Jason De Leon

03/02/23

Grep Lab Step Guide

On this Assignment we will learn about grep and what certain codes do. These are the first steps needed for this assignment.

What is Grep. Well Grep for a lack of better terms is essentially and **search and fine command**. Think of it as the Linux command version of **Control + F**. You give the command it parameters of what you want and it’ll display what you’re searching for.

**TRANSFERING FILE/S OVER**

Step 1.) Please download the following GrepLab File which is provided by the teacher.

Step 2.) **Please be aware that you will need to SCP this file over to the virtual machine**

I learned while doing this Lab you will need to be have the file downloaded on the path of which you command prompt is on. Please reference the pictures.

The Blue box is the path of where my Command prompt is. To make things easier I recommend you drag the file from where it download to, to where your start path is.

While the red Underline is the command, I used to send my file over to my Ubuntu Server.

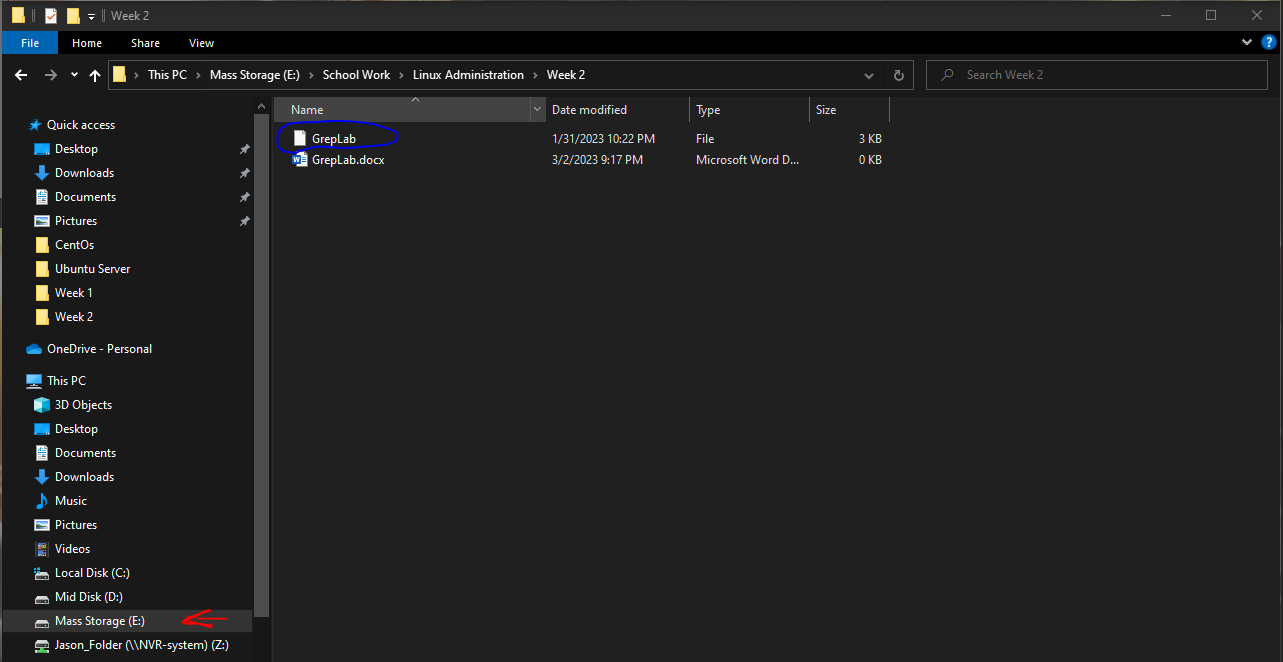
Text

Description automatically generated

The reason as to why I had an error on this was because I didn’t move the file over. Currently my system saves files over to my **E** drive aka my third drive.

There is where I had the file at the time. Notice that on the left the “Mass Storage (E:)” is selected:

The **RED ARROW** is the location, and the **BLUE CIRCLE** is the file.



The following Image is where I place the file.

Please notice that the file is now under the same name as the path for Command Prompt.

“C:\Users\19783” The **RED ARROW** is the drive, and the **BLUE CIRCLE** is the file name.

A screenshot of a computer

Description automatically generated with medium confidence

Once you’ve everything in their respected location we can follow through with the second step.

Here is when we send the file over to the Virtual server.

Due to me already attempting the file transfer It didn’t request me to type yes or no. If this is your first attempt to transfer the file **PLEASE TYPE YES** This will allow your system to trust the server. Without saying yes, your system will stop there and won’t let you send the file over.

My Ubuntu Server’s IP is 10.0.0.186

You know you’ve sent the file over successfully when you get this prompt. Please refer to the picture below:

A screenshot of a computer screen

Description automatically generated with medium confidence

Now over to the server you transferred the file to we can check if the file is there by typing *“****ls****”* on the command line.

This is what I get when I type that command:

Graphical user interface, text

Description automatically generated

**We can now officially start the Lab.**

**What’s in the File.**

Let us see what in the file. Please type the following “cat GrepLab”

What the command ***cat*** essentially does is print out the context of the file onto your screen for you to read.

I get the following when I hit enter:

Text

Description automatically generated

**PART 1.) Finding Lines that contain “Lane” in them:**

Now that we know what’s in the file. Let us commence with Part 1.

We are looking for everything that contains the following in their line: “Lane”. To find this we will use the **GREP** command.

