



CONESTOGA

Connect Life and Learning

Student Name:	Aagam Sanjay Shah
Deliverable:	In-Class Tasks Week 12
Course Name:	NTWK8141-24S-Sec3-Linux Server

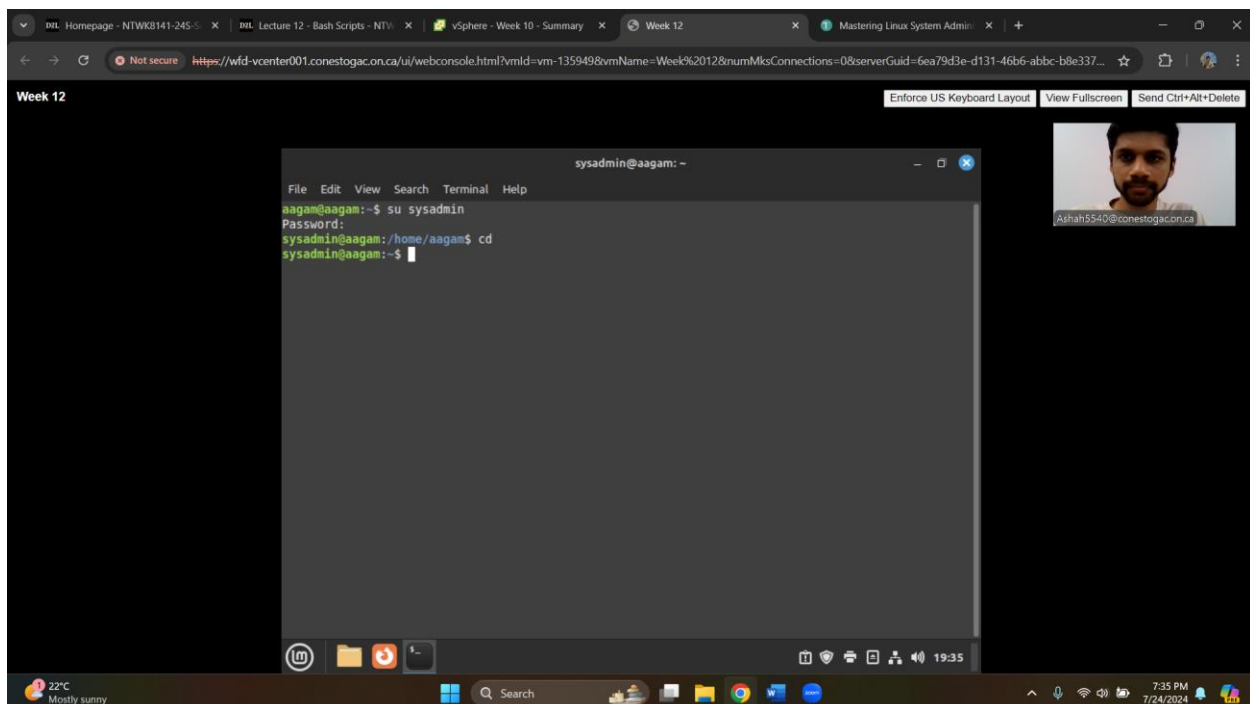
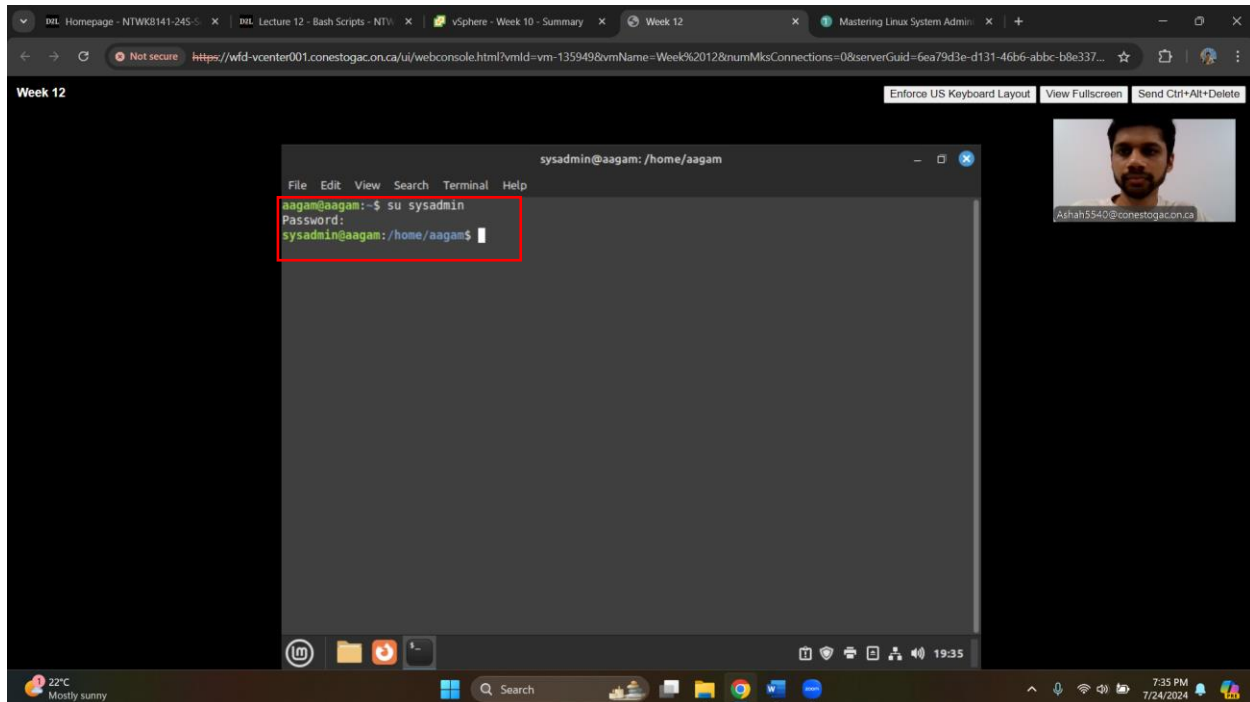
Date Assigned:	24/07/2024
Date Due:	25/07/2024
Rules:	<ul style="list-style-type: none">• Individual.• Cheating is not allowed.• Plagiarism counts as cheating!• That FAILURE to submit work in the course can result in a grade of 'F' or 'I' for failure to complete the course!

In Class Task: scripts

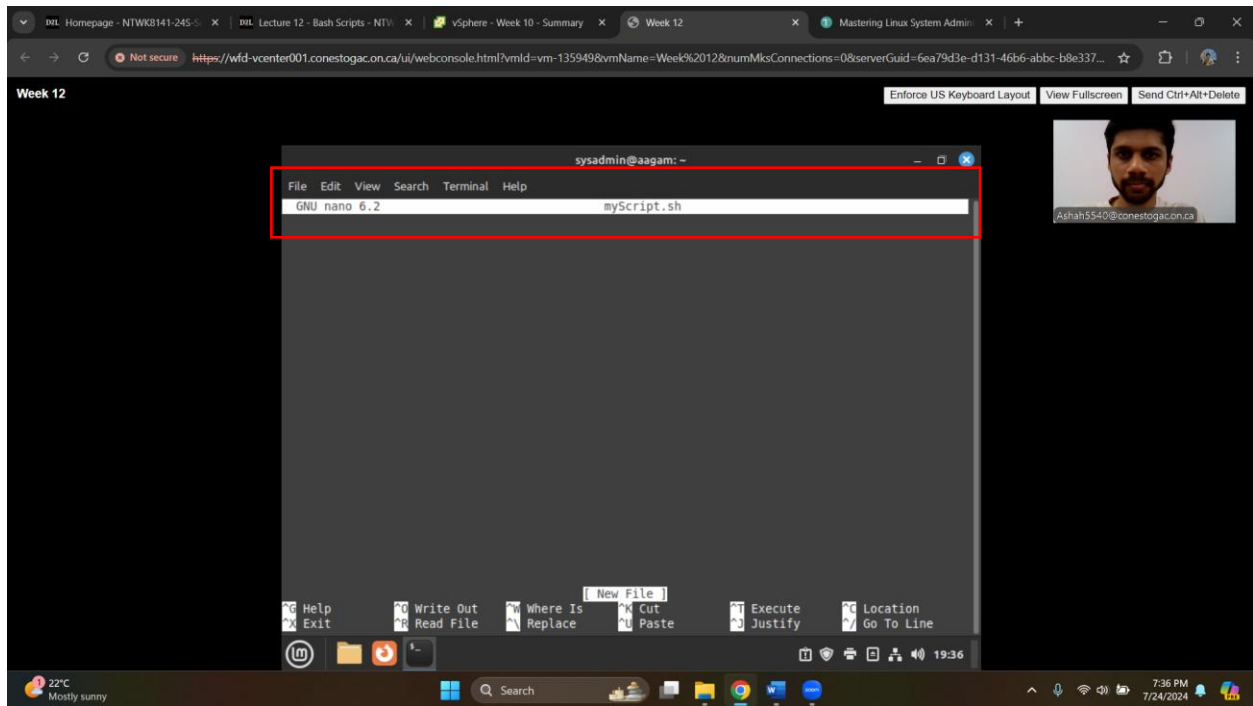
Complete the Real World Scenario: Writing a Simple Shell Script with Variables in Ch 19

WRITING A SIMPLE SHELL SCRIPT WITH VARIABLES

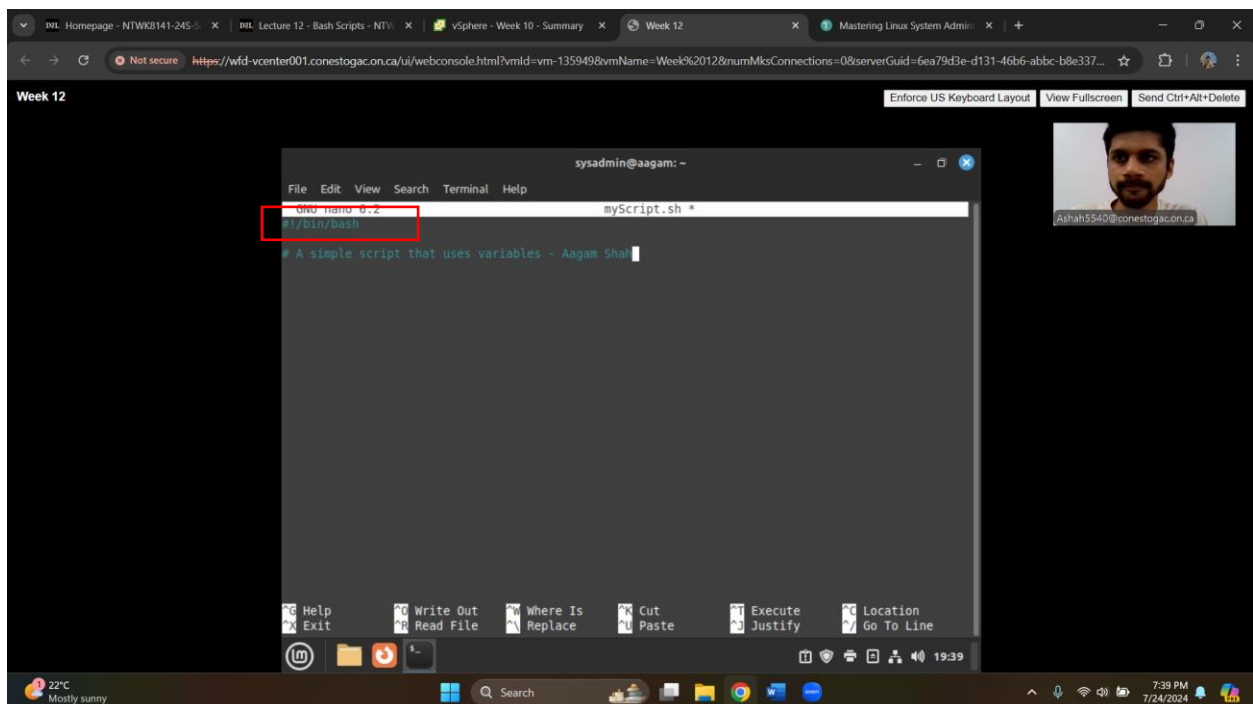
1. Log into your Linux system, using the sysadmin account and the password you created for it.



