Lab 5 – Filtering

Name: Paul Christiansen
Course/Section: IS-1003-001
Date: 4/1/2025

Introduction

Cybersecurity professionals go through a significant amount of data on a day-to-day basis. Whether that is network traffic, code analysis, incident reports, and even just documentation. Some way to simplify the workflow is to develop a script that can sort the given information by a specific identifier. That can be name, e-mail address, date hired or any other identifier. In this lab we explore what the code to do that looks like in the Linux terminal via Bash.

Breakpoint 1

I forgot to get a picture of the files in the actual file explorer but the code after this breakpoint will display that I did download the right file. With that said, how I download the zip file was really easy. I almost downloaded it on my main device but then saw the instructions. After reading that, I immediately spun up my VM and went to canvas via Firefox and downloaded the file to my downloads folder. Afterwards, I extracted it out to my documents folder where I will continue with the lab.

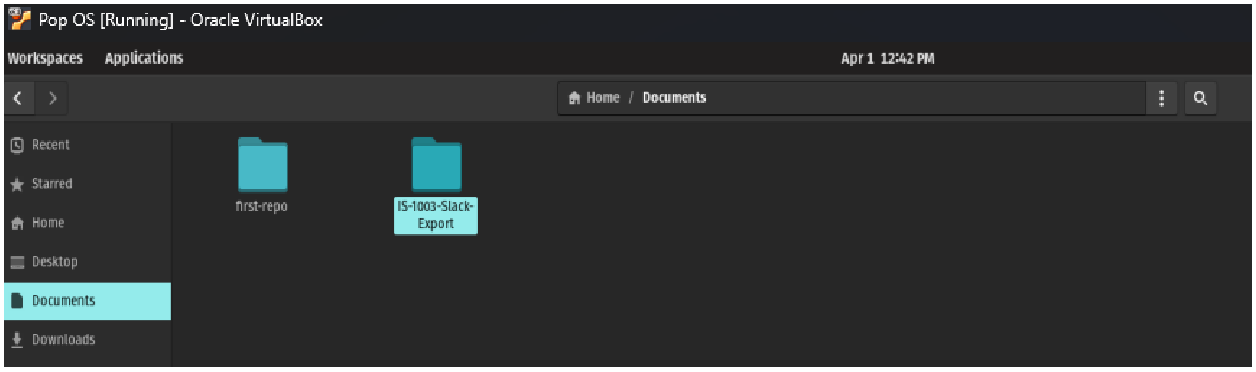
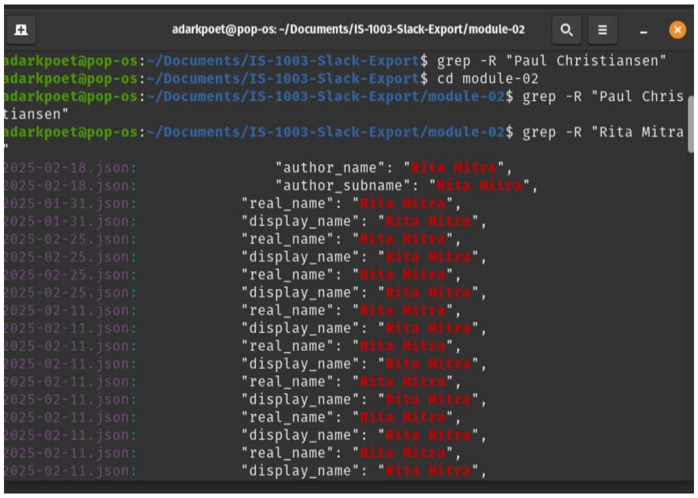


