Jack Berberian

CIS 245

1/27/2022

How to Use Sed on Ubuntu (Linux)

What is Sed?

Sed is a non-destructive text editor which means that you can use it with different commands on your files and it won’t modify them unless you tell them to explicitly.

Some Things to Know Beforehand

In order to use a Sed command, you need three things, first type Sed to signify to the computer that you want to use sed, then the command, something like s/Meg/Megan and then the name of the file SedLab to signify to Sed, where you want Sed to run the command.

So, an example of a sed command could be:

Sed s/Meg/Megan SedLab which would tell sed to substitute Meg for Megan in the file SedLab.

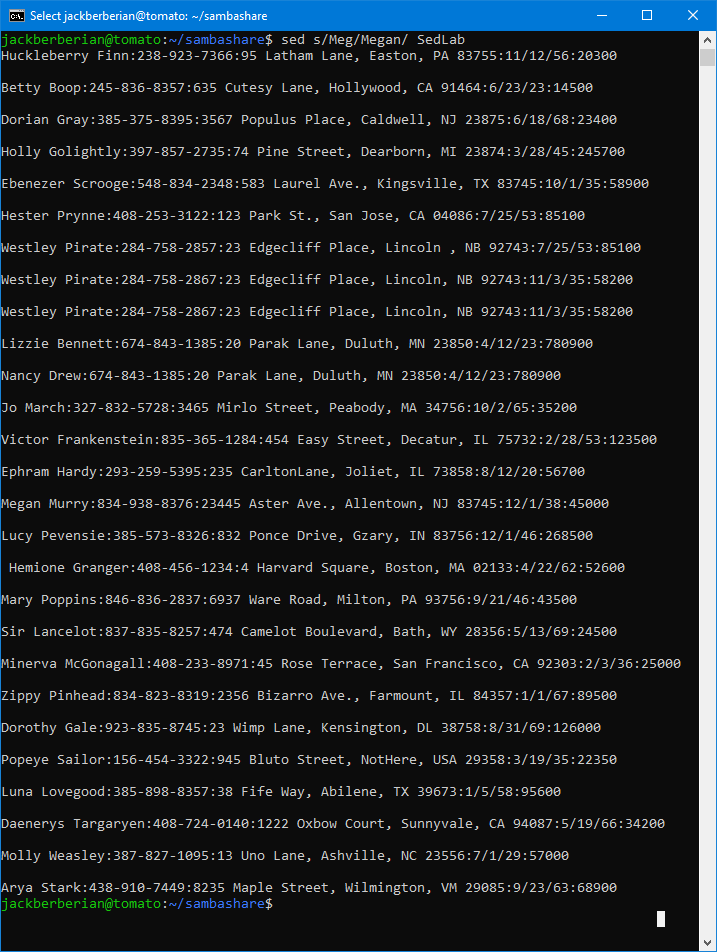
1. Change the name Meg to Megan

The command we need to enter is: sed s/Meg/Megan/ SedLab

Here, sed is going to substitute Meg for Megan because of the s/ (substitution) command in SedLab. A better way to understand it is:

Sed substitute/word to substitute for/word that will be substituted/ file

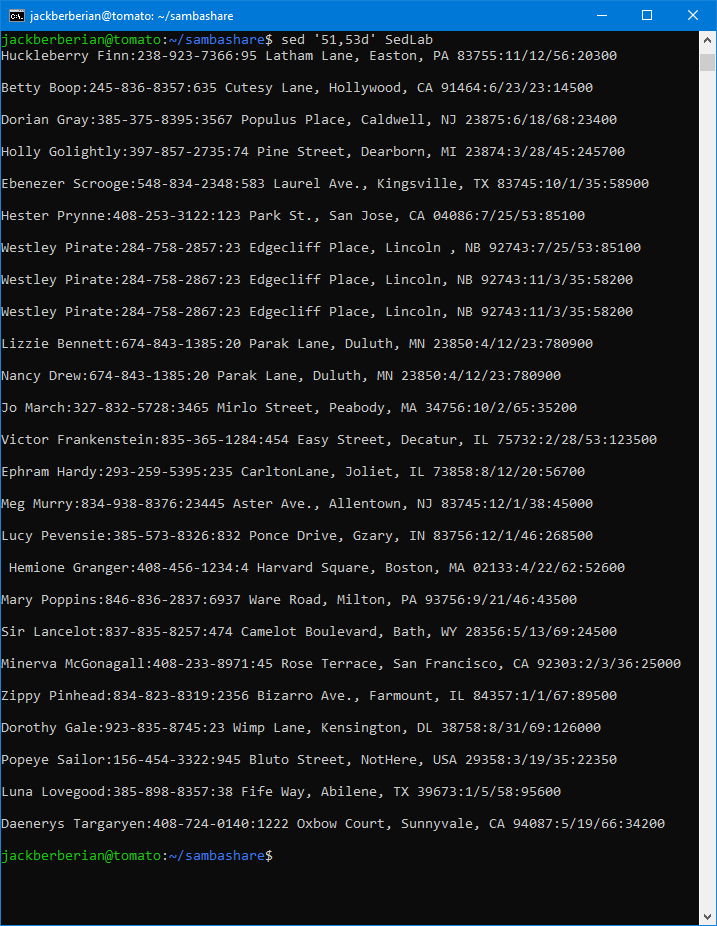
Sed s/Meg/Megan/ SedLab



1. Delete the last 3 lines

The command that we need to enter here is: Sed ’51,53d’ SedLab

What this does here is, sed will delete the last three lines which are 51-53 (excluding the white space). D is the signal to the program that the lines should be deleted, and 51,53 means it’ll go through those lines.

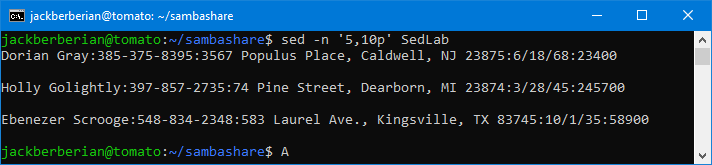


1. Print lines 5-10.

The command that we need to enter here is: Sed -n ‘5-10p’ SedLab

Here, sed will only print the lines that were changed because of the -n flag and then print lines 5-10 in the file SedLab because of the p signal. So if you wanted to just print lines 23-42, then you could do

Sed -n ’23-,42p’ SedLab



1. Delete lines for people who live in California

The command that we need to enter here is: Sed ‘/CA/d’ SedLab

Sed pattern matches for people who live in California because it’s inside the first dashes, and then deletes it because of the d. So for example, if you wanted to find people that live in Massachusetts and then delete it, you’d do sed ‘/MA/d’ SedLab but do keep in mind that the / / is case sensitive.



1. Print all lines where the birthdays are in the first week of the month. Be careful of the dates for birthdays, the format is MM/DD/YY.

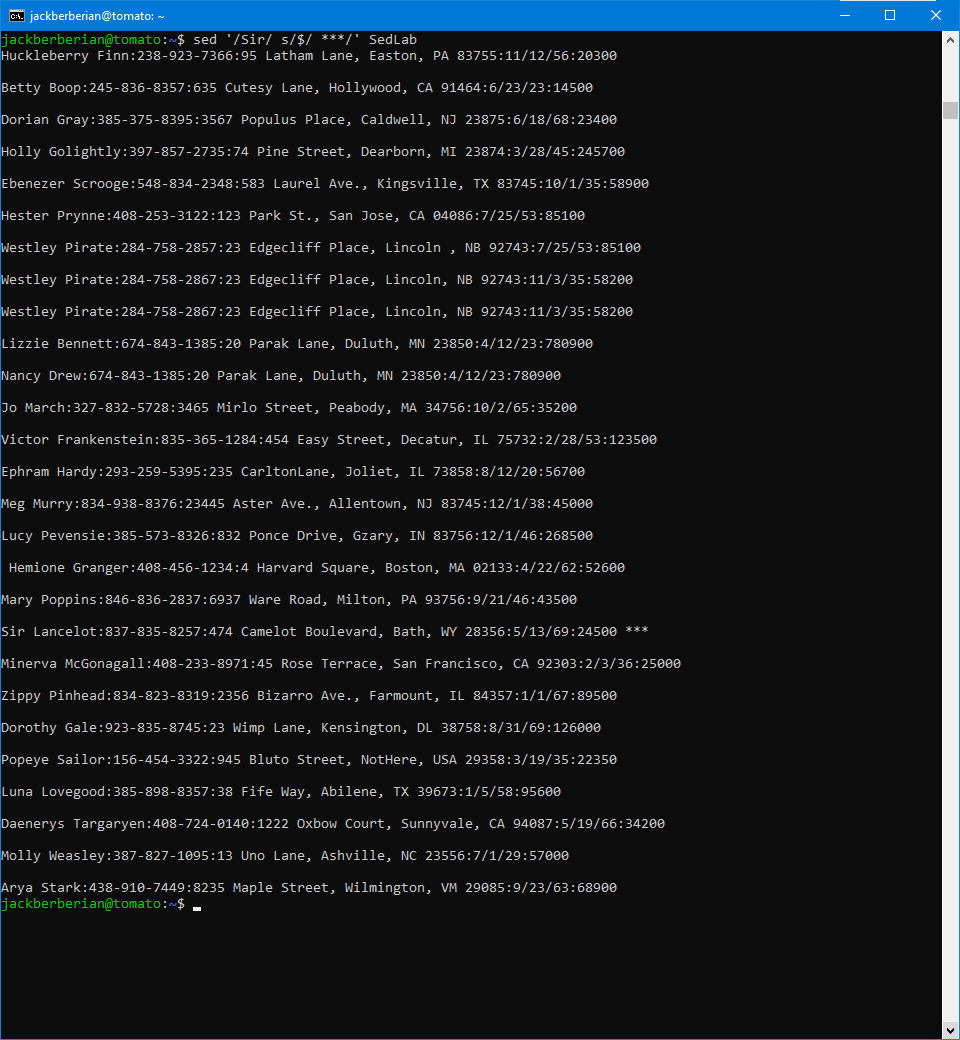
The command that we need to enter here is: sed -n '/\/[1-7]\// p' SedLab