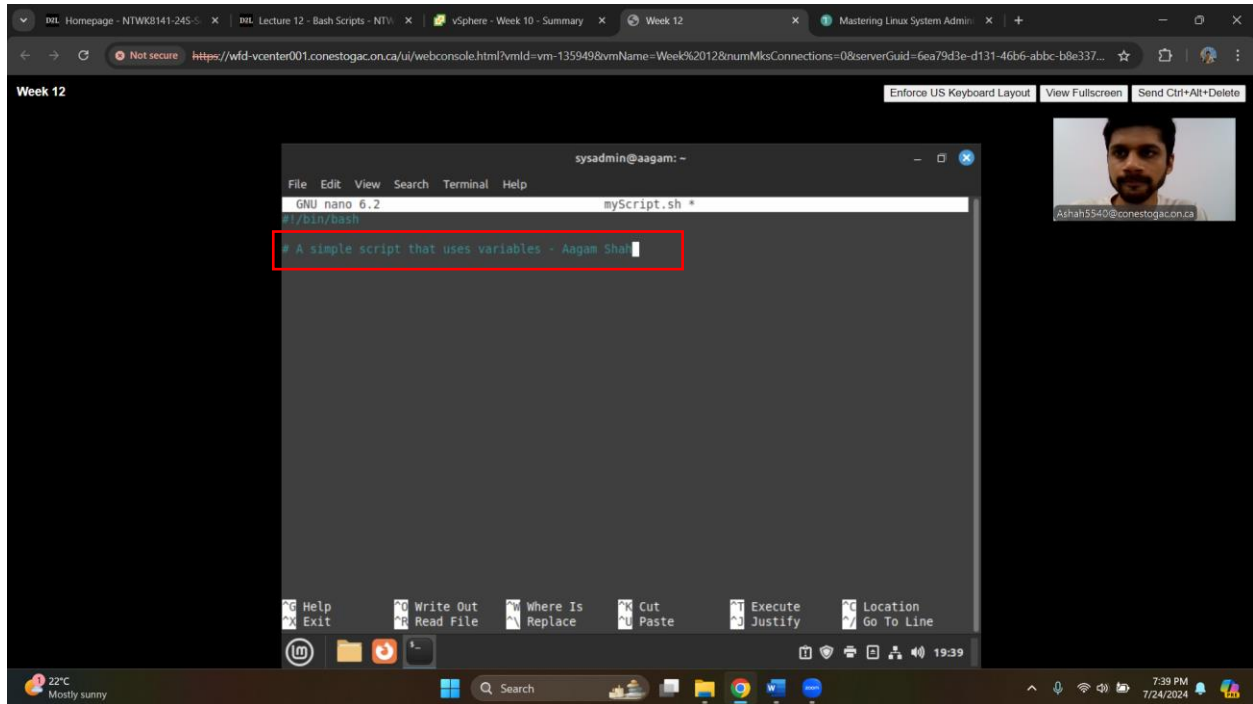
2. Start creating a shell script by typing `nano myScript.sh` and pressing Enter. This will create the file `myScript.sh` and put you into the nano text editor to start editing it.



3. On the file's first line, type `#!/bin/bash` and press Enter. This will select the Bash shell to run the script when it is executed.



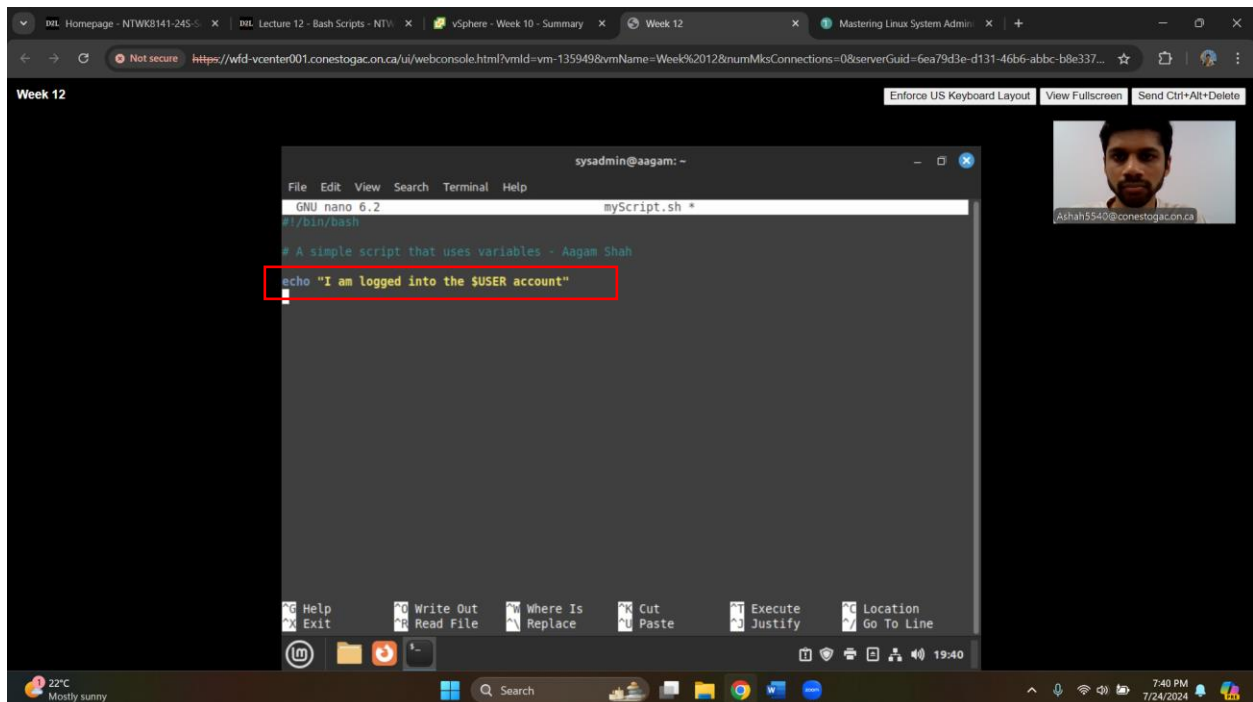
- On the file's second line, type `# A simple script that uses variables` and press Enter. This is a comment line. You can add your name to the end of the comment line if you want.



The screenshot shows a web browser window with a URL that includes 'vm-1359498vmName=Week%2012&numMksConnections=0&serverGuid=6ea79d3e-d131-46b6-abb6-b8e337...'. The browser displays a 'Week 12' session. A terminal window titled 'sysadmin@aagam: ~' is open, showing the GNU nano 6.2 editor editing 'myScript.sh'. The first line is '#!/bin/bash'. The second line, which is highlighted with a red box, is '# A simple script that uses variables - Aagam Shah'. The bottom of the terminal shows various keyboard shortcuts like 'Help', 'Exit', 'Write Out', 'Read File', 'Where Is', 'Replace', 'Cut', 'Paste', 'Execute', 'Justify', and 'Location', 'Go To Line'. The system tray at the bottom indicates '22°C Mostly sunny' and the time '7:39 PM 7/24/2024'.

```
sysadmin@aagam: ~
File Edit View Search Terminal Help
GNU nano 6.2 myScript.sh *
#!/bin/bash
# A simple script that uses variables - Aagam Shah
```

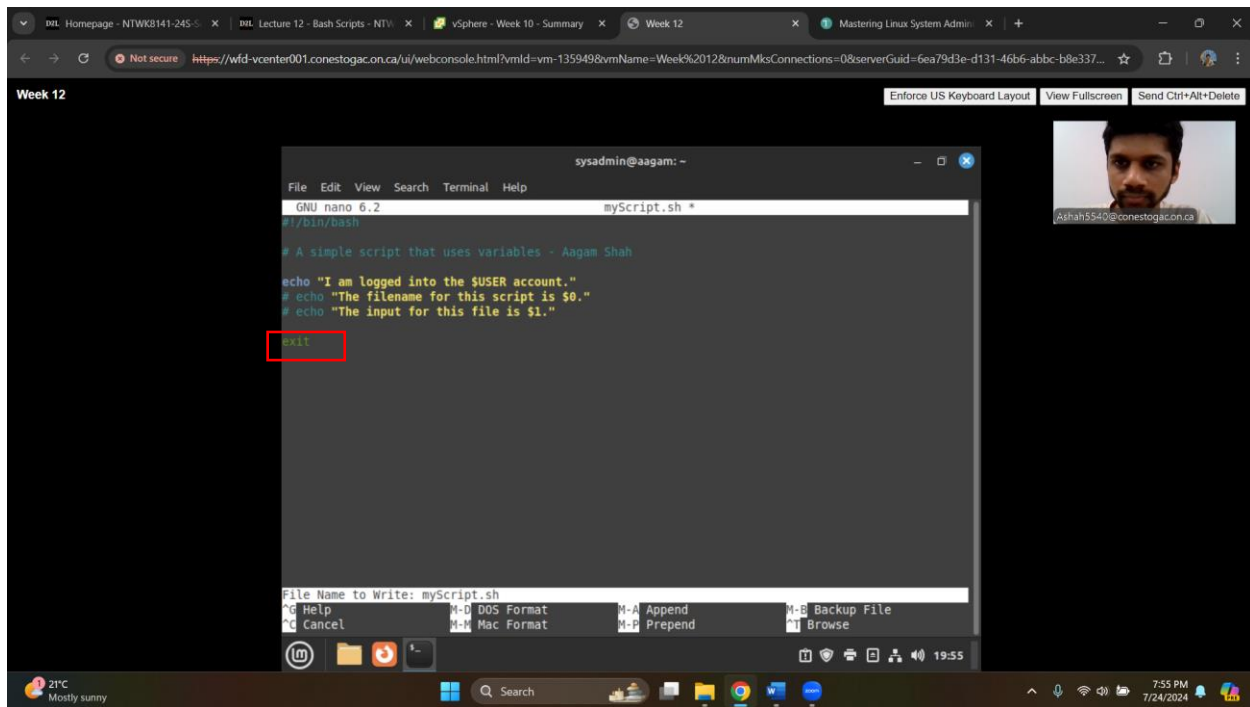
- On the third line of the file, type `echo "I am logged into the $USER account."` and press Enter. This will display the text along with the username of the current account you are using.