Figure 1 Exported Zip File

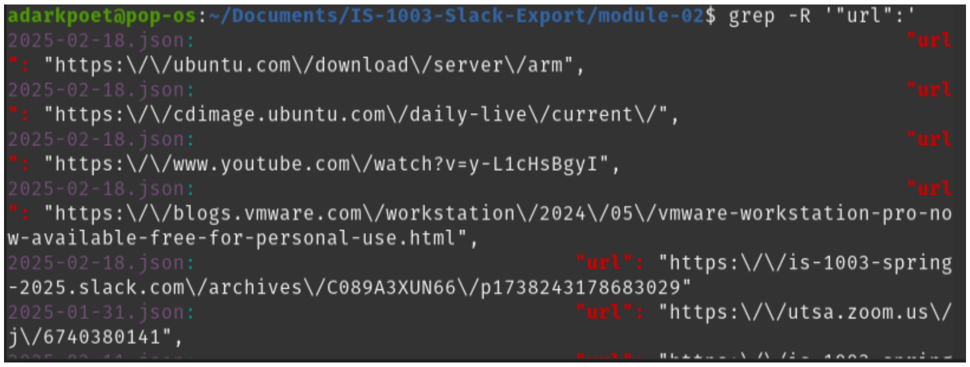
Breakpoint 2

This breakpoint goes through the grep command. What the grep command does is to look through the specific directory for x string and then print out each instance of it. In figure 2, I learned I couldn't use the grep command on my own name as I haven't posted enough in slack for me to find myself. With that said, per the instructions I used the Rita Mitra as my grep target instead. The results of that is show in figure 2. In figure 3, we are changing the specifications for grabbing a specific name to grabbing any string that is a URL. What this can be used for is to find what URLs are listed in a specific workspace. In figure 4, instead of grabbing URLs we atr now grabbing anything that is considered text/string. What that does for this lab is grab all the comments and posts made on slack and displays them. We can add on to this line by specifying that we want to put the text into a text file. Shown in figure 5, what this does is make all the information we pulled more readable, and we can now start to manipulate that file instead of manipulating other files.


adarkpoet@pop-os: ~/Documents/IS-1003-Slack-Export/module-02

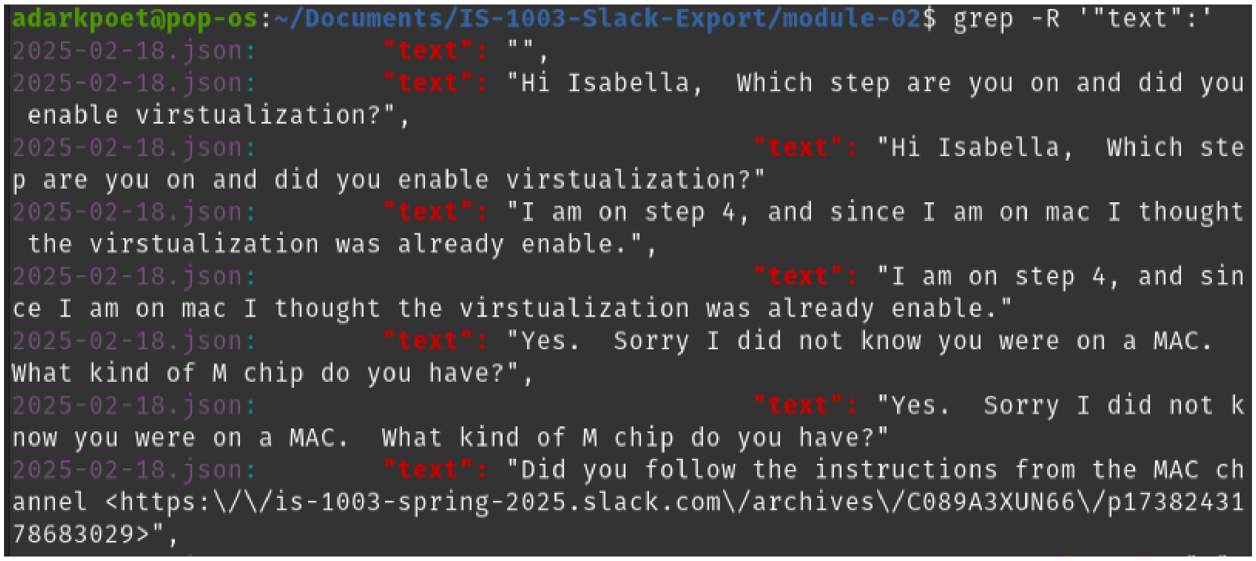
```
adarkpoet@pop-os:~/Documents/IS-1003-Slack-Export$ grep -R "Paul Christiansen"
adarkpoet@pop-os:~/Documents/IS-1003-Slack-Export$ cd module-02
adarkpoet@pop-os:~/Documents/IS-1003-Slack-Export/module-02$ grep -R "Paul Chris
tiansen"
adarkpoet@pop-os:~/Documents/IS-1003-Slack-Export/module-02$ grep -R "Rita Mitra
"
025-02-18.json:                     "author_name": "Rita Mitra",
025-02-18.json:                     "author_subname": "Rita Mitra",
025-01-31.json:                 "real_name": "Rita Mitra",
025-01-31.json:                 "display_name": "Rita Mitra",
025-02-25.json:                 "real_name": "Rita Mitra",
025-02-25.json:                 "display_name": "Rita Mitra",
025-02-25.json:                 "real_name": "Rita Mitra",
025-02-25.json:                 "display_name": "Rita Mitra",
025-02-11.json:                 "real_name": "Rita Mitra",
025-02-11.json:                 "display_name": "Rita Mitra",
025-02-11.json:                 "real_name": "Rita Mitra",
025-02-11.json:                 "display_name": "Rita Mitra",
025-02-11.json:                 "real_name": "Rita Mitra",
025-02-11.json:                 "display_name": "Rita Mitra",
025-02-11.json:                 "real_name": "Rita Mitra",
025-02-11.json:                 "display_name": "Rita Mitra",
025-02-11.json:                 "real_name": "Rita Mitra",
025-02-11.json:                 "display_name": "Rita Mitra",
```