What this regular expression does is search for dates that are in the first week of the month. Using the format of :MM/DD/YY:, telling sed to search for the numbers 1-7 in between two /, it’ll find birthdays in the first week of the month.

1. Append three asterisk to the end of lines starting with Sir.

The command that we need to enter here is: sed '/Sir/ s/$/ \*\*\*/' SedLab

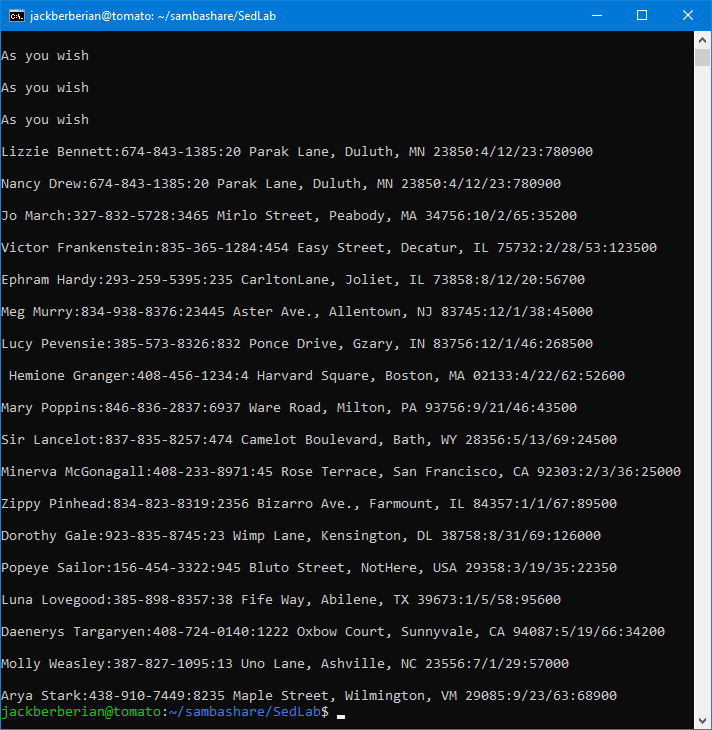
Here, sed searches for lines with Sir, and then appends \*\*\* to the end of the line because of the $ command.



1. Replace the line containing “Westley Pirate” with the phrase “As you wish”. Make sure you replace the whole line not just the name.

The command that we need to enter here is: Sed ‘s/Westley Pirate.\*/As you wish/’ SedLab

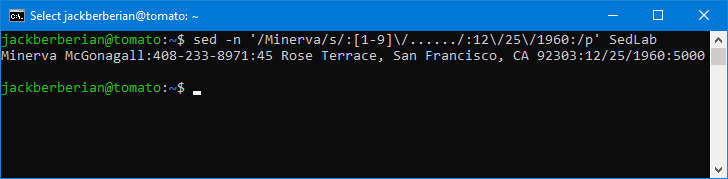
Here, we can replace the whole line containing Westley Pirate with As you Wish. Instead of just substituting Westley Pirate for As you wish, if we add .\* to the end, it’ll look for the whole line.