This screenshot is similar to the previous one, showing the same web console and terminal window. The nano editor now has three lines: '#!/bin/bash', '# A simple script that uses variables - Aagam Shah', and a third line, 'echo "I am logged into the \$USER account"', which is highlighted with a red box. The rest of the interface, including the browser tabs, URL, and system tray, remains the same.

```
sysadmin@aagam: ~
File Edit View Search Terminal Help
GNU nano 6.2 myScript.sh *
#!/bin/bash
# A simple script that uses variables - Aagam Shah
echo "I am logged into the $USER account"
```

6. On the file's last line, type exit.



The screenshot shows a web browser window with a URL starting with 'https://wfd-vcenter001.conestogac.on.ca/ui/webconsole.html'. The browser tabs include 'Week 12' and 'Mastering Linux System Admin'. The main content area displays a terminal window titled 'sysadmin@aagam: -'. Inside the terminal, a nano editor is open for a file named 'myScript.sh'. The editor's menu bar shows 'File Edit View Search Terminal Help'. The file content is as follows:

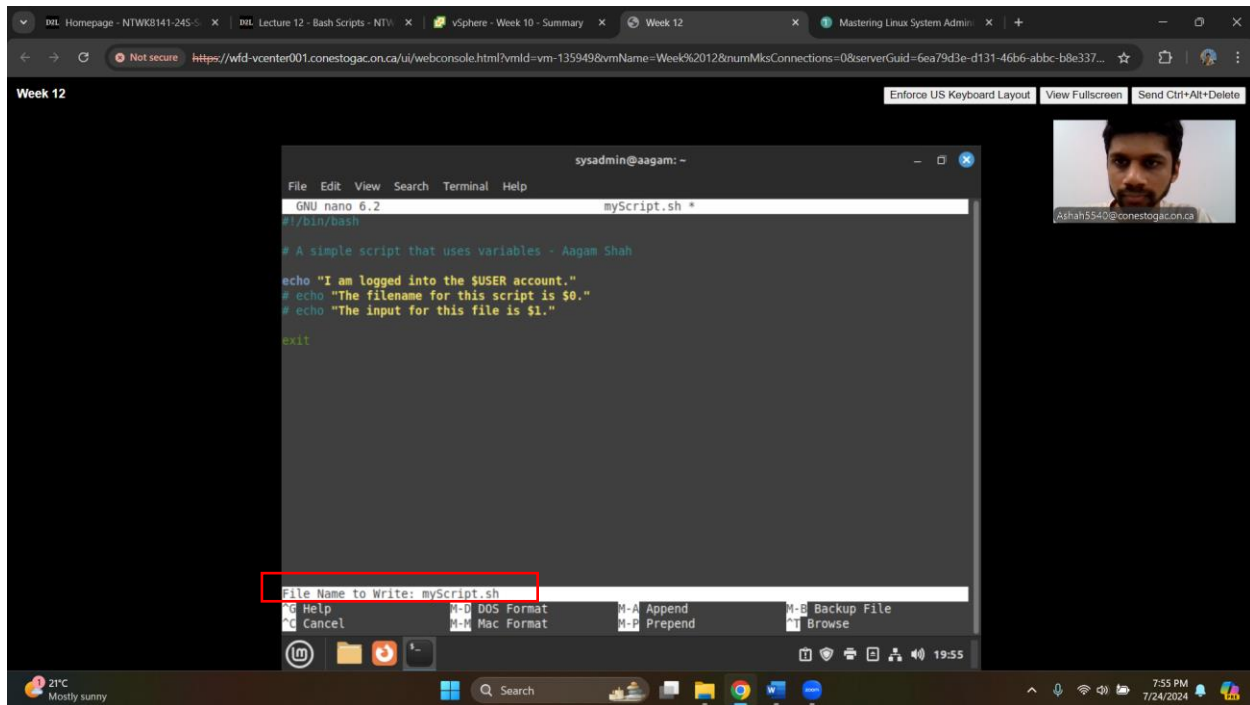
```
GNU nano 6.2 myScript.sh *
#!/bin/bash

# A simple script that uses variables - Aagam Shah

echo "I am logged into the $USER account."
# echo "The filename for this script is $0."
# echo "The input for this file is $1."
exit
```

The word 'exit' on the last line is highlighted with a red box. Below the editor, the status bar shows 'File Name to Write: myScript.sh' and various icons. The bottom of the browser window shows a Windows taskbar with a search bar and system tray icons.

7. Save the entered text to the script file by pressing Ctrl+O and pressing Enter when the file's name displays on the text editor's status line.



This screenshot is similar to the previous one, but the 'exit' command has been removed from the script file. The status bar at the bottom of the nano editor now displays 'File Name to Write: myScript.sh', which is highlighted with a red box. The file content is:

```
GNU nano 6.2 myScript.sh *
#!/bin/bash

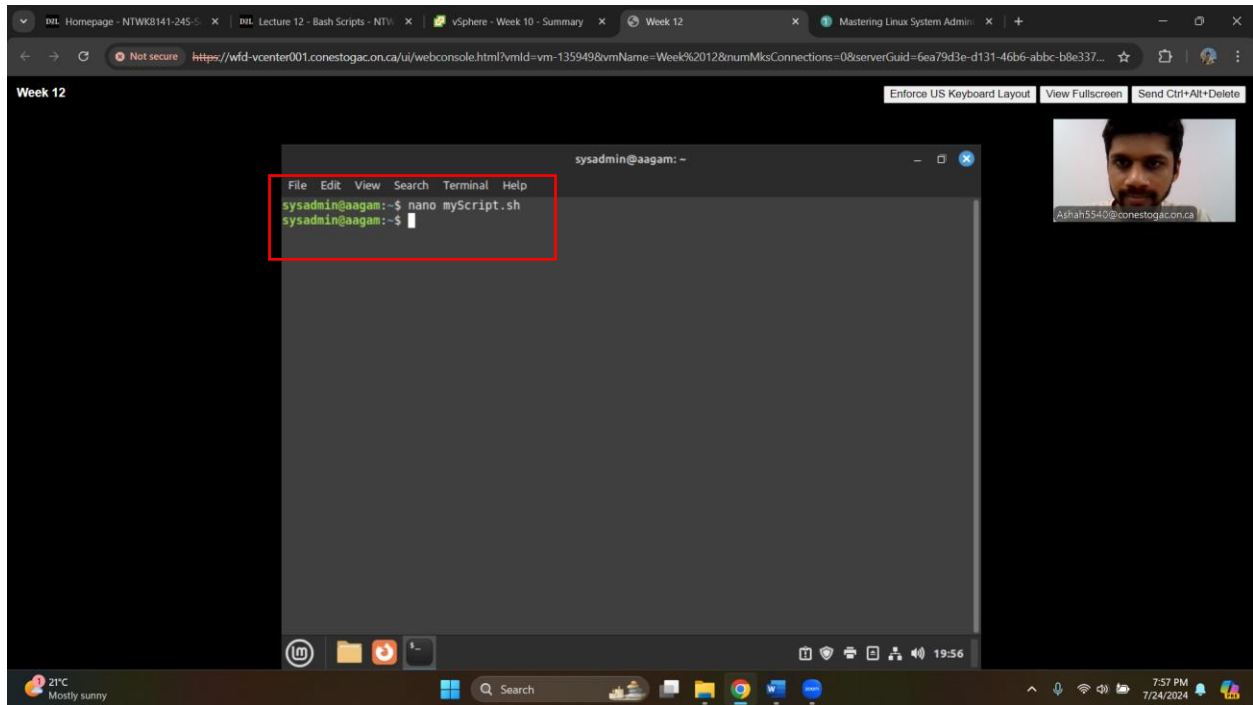
# A simple script that uses variables - Aagam Shah

echo "I am logged into the $USER account."
# echo "The filename for this script is $0."
# echo "The input for this file is $1."

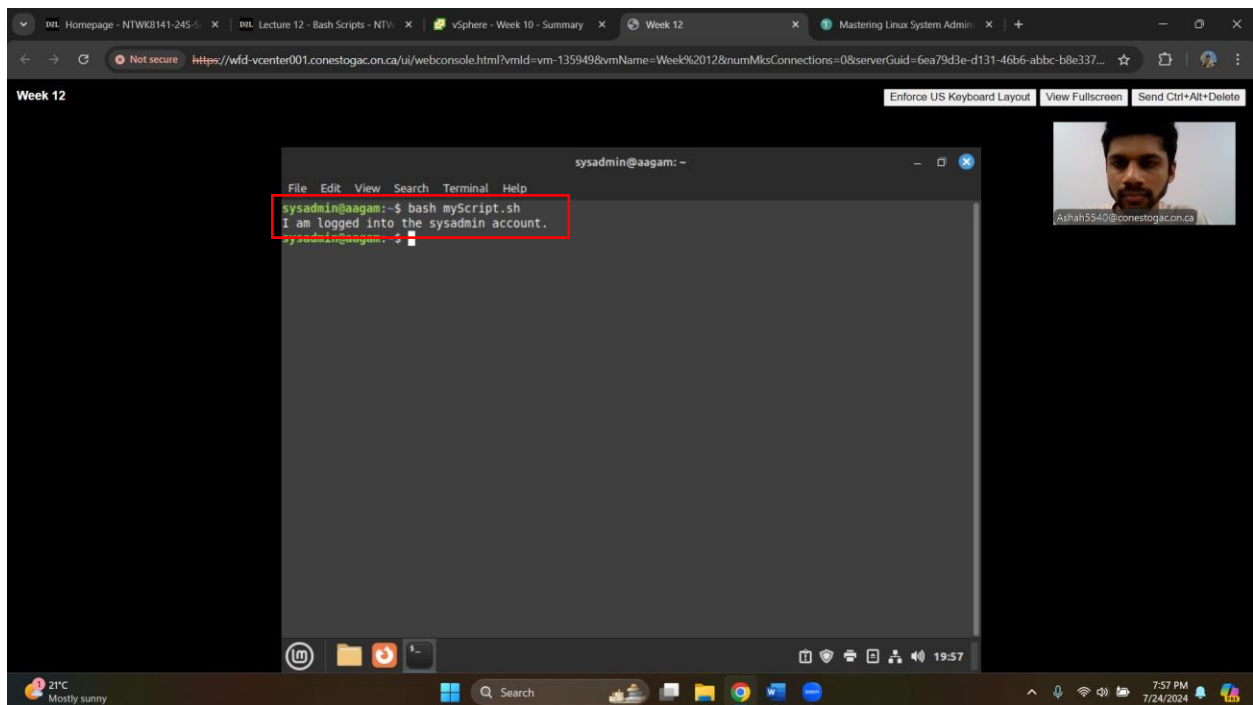
exit
```

The rest of the interface, including the browser tabs and Windows taskbar, remains the same.

