Ethan Kent

Lab Assignment 3: Sed lab

Sed is what is called a “stream editor” which is used to make quick changes to files, whether it be changing, adding, or deleting things. Though Sed won’t do these as a permanent change, it will just show you what the modification that you did will look like. If you wish for it to be saved you can append the results to a different file or overwrite the original file with the changes. I will be going over some prompts to show some of the ways to use Sed.

Prompt 1: We have an entry in this file with the name Jo, we need to change the name to Josephine. To do this we will be using this command, sed -n s/Jo /Josephine /p filename. Now what I did was use “n” to tell sed not to print everything once we finish the edit, I used “p” to tell sed to only print what was actually effected. I also added spaces after so that I would look for just the instant of the name Jo, since we also have some other instances of “Jo” in addresses. The spaces are also to make the line look prettier. The thing telling sed to change is the “s” a substitution.



Prompt 2: We want to delete the last 5 lines of our file. Here is what the end of the file currently looks like.

Text

Description automatically generated

I used “wc -l filename” to count how many lines are in the file so we can find the range we need to delete. The command I used to get the results is, sed ’48,53d’ filename. Now the “d” here is what is telling sed to delete and the “48,53” is saying to delete that range of lines. Which will give us.

Text

Description automatically generated as you can see Luna Lovegood is now the end of our file.

Prompt 3: We want to print the lines three through fifteen in our file. To do so I’ll use a range, sed -n “3,15p” filename. I’m using “n” so that nothing else will print. The “3,15p” is saying “Look for lines 3 through 15” and the “p” saying Text

Description automatically generatedthat “Print these range of lines”

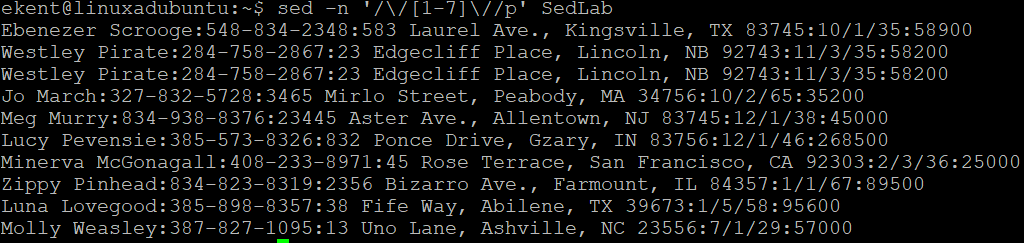
Prompt 4: Let’s get rid of California… From our list of people on file. We don’t have to do too much with this one, we simpley need to use the “d” command again but this time we will be giving it a pattern to get rid of. To do so I used sed “/CA/d” filename.

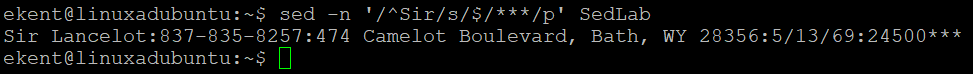
Text

Description automatically generated

Those large gaps in between our other lines are where our California friends used to be. By adding “d” to the end of the pattern we told sed to delete any instances of it that we saw.

Prompt 5: Let’s search for some birthdays, though we only need to look for the first week of birthdays. So we want to find anyone who’s birthday is between the first and seventh of the month they were born. To do this I used sed -n '/\/[1-7]\//p' filename. Now I know that looks rough but let me break things down a little. The “\” backslash is what is called an escape character, we need to use it so we can search the file for “/” since the birthdays are divided with them. The escape means that we aren’t using “/” like we normally do in sed, we are searching for it as a character in the lines. It does get a bit confusing after a bit though having so many escape lines so just take it slow and make sure you know where things are escaped and others are for the command!.



Prompt 6: Let’s append something to the end of a line by finding what a line starts with. In this case we are looking for all lines starting with “Sir” and appending three asterisks to the end of this line. To do so I used sed -n '/^Sir/s/$/\*\*\*/p' SedLab. 

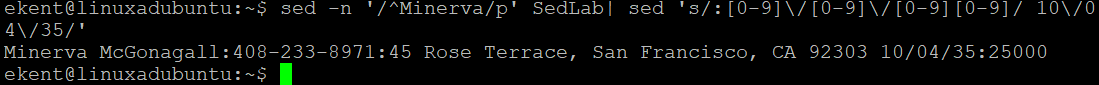
It looks a little daunting but it’s mostly just a long line. By using “^” I’ve asked to look at the start of a string for “Sir” then our “s” for our next part needs to be substituted, the “$” is telling sed to do this at the end of the matching line, our string we want appended “\*\*\*” and then to print it once it’s done.

Prompt 7: Change “Westley Pirate” with “As you wish”. Instead of just changing the name though we are going to make it so the whole line is replaced. To do so we will be using “c”, c will change the lines that it finds that match what we enter so for this I will use, sed “/Westley Pirate/c As you wish” filename.

