Ethan Kent

Lab 2, Grep

Grep is a program that can help searching through directories and files or contents of a specific file as well. It can even search multiple files for a specified string that you are looking for. I’m going to be using grep to look through a single file, but I will be showing how I do it and what I am doing.

Prompt 1. This time I will look for every instance of the string “Lane” specifically capitalized as grep and Linux are case sensitive. To find this I will simply type grep Lane filename. To break it down we use grep at the start so we can begin the command, Lane is what we want to search for and then we have to tell what file we are going to search through.

Text

Description automatically generated Here are the results!

Prompt 2: I have to find all the names that begin with an H for this so to do so I will be using grep ^H filename. the “^” is used to tell that I want to look at the start of each string for a capital H, since we know the format of this file begins with the name of each person.

Text

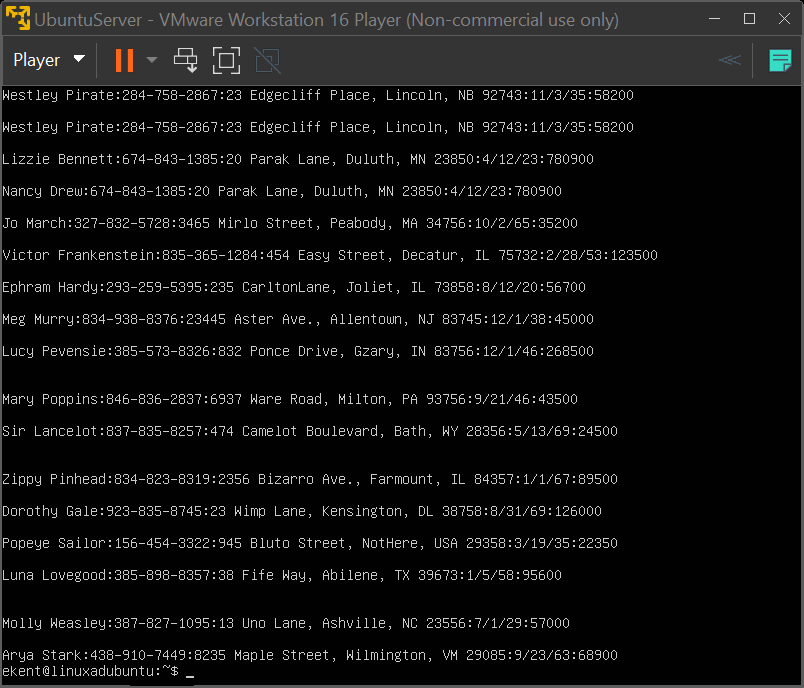
Description automatically generated

Prompt 3: What about things that end in a certain way? For this I’ll find everything that ends in 000. To find things that are at the end of something we can use the $ sign at the end of our search, so we will use grep 000$ filename.

Text

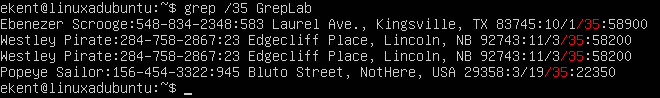
Description automatically generated

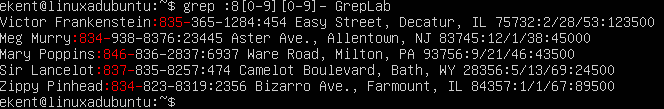
Prompt 4: Sometimes you want to exclude things from a search and just find everything besides that one thing. For this we don’t want to find any area codes that are 408. We can use the -v flag to tell grep to exclude anything with 408 in it. grep -v 408 filname



Now we have everything without 408.

Prompt 5: knowing how something is formatted helps the search a lot. We know the birthday system is day/month/year, and it uses the forward slash. Since nothing else uses that we can do, grep /35 filename. There we have all the birthdays born in 1935.



Prompt 6: For this example, we want to find all the area codes that start with 8. To do this I’m using, grep :8[0-9][0-9]- filename. For this I’m using the ranges [0-9] twice so that it will only look for two digits after the 8. I’m also using the dash as a way to distinguish phone numbers in this search and the colon since that is what is used before the beginning of the area code

Prompt 7: Now we want to find a length of a string that is six characters long and starts and ends with a capital, so something like “GivinG” but in this file we don’t have any words like that, so we have to think a little bit deeper. We can use something like “[A-Z]” at the start because that will look for any capital letters from a to z but then we can use the opposite “[a-z]” to look for all the lower case. The command is a bit lengthy but this is what I will be using, grep “[A-Z][a-z][a-z][a-z][a-z] [A-Z]” in between the last lower case range and the upper case one at the end I used a space. This way we can find another uppercase character after the end of our previous one.Text

Description automatically generated

Prompt 8: Let’s find some addresses based on the length of the address. In this case we want to find two-to-three-digit addresses. To do so I started with a colon “:” since each piece of info is divided by one in this file. After that I’m going to use ranges “[0-9]” twice, though we need to look for two things so I will use and or “\|”. The command will look like this, grep “:[0-9][0-9] \|:0-9][0-9][0-9] “ Notice I’m using spaces after so that it won’t grab things like the beginning of phone numbers.

Text

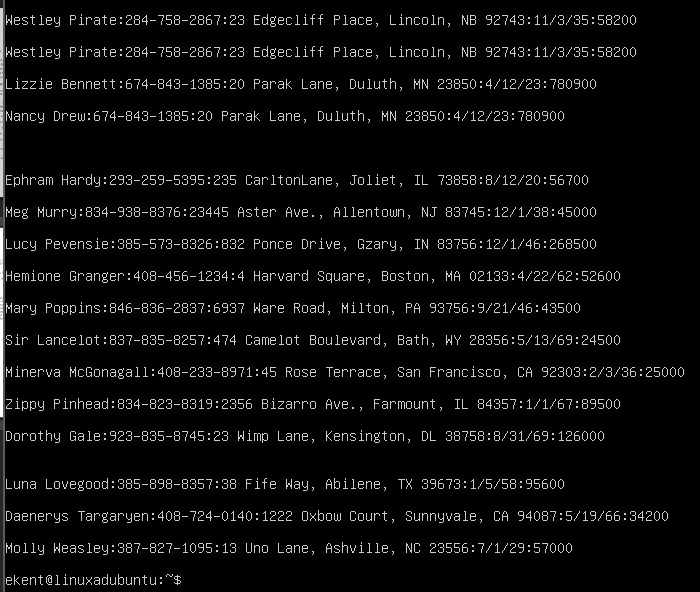
Description automatically generated

Prompt 9: Let’s find everyone on this list who live in Massachusetts or Illinois. To do so we are going to look for two things at the same time. To do so we will have to use quotation marks and the piping system along with the backslash. The command itself will be, grep “MA\|IL” filename. Text

Description automatically generated

We need the backslash in this because normally piping is used for another command with grep, the backslash tells it we are just looking for another pattern and not trying to use another command.

Prompt 10: For this we want to find all streets that aren’t specifically named “Street” or abbreviated. I used the “v” flag again since we want to exclude this from our results. Though some entries are “St” so we’re going to have to use the backslash \ and our piping |. So the command will be, grep -v “Street\|St” filename.



Sources used for lab:

<https://phoenixnap.com/kb/grep-multiple-strings>

<https://support.workiva.com/hc/en-us/articles/4407304269204-Regular-expression-operators>

<https://www.maketecheasier.com/what-is-grep-and-uses/>