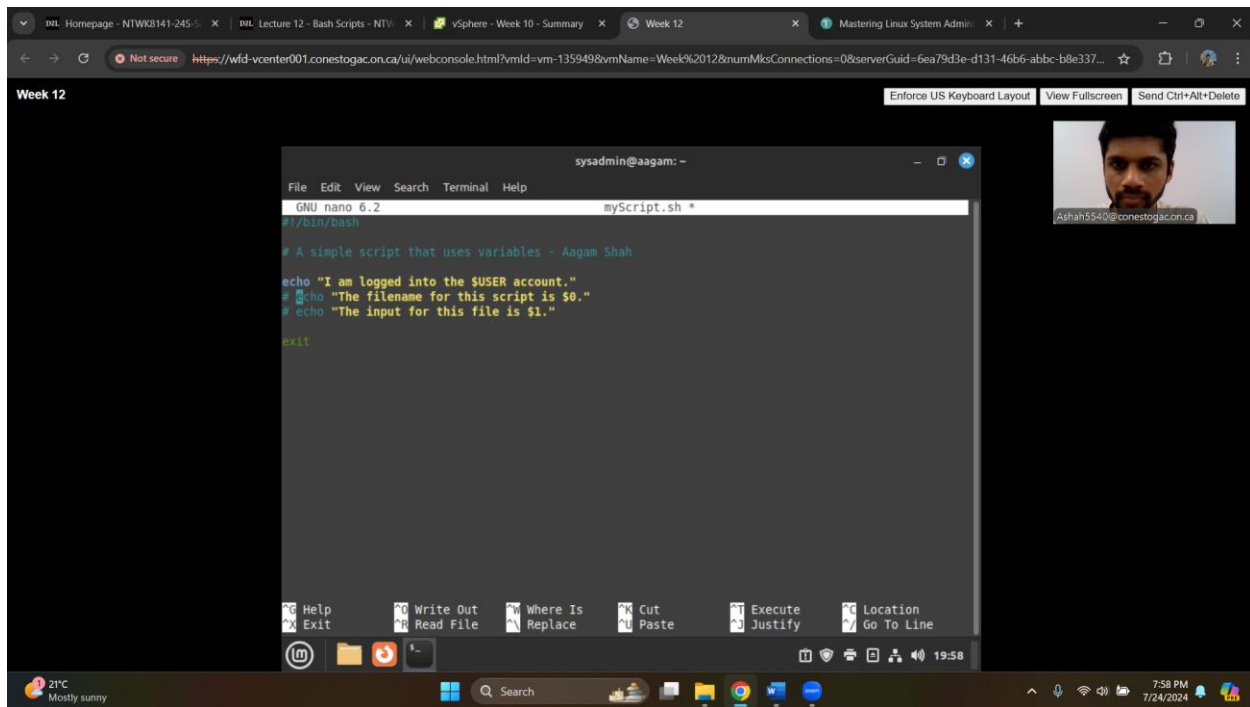
8. Exit the nano text editor by pressing Ctrl+X.



9. Try running the script by typing bash myScript.sh and pressing Enter.



10. If the script did not run successfully, go back and edit the file, making any needed corrections.



The screenshot shows a web browser window with multiple tabs. The active tab is titled "Week 12" and shows a URL from "wfd-vcenter001.conestogac.on.ca". The main content area displays a terminal window titled "sysadmin@aagam: -". Inside the terminal, a nano editor is open, editing a file named "myScript.sh". The script content is as follows:

```
GNU nano 6.2 myScript.sh *
#!/bin/bash

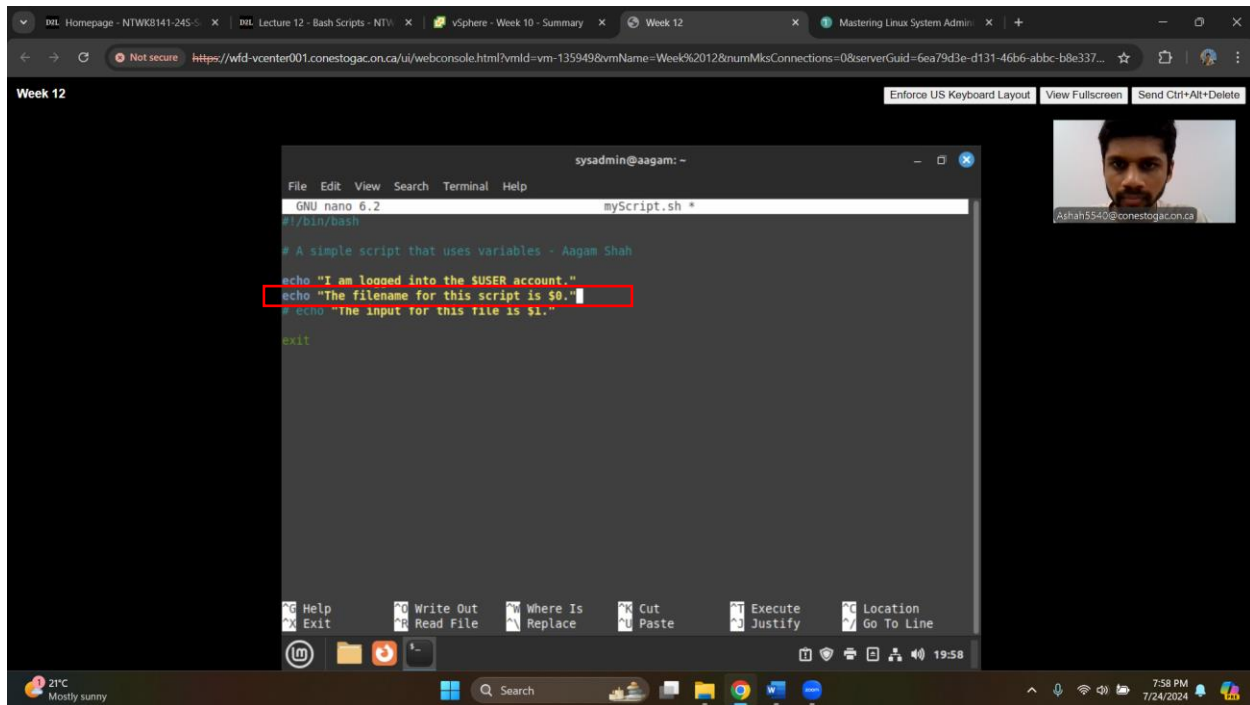
# A simple script that uses variables - Aagam Shah

echo "I am logged into the $USER account."
# echo "The filename for this script is $0."
# echo "The input for this file is $1."

exit
```

The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". At the bottom of the terminal window, there are icons for "Help", "Exit", "Write Out", "Read File", "Where Is", "Replace", "Cut", "Paste", "Execute", "Justify", and "Location". The system tray at the bottom of the browser window shows the temperature as 21°C, the weather as "Mostly sunny", and the time as 7:58 PM on 7/24/2024.

11. Once you have the script running correctly, try adding a new echo command that uses the \$0 parameter value, which is the script's filename.



The screenshot shows the same web browser window as before, but the terminal window now shows the script being edited in the nano editor. The script content is as follows:

```
GNU nano 6.2 myScript.sh *
#!/bin/bash

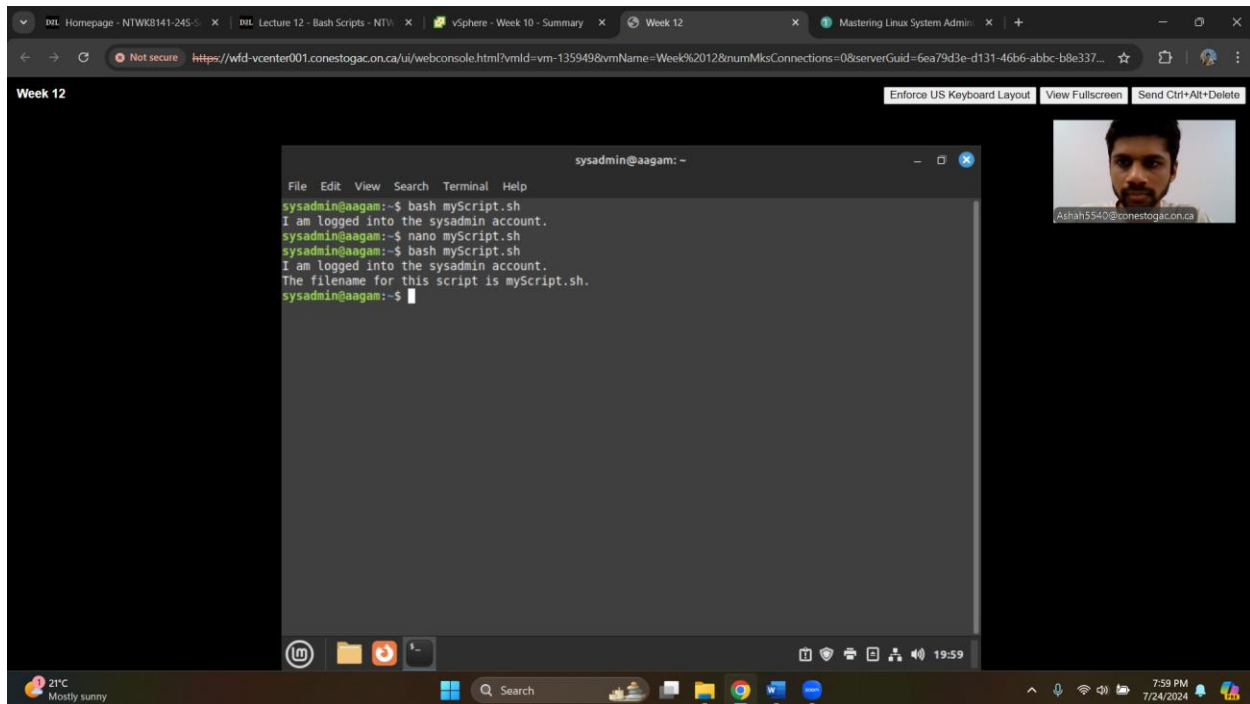
# A simple script that uses variables - Aagam Shah

echo "I am logged into the $USER account."
echo "The filename for this script is $0."
# echo "The input for this file is $1."

exit
```