Figure 3 Grep testing



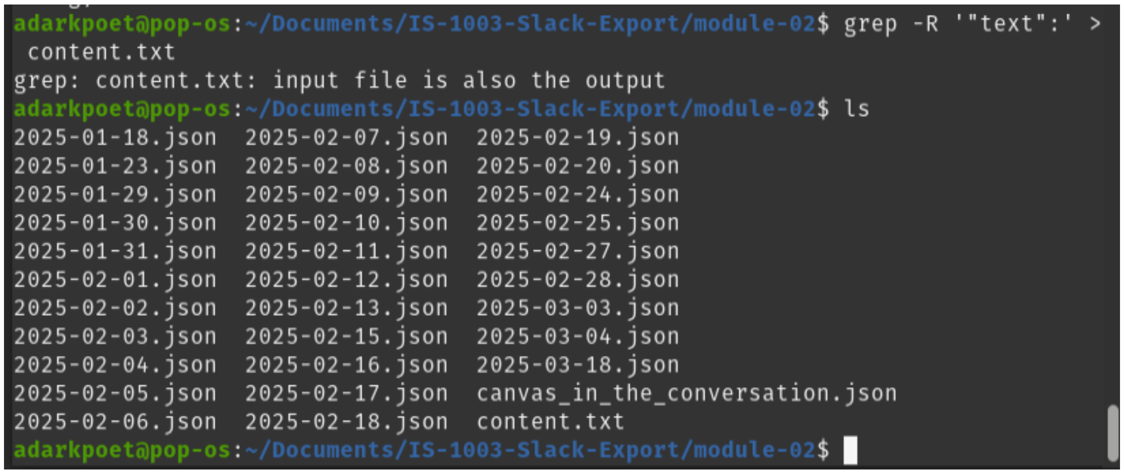
```
adarkpoet@pop-os:~/Documents/IS-1003-Slack-Export/module-02$ grep -R '"url":'
2025-02-18.json:                                                         "url
": "https:\/\/ubuntu.com\/download\/server\/arm",
2025-02-18.json:                                                         "url
": "https:\/\/cdimage.ubuntu.com\/daily-live\/current\/",
2025-02-18.json:                                                         "url
": "https:\/\/www.youtube.com\/watch?v=y-L1cHsBgyI",
2025-02-18.json:                                                         "url
": "https:\/\/blogs.vmware.com\/workstation\/2024\/05\/vmware-workstation-pro-no
w-available-free-for-personal-use.html",
2025-02-18.json:                                "url": "https:\/\/is-1003-spring
-2025.slack.com\/archives\/C089A3XUN66\/p1738243178683029"
2025-01-31.json:                                "url": "https:\/\/utsa.zoom.us\/
j\/6740380141",
```