**Grep** is essentially the command we use to find certain keywords in a file. Please type the following: “**grep (whatever you want to find) GrepLab”** in this case we will do this:

“**grep Lane GrepLab”**

If you did it properly, you should get this or something similar. Grep luckily highlights the words you’re searching for in **RED**, I’ve also **underlined** it you if you can’t find it.

Text

Description automatically generated

**PART 2.) Names that Start with the Letter “H”.**

Luckily this file has the name written first which makes the second part easy.

Just like the last part we will using Grep once more. Since we are looking for names that commence with the Letter “H” we will using the following command:

**“grep ^H GrapLab”**

If you want to find other letters just remove the “H” and type in anything you need.

Now you may be wondering what does “^” do?

Well that essentially means **“Whatever begins with”** it will look for anything that start with what you letter you and, in this case, we used the letter H. Please reference the picture blow to see what I got.

Notice that Grep prints the letter in **RED,** I’ve **CIRCLED** it **BLUE in case you can’t see it**

Text

Description automatically generated

**PART 3.) Lines that** END in “000”.

The following command addon “$” at the end of a parameter means **TO LOOK AT THE END**.

We will be using that addon to look for any line that ends the following number of “000”.

The command we will be using will be the following:

“grep 000$ GrepLab”

Please note that if you want to search for something else. Just remove the 000 and place what you want.

Please refer to the picture below to see what I get and you should end with something similar. Take a close look at the end of each line.

I’ve **CIRCLED** the **RED** numbers incase you’re unable to see the outcome.

Text

Description automatically generated

**PART 4.) Lines that DON’T contain “408”.**

The following command addon “-V” looks through the file and looks for lines that **DON’T** have the parameter you’ve given as an input.

This the command you will need to type:

**“grep – v “parameter” GrepLab”**

In this case “parameter” will be ***408***

**“grep – v 408 GrepLab”**

The command will spit out every line that **DOESN’T** have the number 408 in them. Please refer to my picture below to see what I get:

**NOTE:** None of the lines shown have the following in them, 408. Text

Description automatically generated

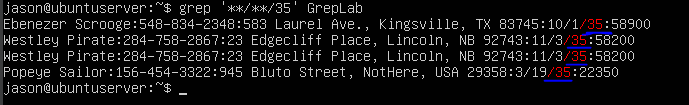
**Part 5.)**

As we want to find DOB that end in 35 we have to type the following line:

**“Grep ‘\*\*/\*\*/35’ GrepLab”**

The reason why we have the ‘ is meant to tell the command that **EVERYTHING** in the ‘’ is one line. The the reason why we have the “\*” is a way to say these are placeholders and have no true meaning.

In the picture below I’ve underlined the year so you can see what you would get when you type the command.

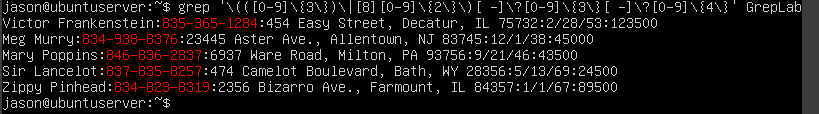


**Part 6.)**

**Source** [**https://www.makeuseof.com/grep-command-practical-examples/**](https://www.makeuseof.com/grep-command-practical-examples/)

In order to print out those who have a cell phone number that start with 8 we had to type the following:

**(grep '\(([0-9]\{3\})\|[8][0-9]\{2\}\)[ -]\?[0-9]\{3\}[ -]\?[0-9]\{4\}' GrebLab)**



**Part 7.)**

Here we will be printing out the lines that start with a Uppercase letter followed by 4 lower case, then a space and finally another letter.

**Command is:( *grep -E ‘[A-Z]{1}[a-z]{4} [A-Z]{1}’ GrepLab )***

A quick explanation as to what this does. **-E** basically means that THIS command must have the following as a pattern.

[A-Z]{1} means that there must be a capital letter one time.

[a-z]{4} A lowercase letter fours times after the first letter

“ “ there must be a space

[A-Z]{1} another upper case after the space

And finally, all inside single quotes. That tells the program “Hey this is one command. Follow it the way it’s written.”

Text

Description automatically generated

**Part 8.)**

On this step we will be searching for streets start either contain two or three numbers to them.

The command is the following:

***(grep -E ‘[1-9]{2} [A-Z]|[0-9]{3} [A-Z] {1}’ GrepLab)***

[1-9]{2} means you want two characters that are numbered then a letter, three more numbers and finally one more letter.

Text

Description automatically generated

**Step 9.)**

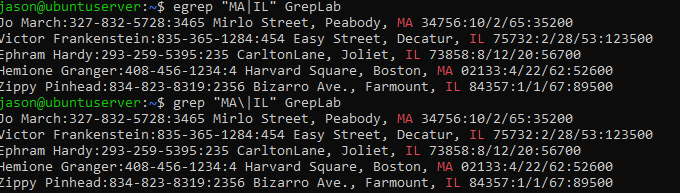
In this step we will be searching for people who live in Massachusetts **OR** Illinois

This will be the following command:

(***grep ‘*MA\|IL’ *GrepLab)***

The \| means OR

Another method is (egrep ‘MA|IL’) Unlike \| the command treats MA|IL as Search for MA and IL, but we want OR so \| is the OR command for Grep



**Step 10.)**

Finally we will be searching for NON streets. This will just be the following command

**grep -v ‘Street|St’ GrepLab**

-v means to exclude the following

Notices how none of these lines show street or st in their address.

Text

Description automatically generated