Assignment 2

The Grep Family

By Nicholas Mueller

Sources used “Regular Expressions in Grep (Regex)”

Link to article is here <https://linuxize.com/post/regular-expressions-in-grep/>

Step 1: Download the file I put it in my home folder so it’s easier to work with shown below.

I used the built in Firefox browser to simply login to necc.mass.edu then downloaded the file.

Now datebook is in my home folder/directory.

A screenshot of a computer

Description automatically generated with medium confidence

Please include the command, a screenshot showing it works as intended, cite all sources you used, and give a short explanation of how the command works and why.

1. Print all lines containing the string Street .

This command works because I’m searching for the pattern Street and grep has found matches to the string.

Note the quotation marks are not required for this problem.

Command: grep "Street" datebook

Text

Description automatically generated

1. Print all lines where the person's first name starts with M.

This command includes a regex expression the ^ when entered into grep the expression finds any line that starts with M the ^ means starts with and is used in front of the pattern.

Command: grep ^M datebook

Text

Description automatically generated

1. Print all lines ending in 000 .

The command works in this instance because of another special character when I enter the pattern 000$ the $ at the end tells grep to look for any line with 000 on the end the $ means end and is used after the pattern is entered.

Command: grep 000$ datebook

Text

Description automatically generated

1. Print all lines that don't contain 408 .

This command uses the inverse search or -v option that grep has, so when I enter grep -v 408 datebook grep looks for lines without the specified pattern 408 in this case.

Command: grep -v 408 datebook

Text

Description automatically generated

1. Print all lines where birthdays are in the year 1923 .

For this regex expression I am using the existing structure of the datebook file to search it the command means -E extended regex [[:digit:]] means I’m looking for one digit then a slash / another 2 digits another / then 23.

Command: grep -E [[:digit:]][/][[:digit:]][[:digit:]][/]23 datebook

Text

Description automatically generated

1. Print all lines where the phone number is in an area code that starts with an 8

For this problem I’m using 8 followed by any two digits and a – mark in the regex expression

Command grep -E 8[[:digit:]][[:digit:]][-] datebook

Picture on next page

Text

Description automatically generated

1. Print all lines containing an uppercase letter, followed by 5 lowercase letters , a comma, and one uppercase letter.

For this command I use extended regex with the -E option the [[:upper:]] means 1 uppercase letter, the [[:lower:]] means a lowercase letter with {5} I specify I want it to be five lowercase letters the , is a literal , it matches a , in order to get this to work I had to use a [[:space:]] then another [[:upper:]] because in this file the comma and next uppercase letter are separated by a space.

Command: grep -E [[:upper:]][[:lower:]]{5},[[:space:]][[:upper:]] datebook

Graphical user interface, text

Description automatically generated

1. Print lines where the address begins with a two or three digit number (so this would be 12 main st or 123 main street but not 1234 main street).

This command uses grep -E for extended regular expressions the code means that I want a colin then a digit repeated 2 times minimum 3 times maximum then a space and then the . means any character. This gives me 2 digit or 3 digit addresses, but no 4 digit ones.

Command: grep -E ':[[:digit:]]{2,3}[[:space:]].' datebook

Text

Description automatically generated

1. Print lines preceded by a line number where the person is from Massachusetts (or MA)

For this problem I use the grep -AN option where N is the number of times to show lines that come after the matching MA pattern. Due to blank lines it was necessary to use the 2 lines option in order to get the following line to show.

Command: grep -A2 “MA” datebook

Text

Description automatically generated

1. Print lines containing  an address that doesn't include Street or St

For this problem I use a grep inverse search for Street this omits the addresses with Street then I piped the grep command again and used another inverse search for St which results in all addresses that don’t have either of these patterns.

Command: grep -v Street datebook | grep -v St

Text

Description automatically generated