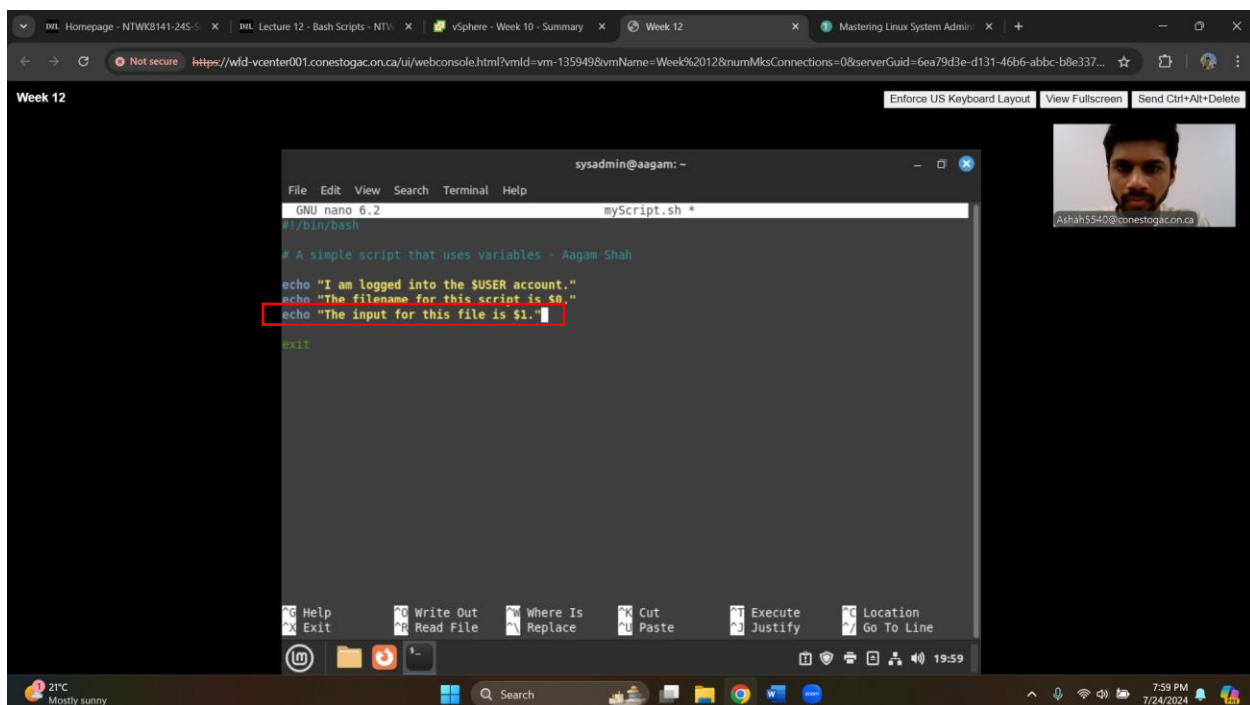
The line `echo "The filename for this script is $0."` is highlighted with a red box. The terminal window and system tray information are the same as in the previous screenshot.

12. After you get the script modification working with the \$0 parameter value, add another echo command that uses a parameter you pass to the script (\$1).



The screenshot shows a web browser window with a URL starting with `https://wfd-vcenter001.conestogac.on.ca/ui/webconsole.html?vmld=vm-135949&vmName=Week%2012&numMksConnections=0&serverGuid=6ea79d3e-d131-46b6-abb6-b9e337...`. The browser tabs include "Homepage - NTWK8141-245", "Lecture 12 - Bash Scripts - NTWK", "vSphere - Week 10 - Summary", "Week 12", and "Mastering Linux System Admin". The page title is "Week 12". In the top right corner, there are buttons for "Enforce US Keyboard Layout", "View Fullscreen", and "Send Ctrl+Alt+Delete". A small video feed of a man is visible in the top right corner. The main content is a terminal window titled "sysadmin@aagam: ~". The terminal shows the following commands and output:

```
sysadmin@aagam:~$ bash myScript.sh
I am logged into the sysadmin account.
sysadmin@aagam:~$ nano myScript.sh
sysadmin@aagam:~$ bash myScript.sh
I am logged into the sysadmin account.
The filename for this script is myScript.sh.
sysadmin@aagam:~$
```



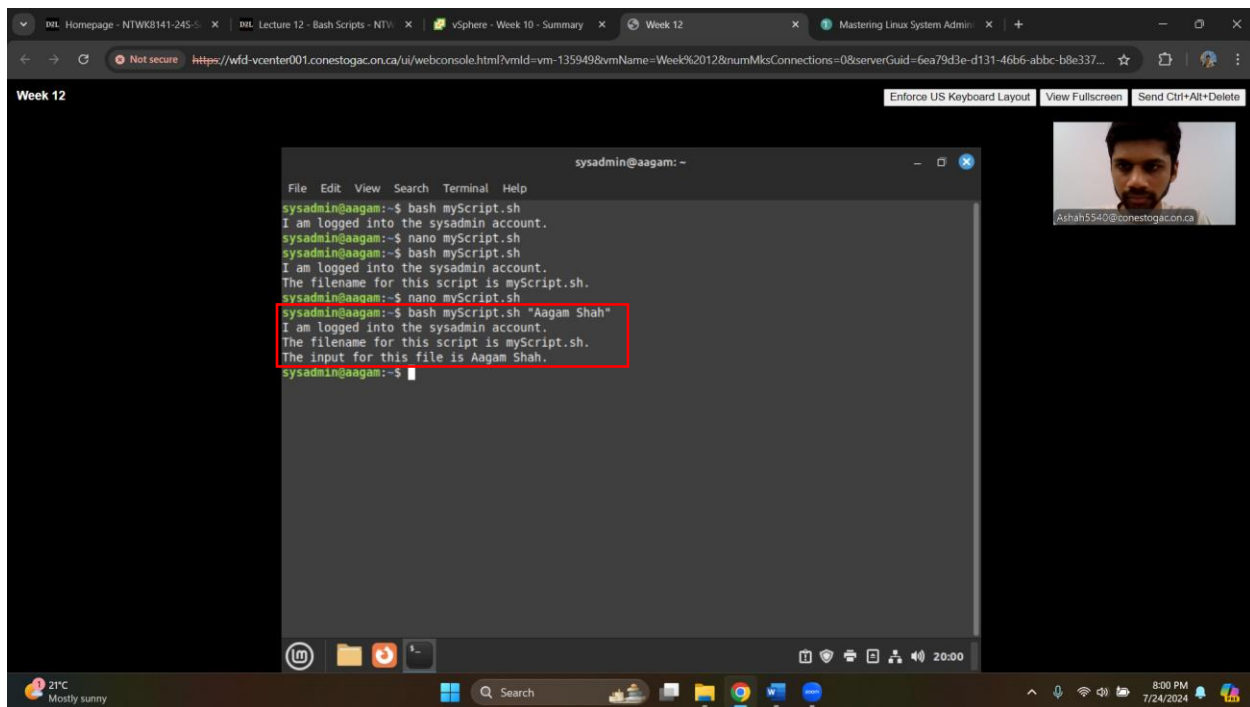
The screenshot shows the same web browser window as the previous one. The terminal window now shows the contents of the `myScript.sh` file being edited in nano. The file content is:

```
GNU nano 6.2 myScript.sh
# /bin/bash

# A simple script that uses variables - Aagam Shah

echo "I am logged into the $USER account."
echo "The filename for this script is $0."
echo "The input for this file is $1."
exit
```

The line `echo "The input for this file is $1."` is highlighted with a red box. At the bottom of the terminal window, there is a menu bar with options: Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, and Go To Line. The bottom status bar shows "21°C Mostly sunny", a search bar, and the time "7:59 PM 7/24/2024".