Figure 2 URL grabbing



```
adarkpoet@pop-os:~/Documents/IS-1003-Slack-Export/module-02$ grep -R '"text":'
2025-02-18.json:          "text": "",
2025-02-18.json:          "text": "Hi Isabella,  Which step are you on and did you
 enable virstualization?",
2025-02-18.json:                                      "text": "Hi Isabella,  Which ste
p are you on and did you enable virstualization?"
2025-02-18.json:          "text": "I am on step 4, and since I am on mac I thought
 the virstualization was already enable.",
2025-02-18.json:                                      "text": "I am on step 4, and sin
ce I am on mac I thought the virstualization was already enable."
2025-02-18.json:          "text": "Yes.  Sorry I did not know you were on a MAC.
What kind of M chip do you have?",
2025-02-18.json:                                      "text": "Yes.  Sorry I did not k
now you were on a MAC.  What kind of M chip do you have?"
2025-02-18.json:          "text": "Did you follow the instructions from the MAC ch
annel <https:\/\/is-1003-spring-2025.slack.com\/archives\/C089A3XUN66\/p17382431
78683029>",
```

Figure 5 Text grabbing

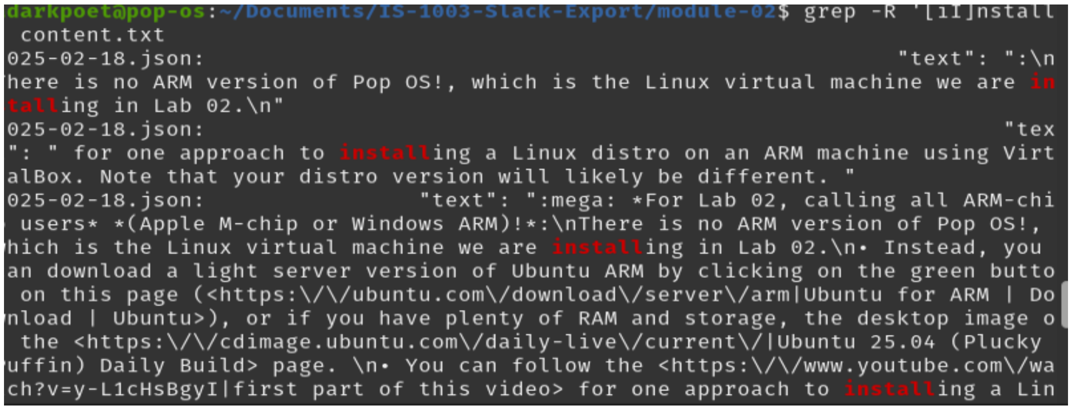


```
adarkpoet@pop-os:~/Documents/IS-1003-Slack-Export/module-02$ grep -R '"text":' >
 content.txt
grep: content.txt: input file is also the output
adarkpoet@pop-os:~/Documents/IS-1003-Slack-Export/module-02$ ls
2025-01-18.json  2025-02-07.json  2025-02-19.json
2025-01-23.json  2025-02-08.json  2025-02-20.json
2025-01-29.json  2025-02-09.json  2025-02-24.json
2025-01-30.json  2025-02-10.json  2025-02-25.json
2025-01-31.json  2025-02-11.json  2025-02-27.json
2025-02-01.json  2025-02-12.json  2025-02-28.json
2025-02-02.json  2025-02-13.json  2025-03-03.json
2025-02-03.json  2025-02-15.json  2025-03-04.json
2025-02-04.json  2025-02-16.json  2025-03-18.json
2025-02-05.json  2025-02-17.json  canvas_in_the_conversation.json
2025-02-06.json  2025-02-18.json  content.txt
adarkpoet@pop-os:~/Documents/IS-1003-Slack-Export/module-02$ 
```

