Nathan Christensen-Padilla

Adrianna Holden-Gouveia

Linux Administration

10/7/2021

Sed

1. Code: sed -n s/Meg/Megan/gp datebook



I used the sed command and the first thing I write after that is -n so it only outputs the printed lines, and then I put s which makes the command substitute whatever I put first with whatever comes next, and that’s how it turned Meg into Megan. Also, I added a g at the end, so it replaces every Meg in the line with Megan (although there was only one instance of meg in this datebook) and p to print out the changed lines.

1. Code: sed '$d' datebook | sed '$d' | sed '$d' >> datebook\_output

Graphical user interface, text

Description automatically generated

I used ‘$d’ to delete the last line of datebook, and it does this because the dollar sign ‘$’makes the command start at the end of the file and the ‘d’ deletes lines, so combined it deletes the last line of the file. So, I piped the same sed command two more times which gets rid of the last three lines of the datebook file.

1. Code: sed -n '2,4p' datebook >> datebook\_output



I used sed then -n which when used with ‘p’ makes it print only the lines I want which were ‘2,4’ also known as two through four.

1. Code: sed '/Place/d' datebook >> datebook\_output

A screenshot of a computer

Description automatically generated with medium confidence

I used sed and then I typed ‘/Place’ which makes the command target everything with Place in the line and then ‘/d’ to delete those lines.

1. Code: sed -n ‘/\/[1-7]\//p’ datebook >> datebook\_output Text

   Description automatically generated

First, I used the command sed then ‘-n’ so it doesn’t print the whole file, and then I put ‘/\/’ and the backslash allows the forward slash to work properly by turning it into a normal character instead of a special character because we need that second forward slash later in the command. After that I put a [1-7]’ which finds any number that’s between 1 and 7 including the numbers themselves. And finally, I put another ‘\/’ so the command in the end searches for numbers that start with a forward slash, a number between 1-7 and lastly end with another forward slash. The reason I did this was, so it finds lines where peoples birthdays are in the first week of the month.

1. Code: sed -n ‘/Sir/s/$/\*\*\*/p’ datebook >> datebook\_output



I used the command sed then ‘-n’ so it doesn’t print the whole file, then I put ‘/Sir/s/$/\*\*\*/p’ which first finds lines that have Sir, then the ‘s’ makes it substitute the ‘$’ last part of the line with ‘\*\*\*’ and finally prints all of those lines.

1. Code: sed ‘/Westly Pirate/c\ As you wish.’ datebook >> datebook\_output

A screenshot of a computer

Description automatically generated with medium confidence

First, I use sed then I put ‘/Westly Pirate/c\ As you wish.’ Which first finds all lines that have Westly Pirate in them then it uses the ‘c’ command which replaces the line with whatever is after the backwards slash in this case ‘As you wish’.

1. Code: sed -n ‘/Minerva McGonagall/{s/[0-9]\/[0-9]\/[0-9][0-9]/12\/25\/1960/p;}’ datebook >> datebook\_output

First I used sed then ‘-n’ so it only prints out this single line, then I wrote ‘/Minerva McGonagall/{s/[0-9]\/[0-9]\/[0-9][0-9]/12\/25\/1960/p;}’ which first finds the lines with Minerva McGonagall and if a line is found then it activates the next part of the code which substitutes the birthday on that line with a new one that I specified, also once again I used backslash and forward slashes ‘\/’ so I could search for the forward slash character.

1. Code: sed ‘/^$/d’ datebook >> datebook\_output

A screenshot of a computer

Description automatically generated with medium confidence

I start the command off with sed then I type ‘/^$/d’ datebook >> datebook\_output which first find lines that begin and end with nothing then deletes them and as shown in the picture above it got rid of all the empty lines.

1. Run command: sed -f LiteraryCharacters.sh datebook >> datebook\_output

Code: I attached the code in a text file, which contains all of the comments on the code, and I also added the screen shots of the output and the code below. (Also I had to change the .sh of the literary characters file cause the file uploaded rejected that file type.)

Text

Description automatically generatedText, letter

Description automatically generated

Sources Used: Your YouTube video and the book helped me solve basically everything.

<https://flylib.com/books/en/4.356.1.49/1/>

<https://youtu.be/YMqOocY0ovs>

My discord friend who works with bash programming all the time, I only asked about commands that I could use when trying to do something and a good way to bash script, but I wrote everything myself.