# The Sed Family

Name: Robert Acosta

Instructor: Adrianna Holden-Gouveia

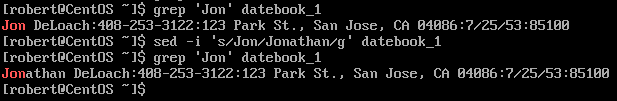
Subject: Linux Administration

Date: 9 September 2020

Comments: In Sed is important to enclose the character with forward slashes for searches and executions.

**Back Slash \** : Special sequences of backslash and one letter or symbol means special instruction registered in the GNU.

1. Change the name Jon to Jonathan .

**sed -i ‘s/old-word/new-word/g’ filename**

**sed -i** = Edit In place. This command allows us to edit files. It overwrites the content of a file without creating a new one.

**s** = substitute

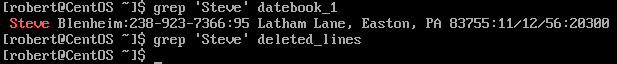
**g** = global

1. Delete the first three lines.

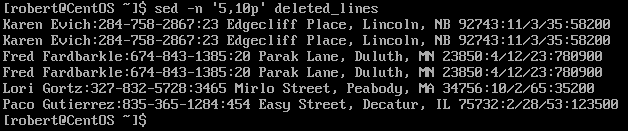
 **sed ‘1,3d’ filename > new\_filename**

**d** = delete

Explanation: ‘1,3d’ lines 1 through 3 will be deleted from datebook\_1 and the result will be in a new file called > deleted\_lines.



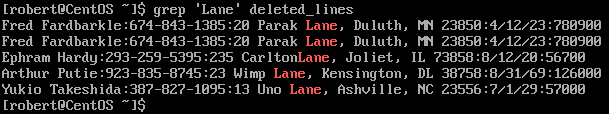
1. Print lines 5 through 10

 **sed -n ‘line#,line#p’ filename**

**-n** = Disables the automatic printing and only prints the lines you do tell to print it.

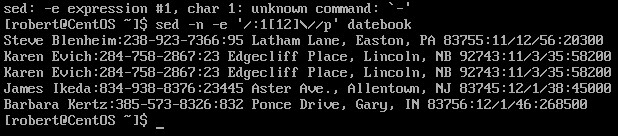
**p** = print

1. Delete lines containing Lane.



 **sed ‘/Lane/d’ deleted\_lines**

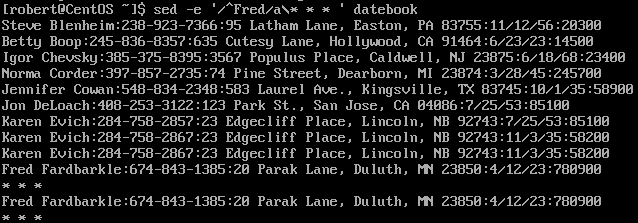
1. Print all lines where the birthdays are in November or December

**sed -n -e ‘/:1[12]\//p’ datebook**

Comments: First thing I got help from a tutor to do question 5. We use **-n** to tell sed to print only lines with a specific pattern. Then we write **-e** to use expressions. To write our expression with the start with forward slash **/**. Then we write **:1[12]** which means 11 or 12 followed by a back as well as a forward slash \/. This is the alternative form to write a forward slash. Finally, we close our expression with a backslash and a “**p**” which mean to print the specific content.

**:1[12] :** The : punctuation means that the first digit will stay the same. The second digit is the only one that would change.

**\/ :** The alternative delimiter of a forward slash.

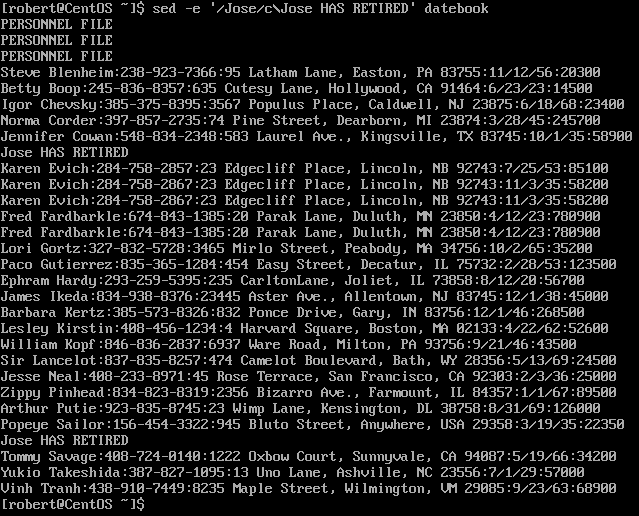
1. Append three asterisks to the end of lines starting with Fred **sed -e ‘/^Fred/a\\* \* \* ‘**

