**Joseph Williams**

**Grep Lab**

**Instructions:**

Grep is a Unix/Linux command line utility that assists the user to locate, filter, and sort strings within a text file. For this lab, we will be using grep commands to search for a pattern of characters in a file or several files. The grep utility can also pick information from a file by searching for the user-specified pattern in the entire filesystem.

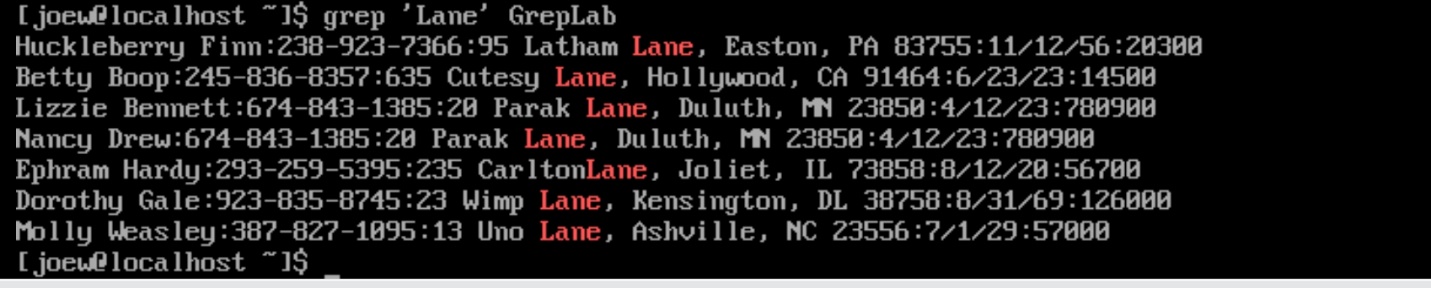
Before we begin, the first step is to launch the terminal. Then, change the directory in Ubuntu or Centos server to where the GrepLab file is stored. In my case, the file is located within the server home directory. I will be using CentOS server for this lab, you can use any Linux distro. The command “*cd*” have me navigate to the file GrepLab.

**Lab Results (Screenshots and Descriptions)**

Below you will find the commands in *Italic writing*and a brief description. You will learn and be able to write grep commands independently on your server.

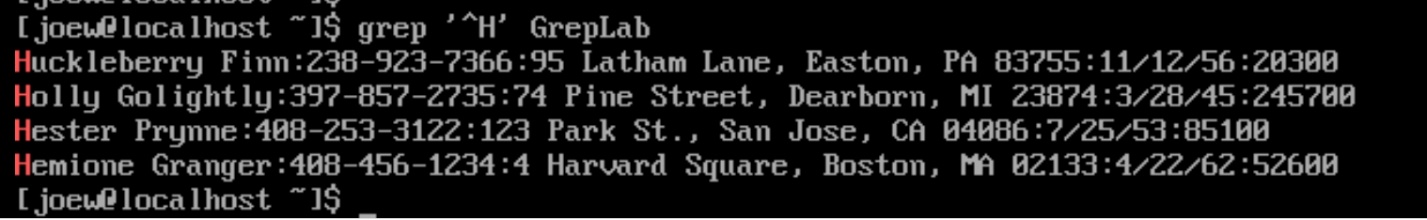
1. Print all lines containing the string Lane.

In this exercise you can see that the command *grep “lane” GrepLab* can be used to print out all lines containing string Lane as shown in the image below.



1. Print all lines where the person’s first name starts with H

The Grep string will search for the following using the pattern: *grep '^H' GrepLab,* we can print out all lines where the person’s name starts with H. The caret (^) indicates the start of a line and using ‘^H’ displays all the lines starting with H.



1. Print all lines ending in three zeros (000)

The command *grep '000' GrepLab* displayed all the statements ending with 000*.* Grep’s dollar sign ($) represents the end of a line.

A screenshot of a computer

Description automatically generated with medium confidence

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

This search pattern: *grep -v -e '408' GrepLab* can be used to display all the lines that do not contain 408 phrases in their statement. (-e) is match control tool argument which lets grep look for the defined lines. On the other hand, -v is used to return line that do not match. Therefore, grep -v -e ‘phrase’ can be interpret as “do-not-match this ‘phrase’ filename”.

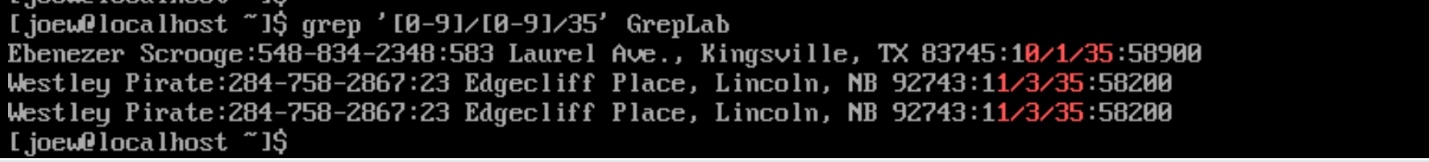


Text

Description automatically generated

1. Print all lines where birthdays are in the year 1935 (be careful of the date format! It’s MM/DD/YY)

We can use the expression: *grep ‘ [0-9]/[0-9]/35’ GrebLab* to find all records of people birthdays in the year 1935.



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

We can also use the expression pattern: *grep -E ':8[0-9]{2}-[0-9]{3}-[0-9]{4}:' GrepLab,* to print all numbers in an area code that starts with 8. As you can see below the expression can be translated to match 8 and then look at the next two numbers between using 0 to 9, proceed to describe the next three digits and then proceed to find the final four numbers using 0 to 9 as the criteria.

A screenshot of a computer

Description automatically generated with medium confidence

1. Print all lines containing an uppercase letter, followed by 4 lowercase letters, a space and one uppercase letter.

The statement *grep '[A-Z][a-z]\{4\} [A-Z]' GrepLab* can be used to display all lines containing a word starting with an uppercase letter, then followed four lowercase letters, space and a final uppercase character. The grep command starts by evaluating if the first letter is in Uppercase, if true the grep utility proceeds to analyze the next fifth lowercase letters and so fifth.

Text

Description automatically generated

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

The grep search command: *grep -E '[0-9]{3} [A-Z].\*[a-z]' GrepLab* can be used to print addresses containing 2 or 3 digits. The grep command can also be interpreted as output phrases with 2 or 3 digits in the address section and then follow by the street name.

Text

Description automatically generated

1. Print lines where the person lives in Mass or Illinois

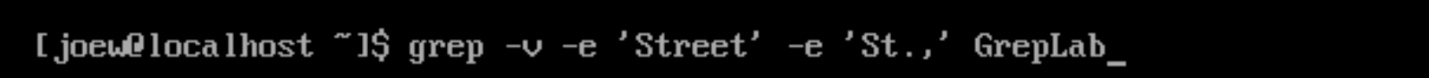
The statement *grep ‘MA’ GrepLab && grep ‘IL’ GrepLab* can be used to print out addresses from Massachusetts and Illinois. (&&) separate the expression independently, and to allow another statement.

Text

Description automatically generated

1. Print lines containing the address that aren’t on a street (You might see St as shorthand for street)

The grep search command: *grep -v -e ‘Street’ -e ‘St.,’ GrepLab,* will display all the lines that do not contain phrases such as Street or St in their line. The grep command only prints out sentences without the words such as St or Street.



Text

Description automatically generated

**References:**

Bambenek, J., & Agnieszka, K. Grep: *pocket reference. O’reilly Media.*

[*https://flylib.com/books/en/4.356.1.26/1/*](https://flylib.com/books/en/4.356.1.26/1/)