
You can write longer scripts also that combine these operations for more complex tasks.

2. Run command: sed –version. What version are is the system running?

On my mac the version is so old it doesn't support -version
On the server it is version 4.2.2

3. Explain the standard input and output in Linux (stdin, stdout).

Most commands I’ve worked with thus far read input from standard input and print their output to standard output.
By default standard input comes from the keyboard but it can also be gotten from other sources such as files or the outputs of other commands. To redirect input to come from a file you use “<“. To redirect input to come from the output of another command you use “|”.
By default standard output prints to the terminal but it can be redirected to files or the inputs of other commands. To redirect to a file you use “>”. To redirect to another command you use “|”.

4. What are redirection operators? Give 2 examples.

The operators are > and < and | and >>

The command

```
   ls -lt | head
```

would print the first ten lines of the “ls -lt” command

The command

```
   sed 10,20 myfile.txt >> lines.txt
```

appends lines 10 through 20 of myfile.txt to the end of lines.txt

5. What is “noclobber”?

When you redirect output into a file it clobbers the old data there.
If you set the shell variable “noclobber” redirection won’t clobber existing files.

The sed (stream editor) utility is a batch (non-interactive) editor. It transforms an input stream that can come from a file or standard input. It is frequently used as filter in a pipe. Unless you instruct it not to, command "sed" sends all lines selected or not to standard output. When you use the -n option on the command line, “sed” sends only certain lines to standard output. You’ll learn sed and stdin and stdout.

Lab:

1.
Locate the nursery rhyme, “Sing a Song of Sixpence” and save it in your nurseryrhyme directory. Name the file sixpence with no file extension. Exclude the title of the rhyme in the document. The document should begin with “sing a song” and end with “nose”.

The sed command is used to replace text in a file. Look at and run the command:

```
sed s/pie/cake/ <sixpence> newrhyme
```

Here the s uses the substitution operation, and the / are delimiters. The “pie” is the search pattern and the “cake” is the replacement string. The < is standard input from the file “sixpence” and > is standard output to the newly created file “newrhyme”.

2.
Run the command:

```
sed –n /bird/p < sixpence > nobirds
```

Here the –n suppresses lines that do not contain the word “bird” and prints all lines with the pattern “bird” to a newly created file called “nobirds”. The < is standard input from the file “sixpence” and > is standard output to the newly created file “nobirds”.

3.
Run the command: sed –e 's/king/queen/g;s/his/her/g' <sixpence> nowaqueen

Here sed provides -e option to run multiple sed commands in a single sed command. You have the output of one sed command as input to another sed command. The < is standard input from the file "sixpence" and > is standard output to the newly created file "nowaqueen". The s is the substitution and g is global. Cat the "nowaqueen" and check the replacements.

4.
1st Run the command: cat –n sixpence
2nd Run the command: sed –n 4p sixpence
3rd Run the command: sed –n 4p sixpence <sixpence> onlyline4
4th Run the command: cat onlyline4
Explain what is happening.

1.Presents sixpence with numbered lines.
2.Outputs "Baked in a pie.", the 4th line of the poem.
3.Puts the 4th line of sixpence in a new file called "onlyline4". The second "sixpence" is ignored.
4.Outputs the content of onlyline4, which is "Baked in a pie.".

5.
Run the command: sed –n '4,8p' sixpence
This command returns lines 4 through 8 inclusive.

Using the nursery rhyme you transferred from the tmp folder. Create and run at least 3 sed commands. Provide and explain the commands along with their output.

Replace spider with elephant.

```
$ sed s/spider/elephant/g itsyBitsySpider
The itsy bitsy elephant climbed up the waterspout.
Down came the rain
and washed the elephant out.
Out came the sun
and dried up all the rain
and the itsy bitsy elephant climbed up the spout again.
```

Show only the lines that contain the word and.

```
$ sed -n /and/Ip
and washed the spider out.
and dried up all the rain
and the itsy bitsy spider climbed up the spout again.
```

Replace all periods with exclamation points.

```
$ sed s/\.$/\!/g itsyBitsySpider
The itsy bitsy spider climbed up the waterspout!
Down came the rain
and washed the spider out!
Out came the sun
and dried up all the rain
and the itsy bitsy spider climbed up the spout again!
```

Finally, research the web on “Hacking with Linux sed”. What did you find?

I searched for “Hacking with Linux sed” and I found many pages of tutorials on “sed”. On StackExtchange I did find someone who thought he had been hacked, and had found the following command online and wanted to know if it was safe.

```
$ find . -type f -name "*.php" -exec sed -i '/base64_decode/d' {} \;
```

He later posted an update upset that the command had deleted the entirety of all lines that contained “base64_decode”.

Can this command be used on the system password file?

Yes, the password file is readable by all, but only writeable by root.

The password file is located inside the /etc directory. From your current location, run the command: cat /etc/passwd. With this command you’re viewing the password file on the system.

Continue to learn!
Have fun!