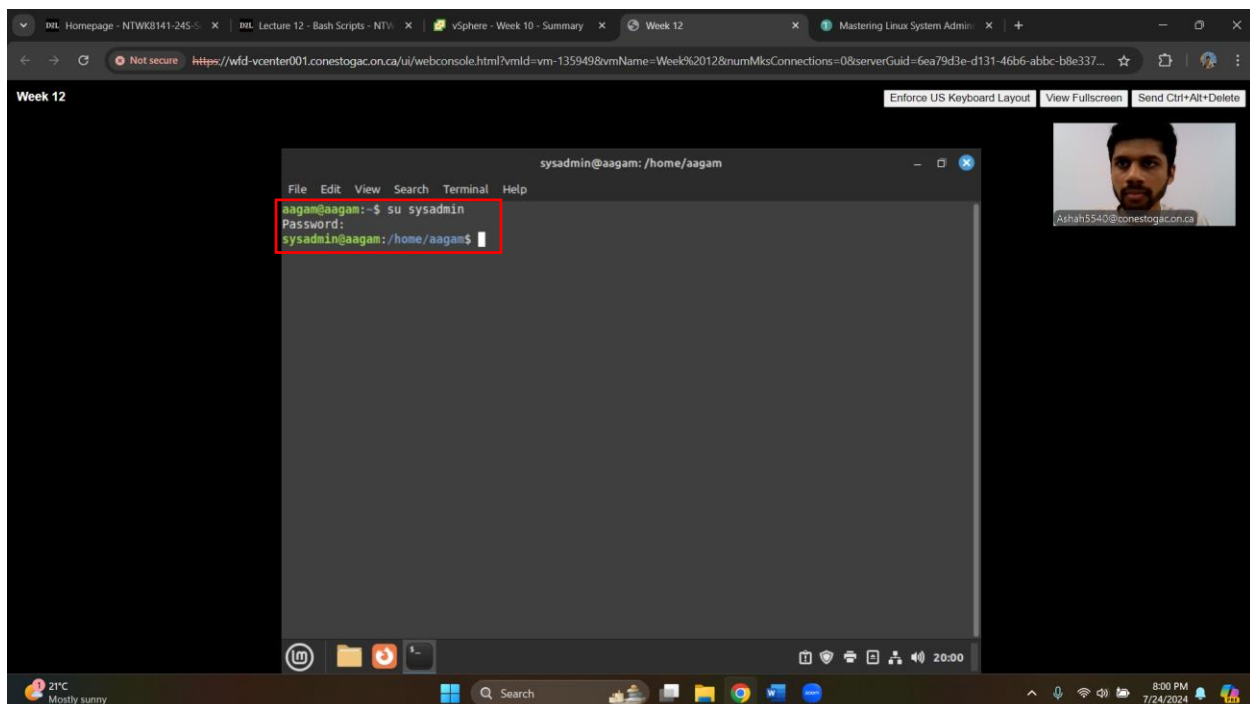


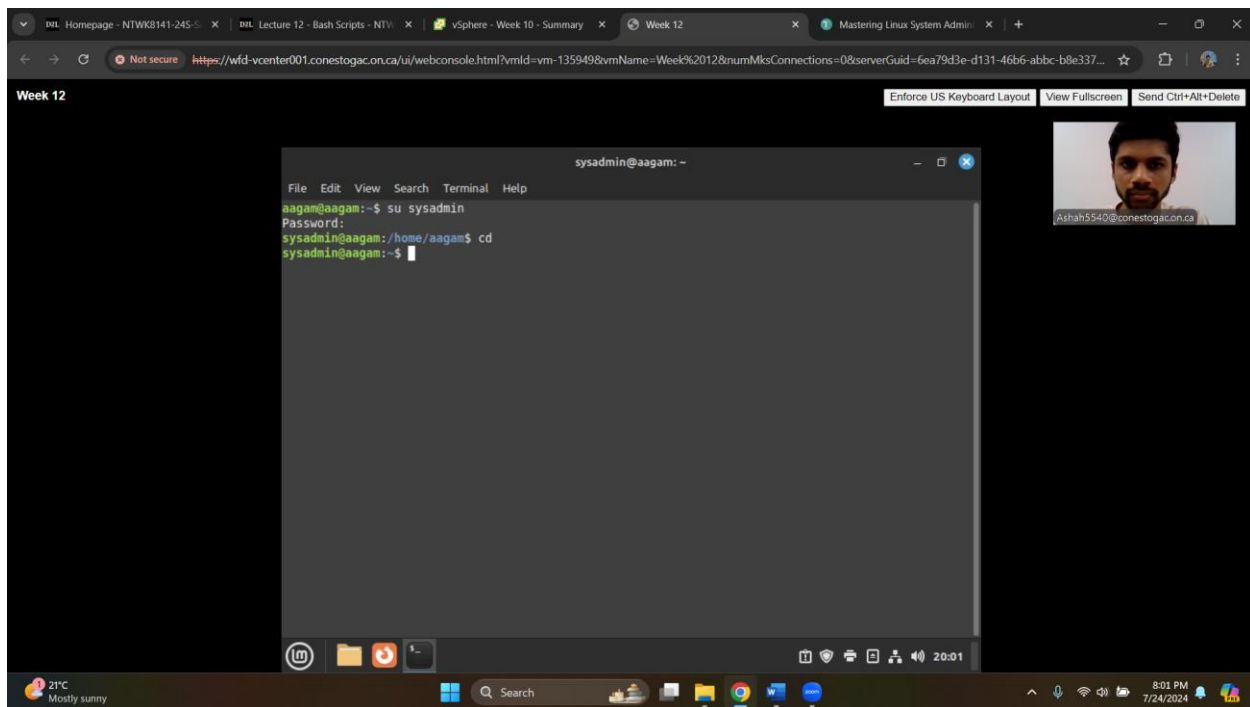
In Class Task: loops

Complete the Real World Scenario: Adding Conditional Expressions and Loops to a Shell Script in Ch 19

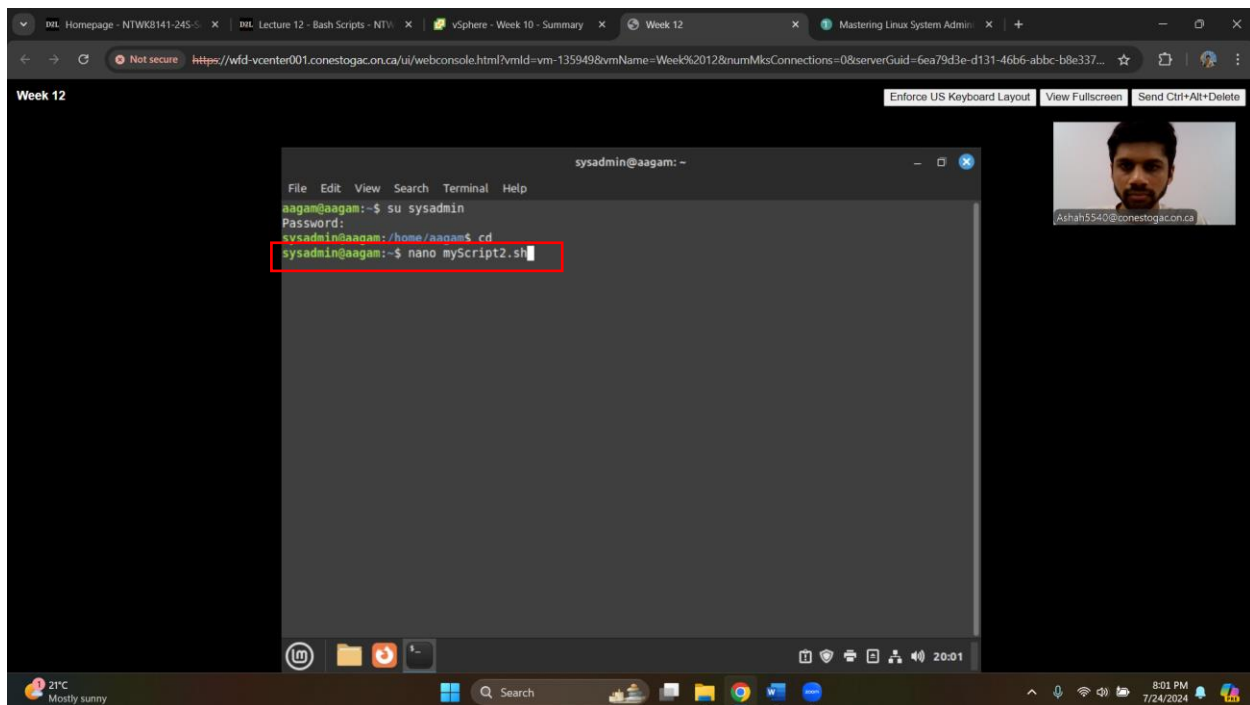
ADDING CONDITIONAL EXPRESSIONS AND LOOPS TO A SHELL SCRIPT

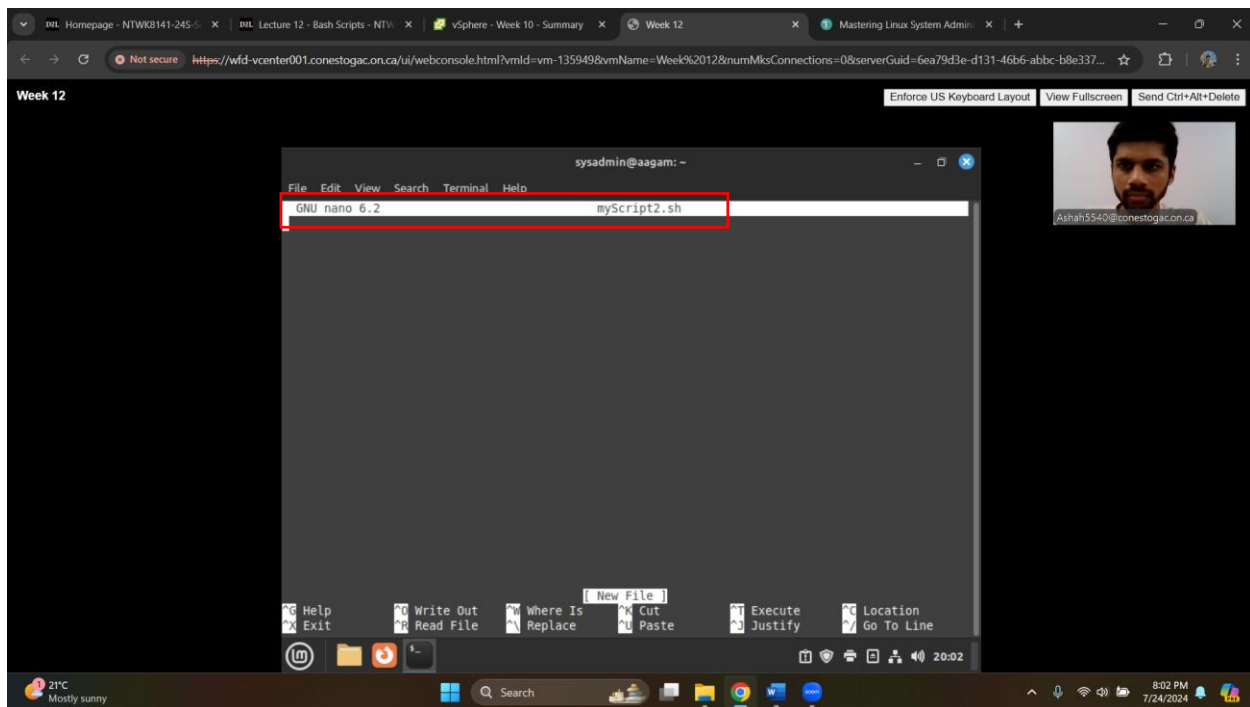
1. Log into your Linux system, using the sysadmin account and the password you created for it.



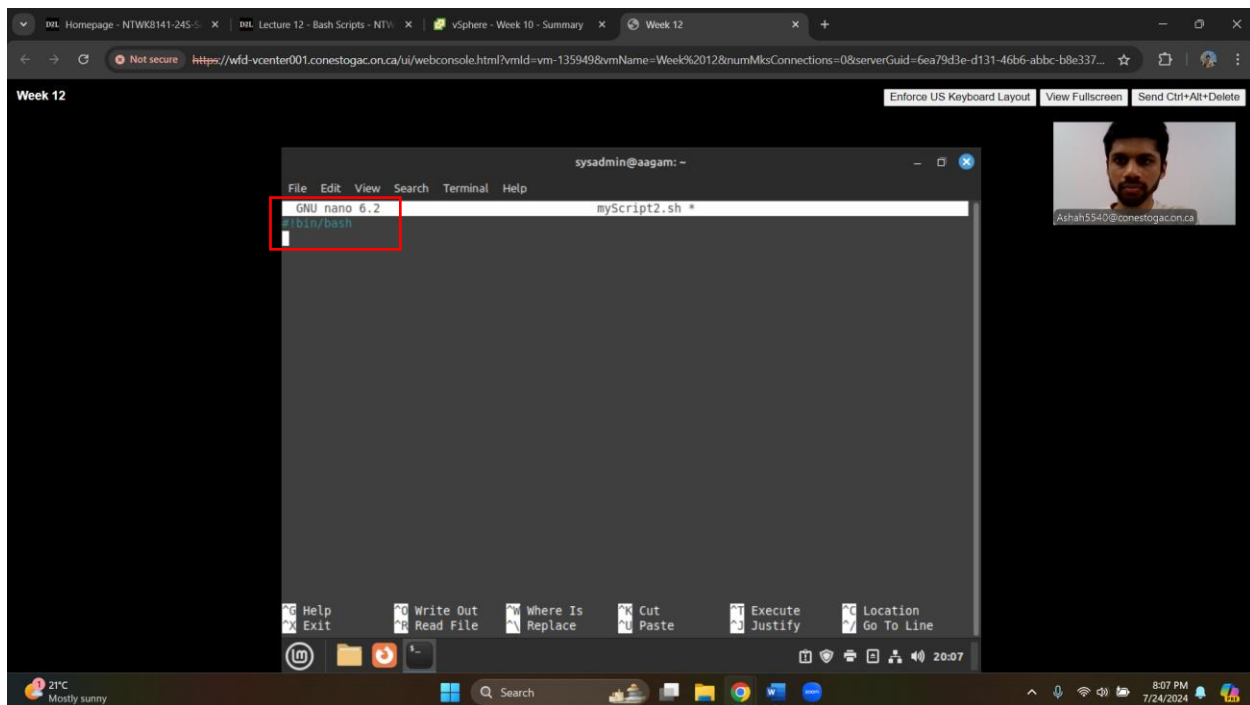


2. Start creating a shell script by typing `nano myScript2.sh` and pressing Enter. This creates the file `myScript2.sh`, and puts you into the nano text editor to start editing it.

