1. Change the name Jon to Jonathan.

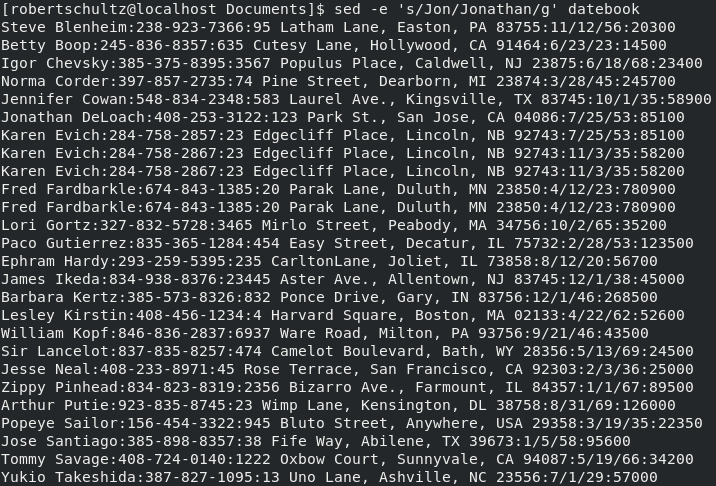
sed is used to make changes to files

There are commands for appending, deleting, substituting, and printing. There are more commands but those four are the easiest to describe with a single word.

Using 's/.../.../' tells the computer to make a substitution.

In the first space we put what pattern we want to find and in the second space we put the pattern we want to replace it with.

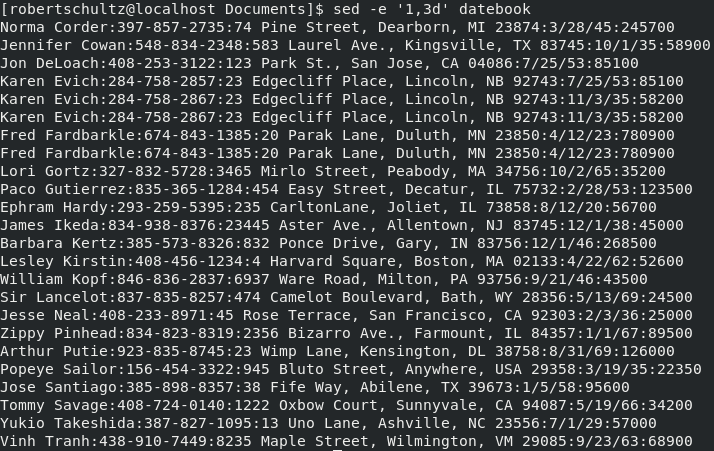
sed -e 's/Jon/Jonathan/g' datebook



2. Delete the first three lines.

Using '...,...d' we can tell the computer to delete a number of lines. Saying '1,3' tells the computer to delete the lines in the range 1-3.

sed -e '1,3d' datebook

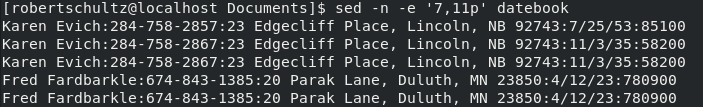


3. Print lines 7 through 11.

Using the print command we tell the computer to print lines 7 through 11. The print command is used the same way as the delete command. '...,...p'

Normally, the entire file would be printed, but the '-n' option prevents printing unless it is specifically requested.

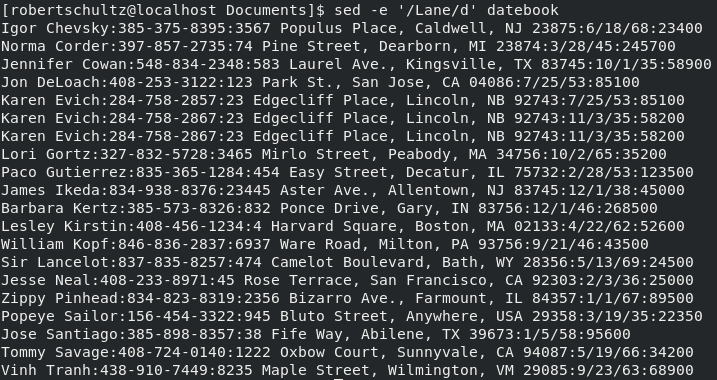
sed -n -e '7,11p' datebook



4. Delete lines containing Lane.

Same idea as 2 but instead of targeting specific lines to delete, we tell it look for the string 'Lane' and delete only the lines that contain that string.

sed -e '/Lane/d' datebook



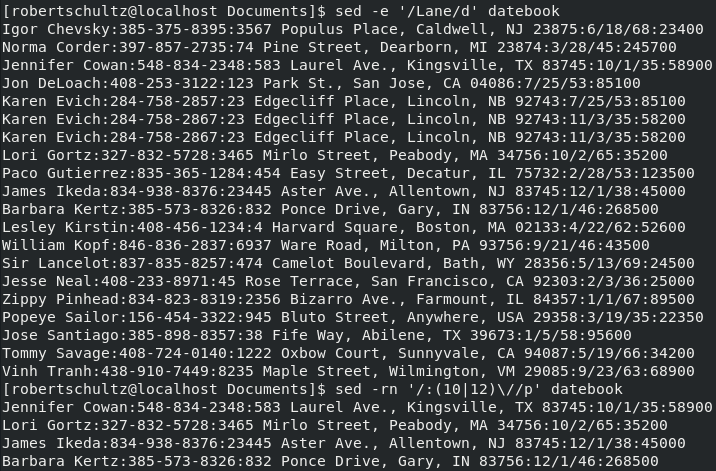
5. Print all lines where the birthdays are in October or December.