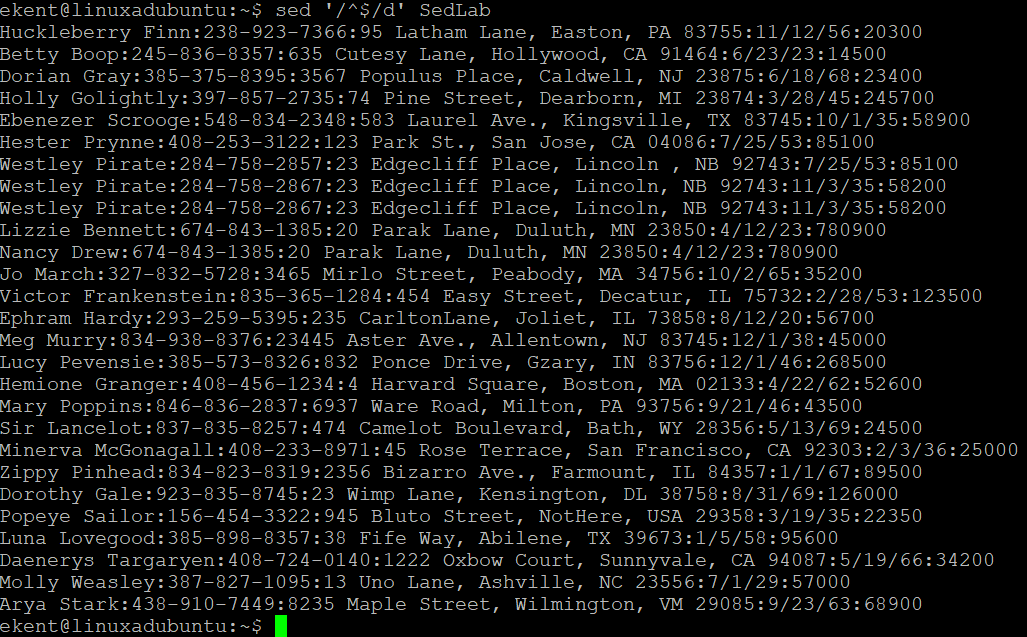
Text

Description automatically generated All instances of Westley are now changed.

Prompt 8: We need to go and change a birthday in a file. We don’t know their current birthday on file but we do have their name, Minerva McGonagall. So to find the line containing her information and then change it we can use the “|” to do two things a bit easier. Here is what I used, sed -n '/^Minerva/p' SedLab| sed 's/:[0-9]\/[0-9]\/[0-9][0-9]/ 10\/04\/35/' It looks long and daunting for sure but that’s mostly since we had to use some many “\” escape characters when looking for the birthday. The first half before the piping “|” is asking “Find a line that starts with Minerva” the second half we are asking it to substitute “s”. The way it’s broken down is “Find a “:” then look for a range of numbers followed by a “/” and then look for another range of number, followed by a “/” and then find two ranges (this meaning the year at the end) and have them replaced with 10/04/35” but we had to use “\” inbetween all the “/” so sed knew to look for a forward slash.



Prompt 9: Blank space takes up room and we want to get rid of it. Let’s use the command, sed ‘/^$/d’ filename and get rid of all of it. To my understanding the way this command works is since we have both “^” the start of a line and “$” end of a line with nothing between them we have no particular pattern, we have nothing at all so this is what tells our command that we want to get rid of everything where there’s nothing.



Prompt 10: Doing everything on one line can be tiresome, we have the ability to write what we want done in a script. To do so we can go into a text editor in this case nano. To make a new one just type nano then what you want the files name to be. We want to add to the top and bottom of this file, a title and an ending. The 1 tells us where we want to add a line in the file, in this case to the top as our title then the “i\” will append it there. The middle section was an attempt to print the last names with phone numbers by using regular expressions. The idea was to look for a capital letter followed by any amount of lower case then a colon since this is what divided the last name and number the rest is asking for a range of numbers from 0 to 9 to look for then the dashes since that’s how the phonenumber is written out. Then the /\1 to register it as a pattern. I tried to add dashes “/” or move around things but I consciously got errors and couldn’t figure out if my regular expression was wrong, syntax, or even if I was just using sed wrong with them. Over all the last line works though. The “$a” says to put this line at the end of the file.

Text

Description automatically generated

Then below is the file printed out with the title and ending sentence after I comment out the middle part. I also had to change the “-nf” to “-f” since the rest of the name wouldn’t print due to the n’s function.



Sources used for lab

<https://phoenixnap.com/kb/linux-sed>

<https://www.golinuxhub.com/2017/06/sed-replace-whole-line-when-match-found/>

<https://stackoverflow.com/questions/13202715/sed-get-lines-beginning-with-some-prefix>

<https://stackoverflow.com/questions/15978504/add-text-at-the-end-of-each-line>

<https://www.cyberciti.biz/faq/using-sed-to-delete-empty-lines/>

<https://unix.stackexchange.com/questions/99350/how-to-insert-text-before-the-first-line-of-a-file>

<https://unix.stackexchange.com/questions/31947/how-to-add-a-newline-to-the-end-of-a-file>