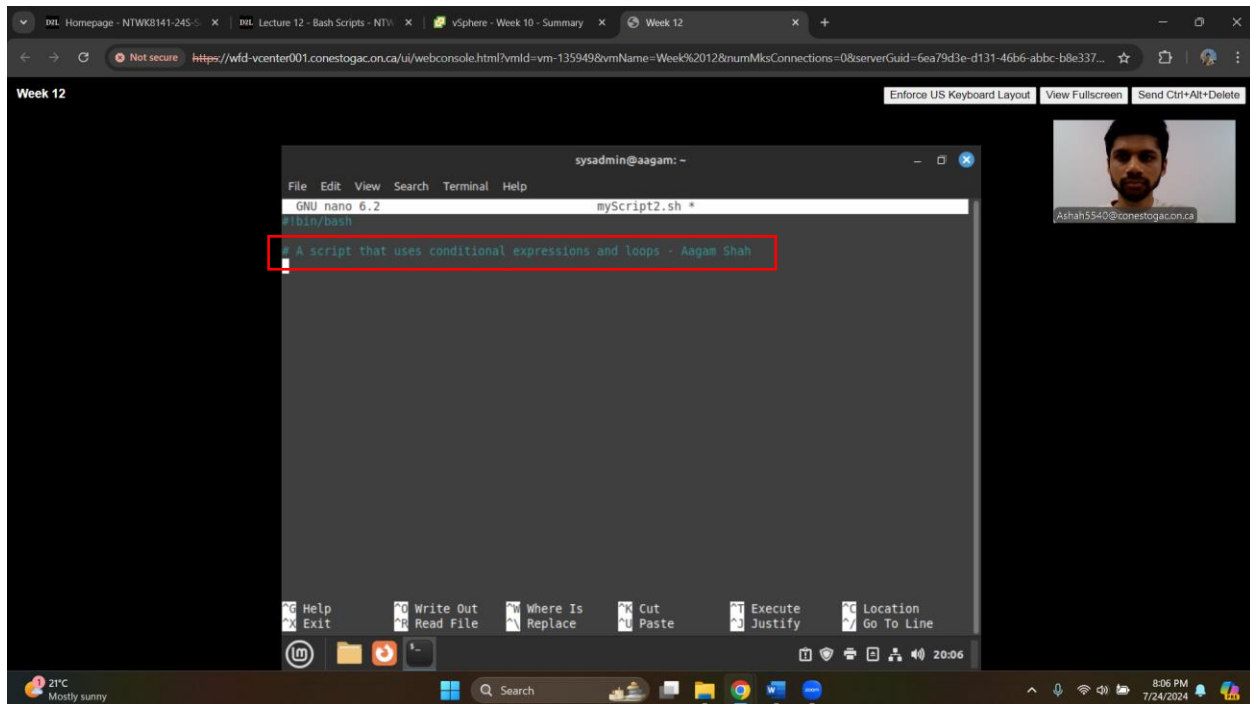




3. On the file's first line, type `#!/bin/bash` and press Enter to select the Bash shell to run the script when it is executed.



4. On the file's second line, type # A script that uses conditional expressions and loops and press Enter. You can add your name to the end of the comment line, if you want to.

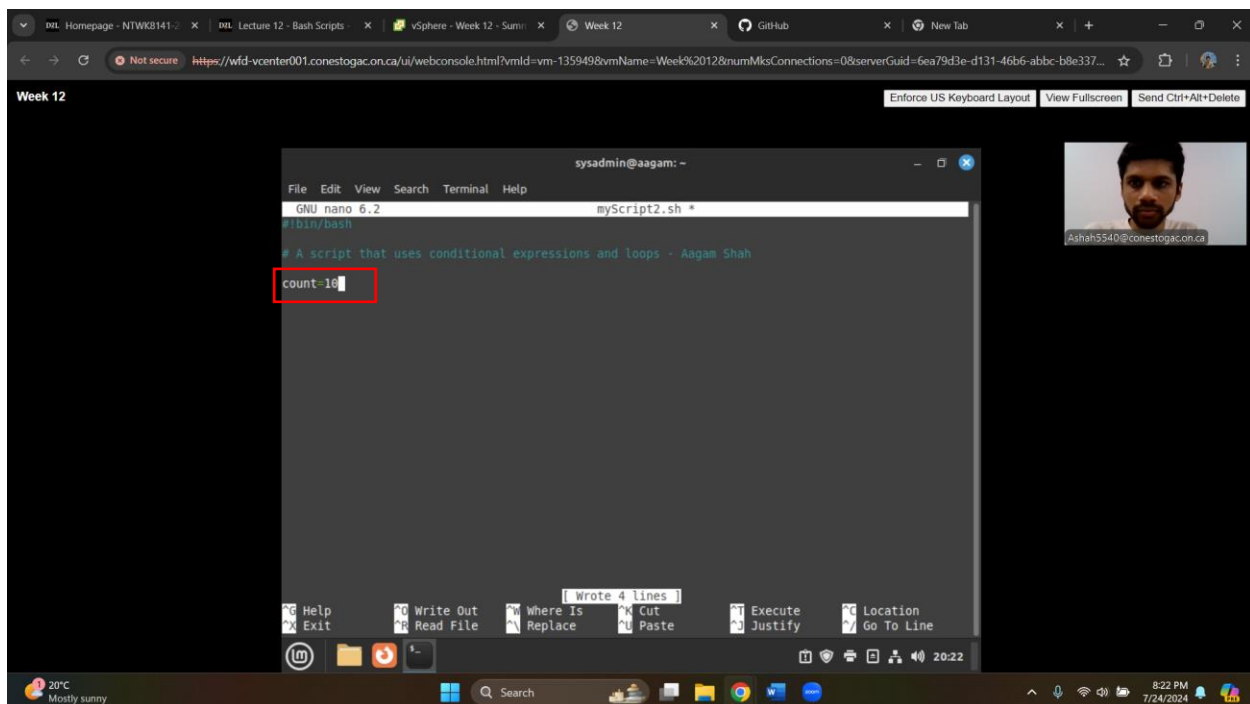


The screenshot shows a web browser window with a URL from vcenter001.conestogac.on.ca. The main content is a terminal window titled 'sysadmin@aagam: -' running 'myScript2.sh'. The terminal shows the 'nano' editor with the following content:

```
GNU nano 6.2 myScript2.sh *
#bin/bash
# A script that uses conditional expressions and loops - Aagam Shah
```

The third line is highlighted with a red box. The bottom of the terminal window shows a menu with options like 'Help', 'Exit', 'Write Out', 'Read File', 'Where Is', 'Replace', 'Cut', 'Paste', 'Execute', 'Justify', and 'Location', 'Go To Line'. The system tray at the bottom shows a temperature of 21°C and the date 7/24/2024.

5. On the third line of the file, create a variable and give it a value by typing count=10 and pressing Enter.

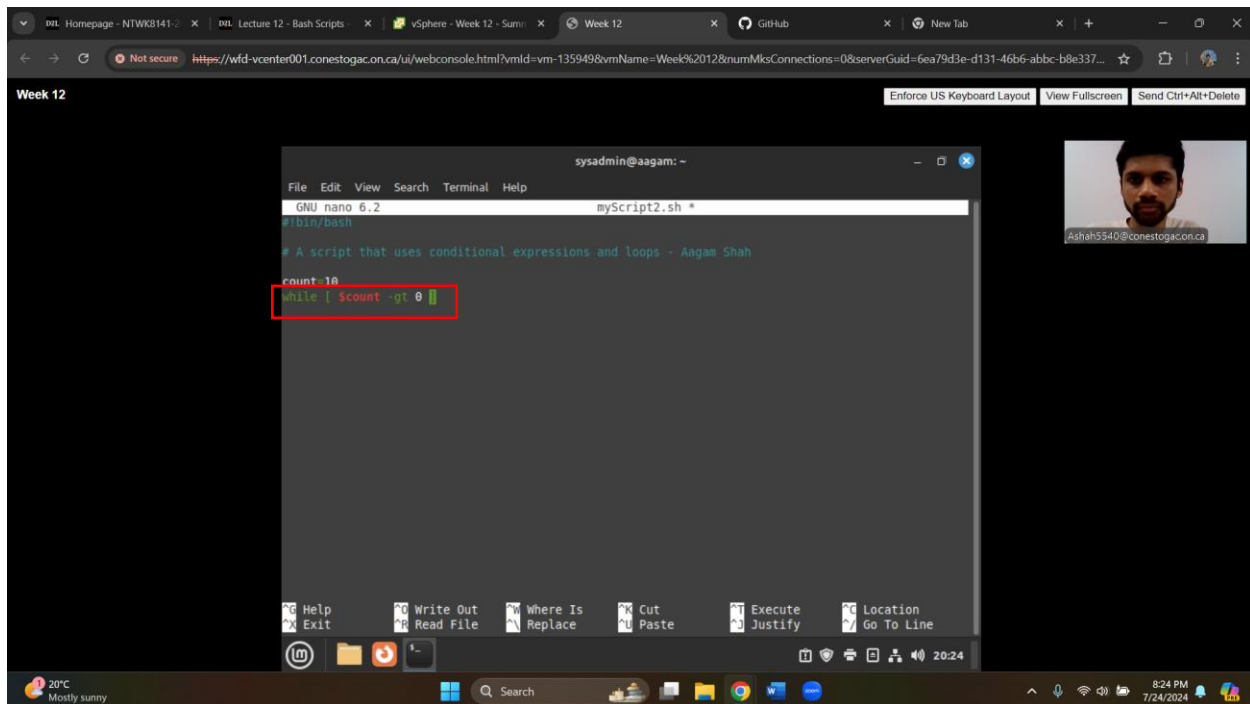


The screenshot shows the same web browser window as before, but the terminal window now shows the following content:

```
GNU nano 6.2 myScript2.sh *
#bin/bash
# A script that uses conditional expressions and loops - Aagam Shah
count=10
```

The fourth line, 'count=10', is highlighted with a red box. The bottom of the terminal window shows a menu with options like 'Help', 'Exit', 'Write Out', 'Read File', 'Where Is', 'Replace', 'Cut', 'Paste', 'Execute', 'Justify', and 'Location', 'Go To Line'. The system tray at the bottom shows a temperature of 20°C and the date 7/24/2024.