Figure 4 showing text file

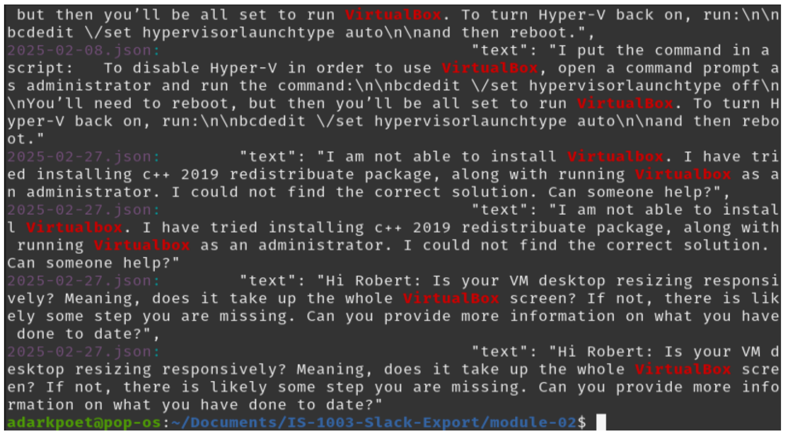
Breakpoint 3

For this break point we are now looking for words inside the entire directory. In figure 6 we use the grep command to find the word “install” with both the I and i. We specify this by doing [iI] which looks for both variations. In figure 8, we do the same thing but with a minor adjustment that makes a big difference is that we don’t care about the casing in either the V or the B in VirtualBox which is why we put the i in the -iR section and then \? In the second half of the word. The last one, figure 8, is looking for all variations of verify but ONLY those with the iy ending and not the ed ending which is why the [id] is grouped like that and the asterisks are next to the ed.



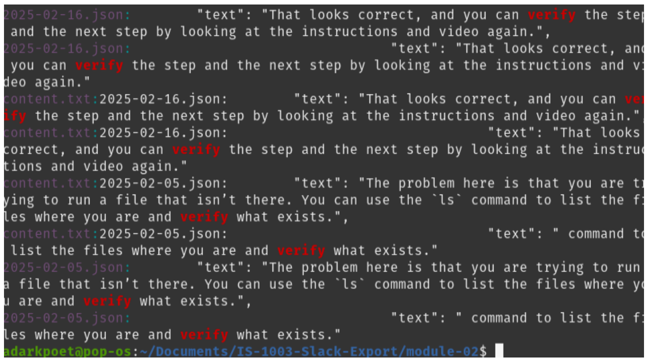
```
darkpoet@pop-os:~/Documents/IS-1003-Slack-Export/module-02$ grep -R '[iI]nstall
 content.txt
025-02-18.json:                                                   "text": ":\n
here is no ARM version of Pop OS!, which is the Linux virtual machine we are in
talling in Lab 02.\n"
025-02-18.json:                                                          "tex
": " for one approach to installing a Linux distro on an ARM machine using Virt
alBox. Note that your distro version will likely be different. "
025-02-18.json:                    "text": ":mega: *For Lab 02, calling all ARM-chi
 users* *(Apple M-chip or Windows ARM)!*:\nThere is no ARM version of Pop OS!,
hich is the Linux virtual machine we are installing in Lab 02.\n• Instead, you
an download a light server version of Ubuntu ARM by clicking on the green butto
 on this page (<https:\/\/ubuntu.com\/download\/server\/arm|Ubuntu for ARM | Do
nload | Ubuntu>), or if you have plenty of RAM and storage, the desktop image o
 the <https:\/\/cdimage.ubuntu.com\/daily-live\/current\/|Ubuntu 25.04 (Plucky
uffin) Daily Build> page. \n• You can follow the <https:\/\/www.youtube.com\/wa
ch?v=y-L1cHsBgyI|first part of this video> for one approach to installing a Lin
```

Figure 6 Grepping Install



```
 but then you’ll be all set to run VirtualBox. To turn Hyper-V back on, run:\n\n
bcdedit \/set hypervisorlaunchtype auto\n\nand then reboot.",
2025-02-08.json:                                   "text": "I put the command in a 
script:   To disable Hyper-V in order to use VirtualBox, open a command prompt a
s administrator and run the command:\n\nbcdedit \/set hypervisorlaunchtype off\n
\nYou’ll need to reboot, but then you’ll be all set to run VirtualBox. To turn H
yper-V back on, run:\n\nbcdedit \/set hypervisorlaunchtype auto\n\nand then rebo
ot."
2025-02-27.json:          "text": "I am not able to install Virtualbox. I have tri
ed installing c++ 2019 redistribuate package, along with running Virtualbox as a
n administrator. I could not find the correct solution. Can someone help?",
2025-02-27.json:                                   "text": "I am not able to instal
l Virtualbox. I have tried installing c++ 2019 redistribuate package, along with
 running Virtualbox as an administrator. I could not find the correct solution. 
Can someone help?"
2025-02-27.json:          "text": "Hi Robert: Is your VM desktop resizing responsi
vely? Meaning, does it take up the whole VirtualBox screen? If not, there is lik
ely some step you are missing. Can you provide more information on what you have
 done to date?",
2025-02-27.json:                                   "text": "Hi Robert: Is your VM d
esktop resizing responsively? Meaning, does it take up the whole VirtualBox scre
en? If not, there is likely some step you are missing. Can you provide more info
rmation on what you have done to date?"
adarkpoet@pop-os:~/Documents/IS-1003-Slack-Export/module-02$ 
```