1. Change Minerva McGonagall’s birthday to 12/25/1960. Assume you don’t know Minerva’s original birthday. Use a regular expression to search for it.

Here, sed will search for lines that contain Minerva, and then replace the text in the brackets after /s/ until the / before :/12 which whatever comes after it until the final / which in this case is /:12\/25\/1960:/. Anytime there is a /, sed requires the / to be escaped with a \ as well.

The command that we need to enter here is: sed -n '/Minerva/s/:[1-9]\/....../:12\/25\/1960:/p' SedLab

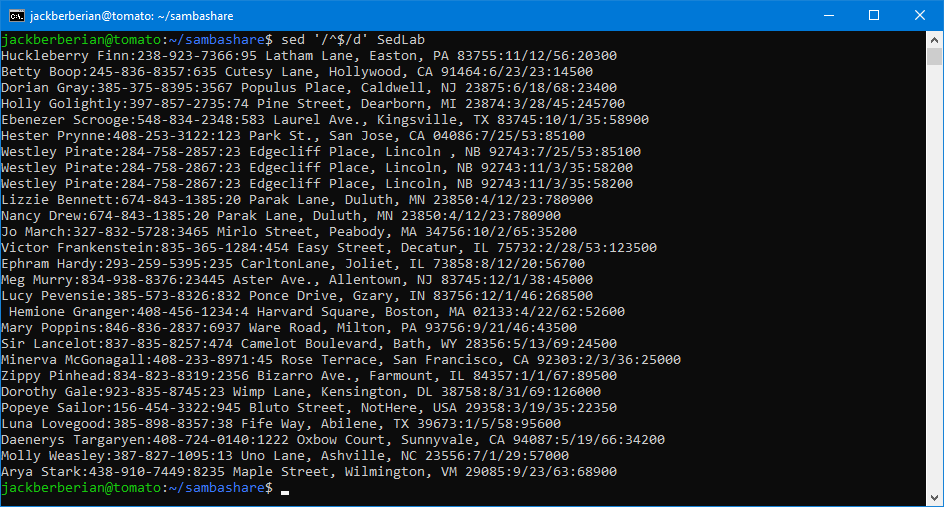


1. Delete all blank lines.

The command that we need to enter here is: sed '/^$/d' SedLab

Here to delete all the blank lines, we can use: sed '/^$/d' SedLab. What this does is look for all the blank lines and delete them.

^ is a ‘beginning of line anchor and $ is a end of line anchor. So since there is nothing (blank) in between the lines, it simply looks for lines with nothing, which would be the blank ones.



1. Write a sed script that will (actual sed script, not just the commands on the command line)
   1. Insert above the first line the title – Great Literary Characters

What this does is append above the first line, the title.

‘1 i\Great Literary Characters’

* 1. Remove the duplicate lines

See script.

* 1. Print the contents of the file with only the last name then the phone number

What this does is use a regular expression to go replace each line with an output of LName:Phone-number-number. It only searches for the characters A-z that are attached to a semi-colon and phone number.

s/.\*\([A-Z][a-z]\*:...-...-....\).\*/\1/

* 1. Append at the end of the file “Happily Ever After The End”

This appends at the end, Happily Ever After The End

Sed ‘$a Happily Ever After the End’ SedLab

Citations:

<https://stackoverflow.com/questions/15668815/removing-duplicate-lines-with-sed-in-batch>

<https://www.grymoire.com/Unix/Sed.html#uh-19>

<https://flylib.com/books/en/4.356.1.40/1/>

<https://www.gnu.org/software/sed/manual/html_node/sed-commands-list.html>

To run the sed script:

First add #!/bin/sed -f to indicate it’s a sed file at the top and then enter your commands below. (Screenshot will demonstrate.)

To run, enter: sed -f (nameofscript.sh) (name-of-file-to-test-on)

Example: sed -f removeduplicates.sed SedLab

Text

Description automatically generated

Another way to remove duplicates would be to use sort and use that to display only the unique lines with the flag, sort -u filename. Then you could pipe the output to SedLab and run the other commands.

Examples:

Sort -u SedLab or sort -u SedLab | sed -f removeduplicates.sed

Text

Description automatically generated

Or sort SedLab | uniq -u

Text

Description automatically generated