**sed -e :** Means expression**.** It tells sed to execute the line argument or expression as sed program.

**^** : Beginning of the word. If we write ^Fred, any line that contains Fred at the beginning will be search.

**a** : a\ Appending text after a line.

1. Replace the line containing Jose with JOSE HAS RETIRED.

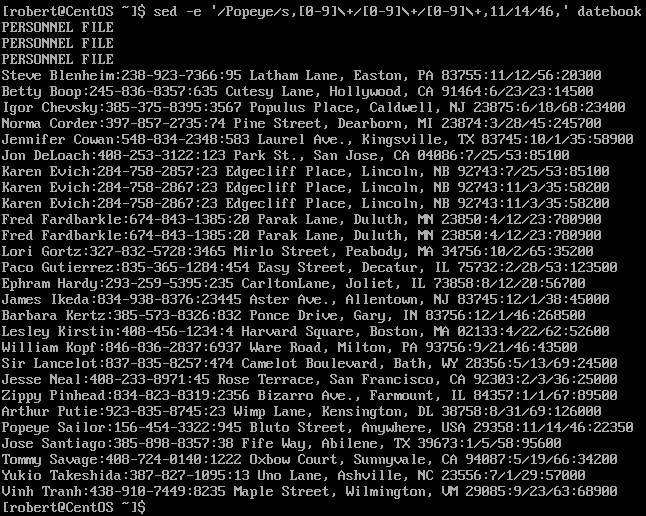


**sed -e ‘/Jose/c\Jose HAS RETIRED’ datebook**

**c** : Replace the line with text

Explanation: We start writing **sed -e**, to tell sed that we are going to use expression. Then we enclosed the word /Jose/ with forward slashes. Because that is the word that we want to replace. **c\** allow to replace the line that contains Jose with the following text after the \ back slash.

1. Change Popeye 's birthday to 11/14/46 . Assume you do not know Popeye's original birthday. Use a regular expression to search for it.



**sed -e ‘/Popeye/s, [0-9]\+/[0-9]\+/[0-9]\+,11/14/46, ‘ datebook**

Explanation: We start with sed -e to use special expression. Then we start with the pattern /Popeye because that is the entry we want to search for, then another forward slash and add an ‘**s’**.

As I previously mentioned, ‘s’ stands for substitute. After the s we add a **comma** to write the old data. Note: We can substitute the **comma** for a **#** or a **/**, it is up to the user which one he or she prefers. I decided to use the comma on this case.

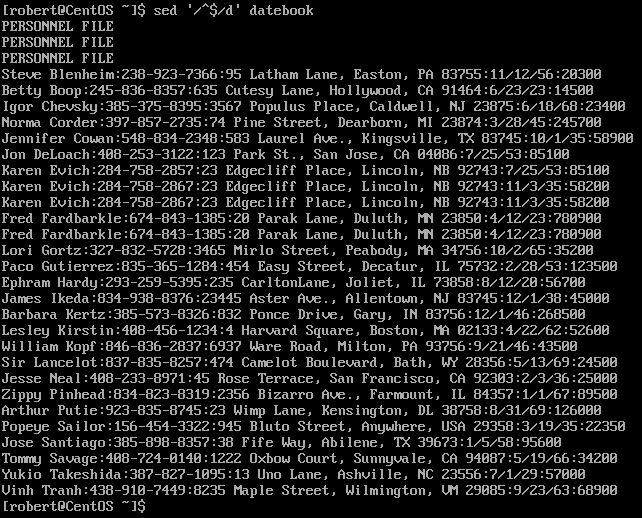
After the  **s,** we type the old date, but we do not know the Popeye’s date. For this reason, we use the command [0-9], to tell sed to search for numerical character from a range of 0-9. A date format is written as 11/12/80. We type [0-9] +\. **+\** Means followed by. Therefore, we end up typing [0-9]\+/[0-9]. This translate to 0-9/0-9/0-9.

Then we enclosed the new data with commas. Ex: , 11/14/46, . Remember that we start the substitute command as “ **s,** “. If we would use the **s#** symbol, we would enclose the new date as #11/14/46#. The results would be the same. You can use different symbols for the substitute command as I mentioned.

**[0-9]** : We are telling sed to search for numerical character from a rage of 0-9

**/+** : Followed by a …

1. Delete all blank lines.

 **sed ‘/^$/d’ datebook**

Comments: If we combine the commands, “**^** beginning go the line” without any text, plus the command “**$** at the end of the line” followed by the “command **d**”, which is the delete command, all the blank lines will be deleted.

^ : Beginning of the line anchor

$ : End of the Line anchor

Write a sed script that will

a. Insert above the first line the title  PERSONNEL FILE .

b. Remove the salaries ending in  500 .

c. Print the contents of the file with the last names and first names reversed .

d. Append at the end of the file  THE END .

On this section I had help from tutors. However, I will explain what each command does.

A shell scrip is generally end in “**.sh**”. However, for sed is different.

To create a sed scrip we type the following: **vi “name of the file”.sed**

To make the file executable: **chmod +x filename.sed**

To run the script: **sed -f script.sed filename**

**1i\”Insert Text”**: 1 means first line and “i” insert. If we combine both together ‘**1i\**’, sed will insert the writing text in the first line.

**/500$/d** : Similar to the grep assignment question #3. **$** means at the end of the line. If we type **500$** sed will search for any text that ends with 500. If we add **/d,** which means the delete command. Sed will delete any line ending with 500.

**s****/\([a-zA-Z]\*\) \([a-zA-Z]\*\)/\2 \1/** :This one was the hardest one to do. I had to look on the internet to see an example. What I understand is that “**s**” is for substitute. **([a-zA-Z])** This is telling sed that we are about to replace a word that has alphabetical characters from a range of a-z lower case or A-Z upper case. **\*\** this mean anything that match the previous criteria. **\2 \1/** This means to replace the column 2 with 1 and vice versa.

**$a\”Insert Text”**. Question 6 of the assignment. We use the “**a\**” command to append. We already know that **$** means “at the end of the line anchor”. If we combined both “**$a\**” sed will append the follow up text at the end of the file.

# References

*Sed, a Stream Editor*, www.gnu.org/software/sed/manual/sed.html.