Figure 7 Grepping for VirtualBox

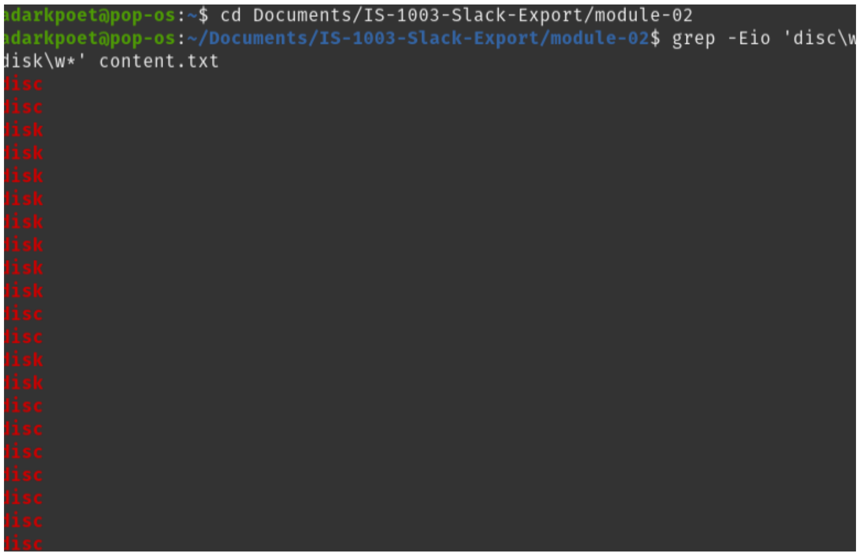


```
2025-02-16.json:         "text": "That looks correct, and you can verify the step
 and the next step by looking at the instructions and video again.",
2025-02-16.json:                                  "text": "That looks correct, and
 you can verify the step and the next step by looking at the instructions and vi
deo again."
content.txt:2025-02-16.json:        "text": "That looks correct, and you can ver
ify the step and the next step by looking at the instructions and video again.",
content.txt:2025-02-16.json:                                 "text": "That looks 
correct, and you can verify the step and the next step by looking at the instruc
tions and video again."
content.txt:2025-02-05.json:        "text": "The problem here is that you are tr
ying to run a file that isn’t there. You can use the `ls` command to list the fi
les where you are and verify what exists.",
content.txt:2025-02-05.json:                                 "text": " command to
 list the files where you are and verify what exists."
2025-02-05.json:         "text": "The problem here is that you are trying to run 
a file that isn’t there. You can use the `ls` command to list the files where yo
u are and verify what exists.",
2025-02-05.json:                                  "text": " command to list the fi
les where you are and verify what exists."
adarkpoet@pop-os:~/Documents/IS-1003-Slack-Export/module-02$ 
```

Figure 8 Verify Grep

BREAKPOINT 4

What I wanted to see was how many times people said disc or the other variation, disk. The catch, since this is not my first time with Bash, I just wanted to see the words themselves and not anything else. This changes my -r command to -EIO where it prints out the words themselves and nothing else. I made sure to include that it prints disc with (\w) disk. And the figure below shows the execution of that command.