The '-r' option indicates the use of an extended regular expression. We also use '-n' to prevent printing unless we ask for it.

':(10|12)\/' tells the computer to look for ':10/' or ':12/'.

We then use the print command to print only lines that contain one of those patterns.

sed -rn '/:(10|12)\//p' datebook



6. Append three asterisks to the end of lines starting with 'Fred'.

Using 'a\' we tell the computer to append new characters after a line containing our pattern.

sed -e '/Fred/a\\*\*\*' datebook



7. Replace the line containing 'Jose' with 'JOSE HAS RETIRED'.

Using the substitution command again, only we replace the entire line that contains the pattern instead of just replacing the pattern.

Including '.\*' after 'Jose' tells the computer there will be an undefined number of characters before and after 'Jose'. It can even be zero characters.

sed -e 's/^Jose.\*/JOSE HAS RETIRED\./g' datebook



8. Change Popeye's birthday to 11/14/99. Assume you don't know Popeye's original birthday. Use a regular expression to search for it.

First, we have the computer look for the pattern 'Popeye'. Once that pattern is found, we tell it to make a substitution on that line.

Since we don't know the original date we are substituting, we tell the computer to look for numbers separated by '/'. Using '[0-9]', we tell the computer to look for any number.

The '\+' tells the computer that there may or may not be a second digit to look for.

We can't use '/' as a separator, so we use ',' instead.

sed -e '/Popeye/s,[0-9]\+/[0-9]\+/[0-9]\+,11/14/99,' datebook



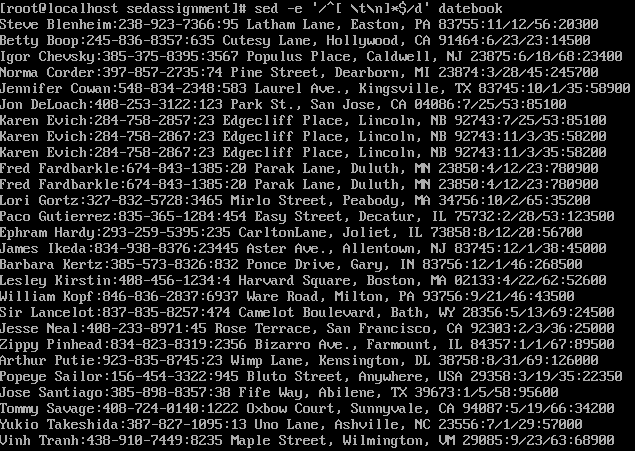
9. Delete all blank lines.

This one, again uses the delete command, only we specifically look for empty lines.

The presence of '^' and '$' tells the computer that the pattern we are looking for covers the entire line.

Using '[ \t\n]\*' we tell the computer the pattern is an indefinite amount of white space characters. This amount can even be zero.

sed -e '/^[ \t\n]\*$/d' datebook



10. Write a sed script that will

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

b. Remove the salaries ending in 600.

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.

create script

This creates a file called script with the extension .sed.

The file is opened in vi and you can start writing the script.

vi script.sed

make script executable

This is used after the script has been written to give the permissions to execute.

chmod +x script.sed

run script

sed -f script.sed datebook

PERSONNEL FILE above first line

We use '1i\' to tell the computer we want to add a new line at the beginning. What follows is the content of that new line.

By adding the '1' at the beginning, we tell the computer to only do this on the first line.

1i\PERSONNEL FILE

remove salaries ending in 600

Using the delete command again. This time we tell the computer to look for lines that end with '600'. The use of '$' tells the computer that the pattern we are looking for is at the end.

/600$/d

reverse first and last names

Again we use the substitute command. Every line starts with first name, so we can use '^' to tell the computer where the pattern begins.

The first and last names are separated by a space, so we account for that with '[ \t]' and after the last name there is a ':'.

We don't know how long each name is so using '\*' we can tell the computer that these patterns appear in varying lengths.

After that '.\*\' tells the computer that a number of characters will follow but we do not know how many.

Using '$' we tell the computer that the line will end after that group of characters.

The data is now broken into four groups. 1 is the first name, 2 is a blank space, 3 is the last name, and 4 is everything else

All we want to do is switch first and last names, so we substitute 1 for 3 and 3 for 1.

Pretty sure there is a better way to do this. There may not be, but I think this is too complicated for it to be the only way.

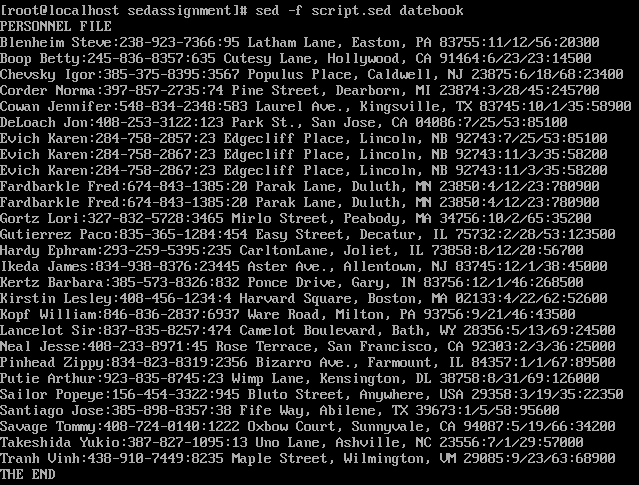
s/\(^[A-Za-z]\*\)\([ \t]\*\)\([A-Za-z]\*\):\(.\*\)$/\3\2\1:\4/g

THE END at the end

We use append again, except we add '$' to tell the computer we want this to happen only at the end.

Without '$', THE END would be added after every line.

$a\THE END



Most of what I used was found in Sed - An Introduction and Tutorial - by Bruce Barnett.