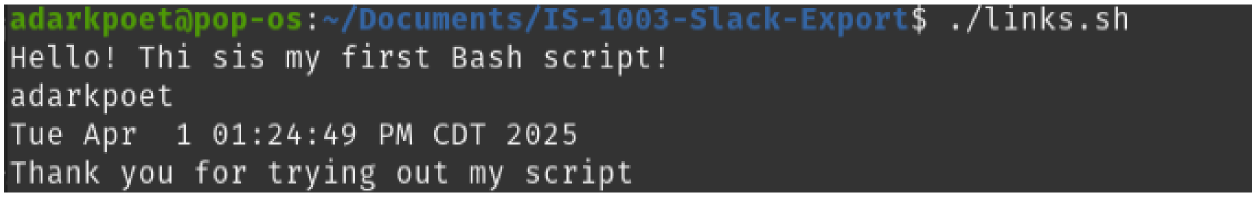


```
adarkpoet@pop-os:~$ cd Documents/IS-1003-Slack-Export/module-02
adarkpoet@pop-os:~/Documents/IS-1003-Slack-Export/module-02$ grep -Eio 'disc\w
disk\w*' content.txt
disc
disc
disk
disk
disk
disk
disk
disk
disk
disk
disc
disc
disk
disk
disc
disc
disc
disc
disc
disc
disc
```

Figure 7 Disc/Disk Execution

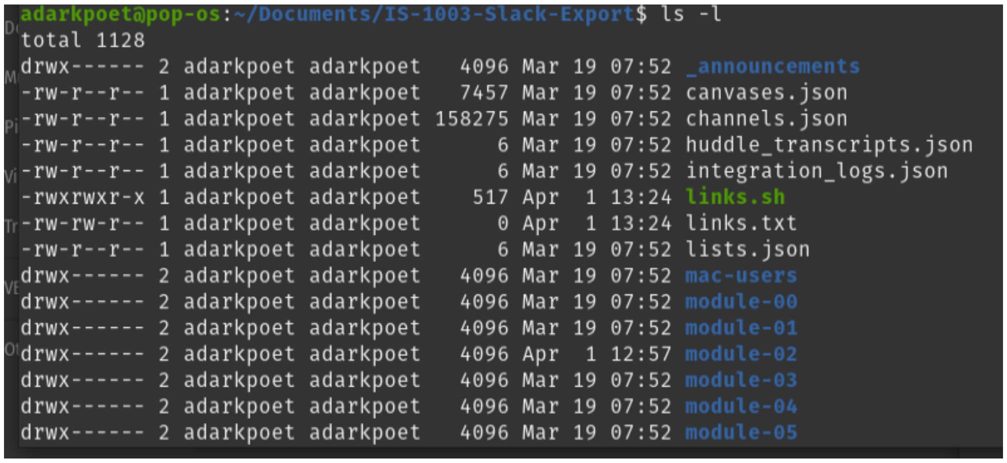
BREAKPOINT 5

This breakpoint was extremely easy. The whole point was to create a script that went through the directory and picked out the URLs and then organized then alphabetically. This involved creating and executable in the terminal via using the nano command. We also used the cdmod command to make the text file turn into an executable. In the figure below, you can see how the links.sh file is yellow, indicating it an executable file that you can put code in to execute. What you will notice is that while the code executed was in the main directory the urls themselves were in the module-02 directory. That was due to the fact that I was having problems getting it to run in the main directory, so I moved the bash file to module-02 and retyped all the code and was able to get it to work. My guess was that I wrote the script code wrong and resulted in the files not being retrieved and placed into the links.txt file (that was my problem).



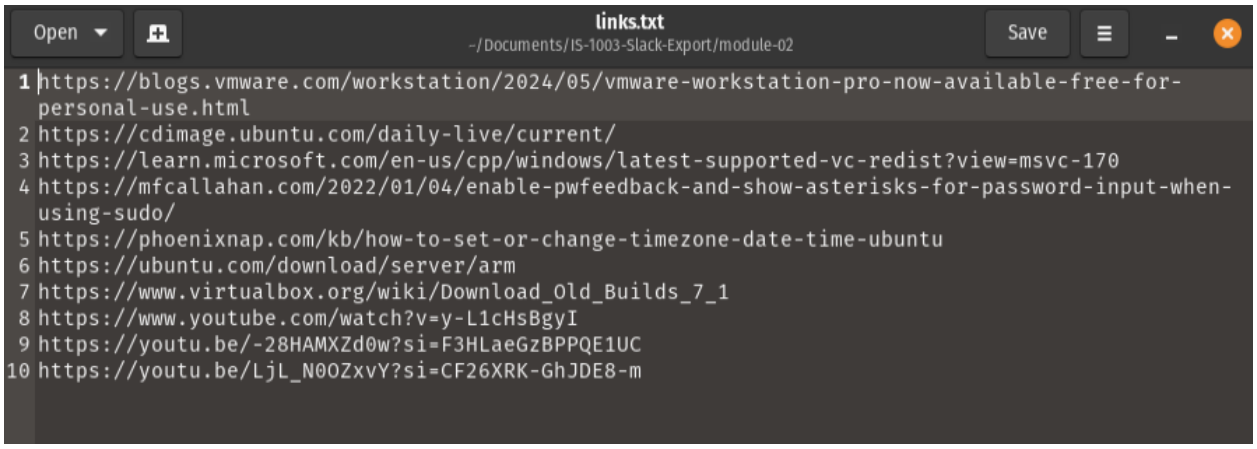
```
adarkpoet@pop-os:~/Documents/IS-1003-Slack-Export$ ./links.sh
Hello! Thi sis my first Bash script!
adarkpoet
Tue Apr  1 01:24:49 PM CDT 2025
Thank you for trying out my script
```

Figure 8 Bash code execution



```
adarkpoet@pop-os:~/Documents/IS-1003-Slack-Export$ ls -l
total 1128
drwx------ 2 adarkpoet adarkpoet   4096 Mar 19 07:52 _announcements
-rw-r--r-- 1 adarkpoet adarkpoet   7457 Mar 19 07:52 canvases.json
-rw-r--r-- 1 adarkpoet adarkpoet 158275 Mar 19 07:52 channels.json
-rw-r--r-- 1 adarkpoet adarkpoet      6 Mar 19 07:52 huddle_transcripts.json
-rw-r--r-- 1 adarkpoet adarkpoet      6 Mar 19 07:52 integration_logs.json
-rwxrwxr-x 1 adarkpoet adarkpoet    517 Apr  1 13:24 links.sh
-rw-rw-r-- 1 adarkpoet adarkpoet      0 Apr  1 13:24 links.txt
-rw-r--r-- 1 adarkpoet adarkpoet      6 Mar 19 07:52 lists.json
drwx------ 2 adarkpoet adarkpoet   4096 Mar 19 07:52 mac-users
drwx------ 2 adarkpoet adarkpoet   4096 Mar 19 07:52 module-00
drwx------ 2 adarkpoet adarkpoet   4096 Mar 19 07:52 module-01
drwx------ 2 adarkpoet adarkpoet   4096 Apr  1 12:57 module-02
drwx------ 2 adarkpoet adarkpoet   4096 Mar 19 07:52 module-03
drwx------ 2 adarkpoet adarkpoet   4096 Mar 19 07:52 module-04
drwx------ 2 adarkpoet adarkpoet   4096 Mar 19 07:52 module-05
```

Figure 10 -ls l command execution



```
https://blogs.vmware.com/workstation/2024/05/vmware-workstation-pro-now-available-free-for-personal-use.html
https://cdimage.ubuntu.com/daily-live/current/
https://learn.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170
https://mfcallahan.com/2022/01/04/enable-pwfeedback-and-show-asterisks-for-password-input-when-using-sudo/
https://phoenixnap.com/kb/how-to-set-or-change-timezone-date-time-ubuntu
https://ubuntu.com/download/server/arm
https://www.virtualbox.org/wiki/Download_Old_Builds_7_1
https://www.youtube.com/watch?v=y-L1cHsBgyI
https://youtu.be/-28HAMXZd0w?si=F3HLaeGzBPPQE1UC
https://youtu.be/LjL_N0OZxvY?si=CF26XRK-GhJDE8-m
```

Figure 9 Links.txt file

Conclusion

Closing out this lab, overall, it was very easy and a good brush up on my bash coding! I did do some messing around after the lab with the cp command to see if I could create multiple executables and just spam them but that didn't work, the computer said it wasn't able to see the files upon trying to execute them. The hardest part about this lab was rewriting the code after I was struggling getting it to run in the main directory but that was fix by changing the directory and rewriting the code. Overall, this was a fun and enjoyable lab!

References

R. Mitra, "*Lab 5: Filtering*," The University of Texas at San Antonio (2024). Last accessed: April 1, 2025
.

Collaboration

I was by myself.