6. On the next file line, begin a while loop by typing `while [$count -gt 0]` and pressing Enter.



The screenshot shows a web browser window with a URL starting with `https://wfd-vcenter001.conestogac.on.ca/ui/webconsole.html?vmId=vm-135949&vmName=Week%2012&numMksConnections=0&serverGuid=6ea79d3e-d131-46b6-abbc-b8e337...`. The browser tabs include "Week 12" and "vSphere - Week 12 - Summary". The main content area displays a terminal window titled "sysadmin@aagam: -" with a nano editor open. The editor shows the following code:

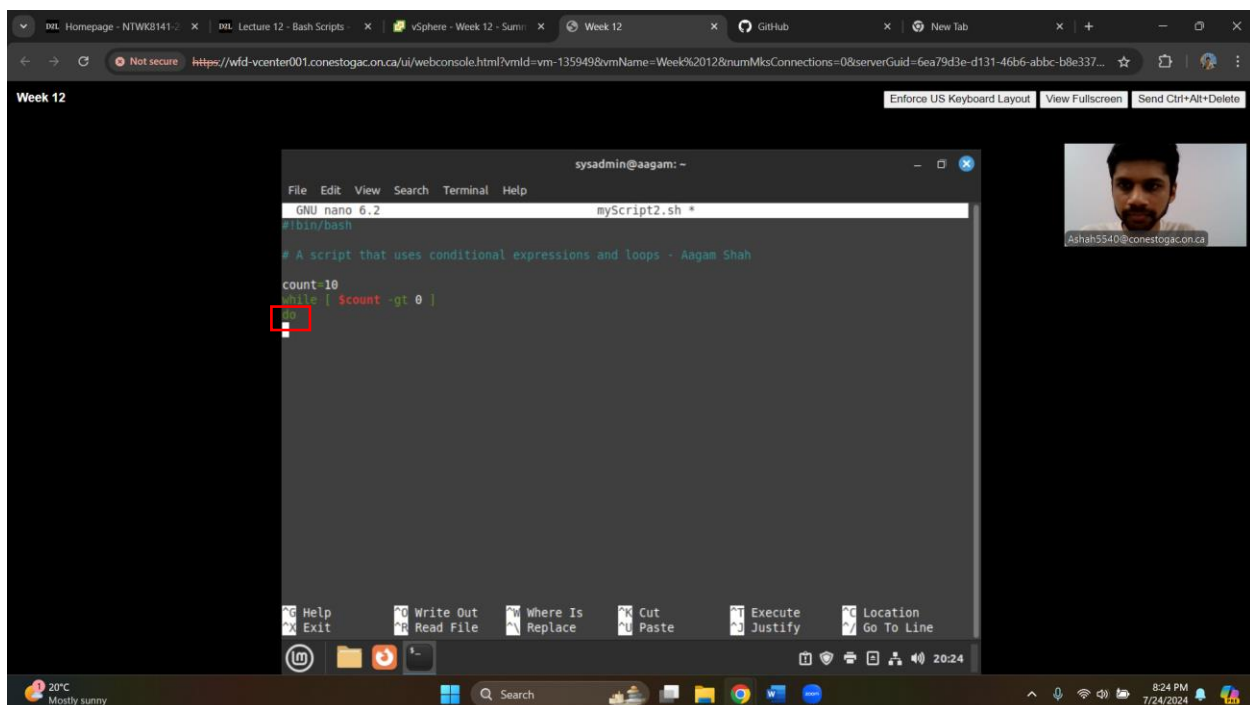
```
File Edit View Search Terminal Help
GNU nano 6.2 myScript2.sh *
#bin/bash

# A script that uses conditional expressions and loops - Aagam Shah

count=10
while [ $count -gt 0 ]
```

The line `while [$count -gt 0]` is highlighted with a red box. The terminal window also shows a video feed of a person in the top right corner.

7. On the file's next line, type `do` and press Enter.



The screenshot shows the same web browser window as before. The nano editor now shows the following code:

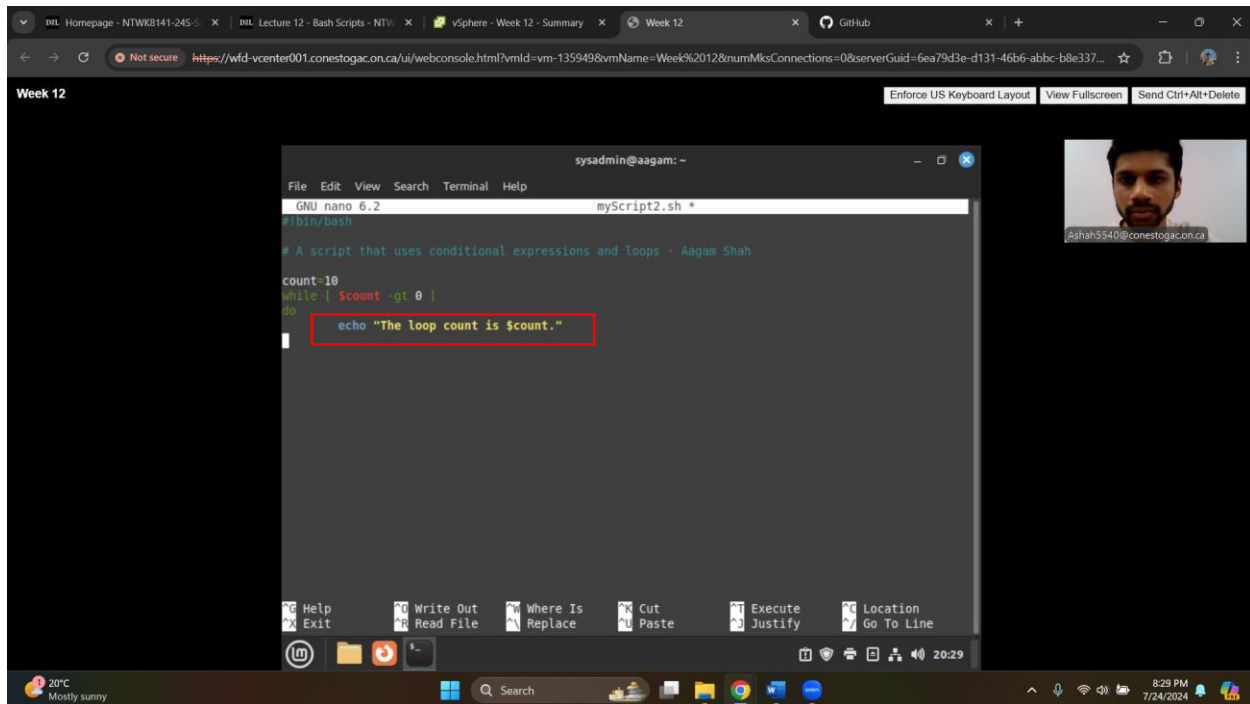
```
File Edit View Search Terminal Help
GNU nano 6.2 myScript2.sh *
#bin/bash

# A script that uses conditional expressions and loops - Aagam Shah

count=10
while [ $count -gt 0 ]
do
```

The line `do` is highlighted with a red box. The terminal window also shows a video feed of a person in the top right corner.

8. Now add a command to the loop by pressing Tab, typing `echo The loop count is $count`, and pressing Enter.



The screenshot shows a web browser window with a URL that includes a VM ID and a server GUID. The page title is "Week 12". A terminal window titled "sysadmin@aagam: ~" is open, displaying a nano editor editing a file named "myScript2.sh". The script content is as follows:

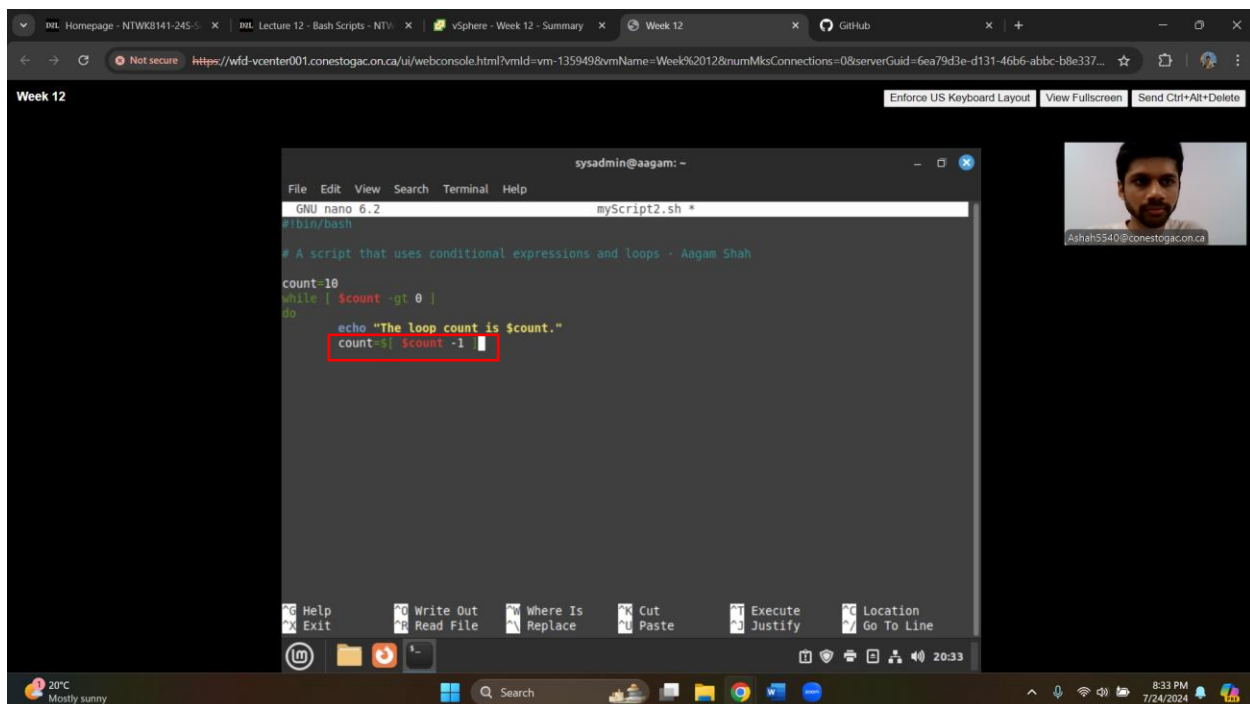
```
GNU nano 6.2 myScript2.sh *
#bin/bash

# A script that uses conditional expressions and loops - Aagam Shah

count=10
while [ $count -gt 0 ]
do
    echo "The loop count is $count."
done
```

The line `echo "The loop count is $count."` is highlighted with a red box. The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The bottom of the terminal window shows a status bar with "Help", "Exit", "Write Out", "Read File", "Where Is", "Replace", "Cut", "Paste", "Execute", "Justify", and "Location", "Go To Line". The bottom of the browser window shows a taskbar with a search bar and various icons. The system tray at the bottom right shows the date and time as "8:29 PM 7/24/2024".

9. Add a second command to the loop by pressing Tab, typing `count=[$count - 1]`, and pressing Enter.



The screenshot shows the same web browser window as the previous one. The terminal window titled "sysadmin@aagam: ~" is open, displaying the nano editor editing the file "myScript2.sh". The script content is as follows:

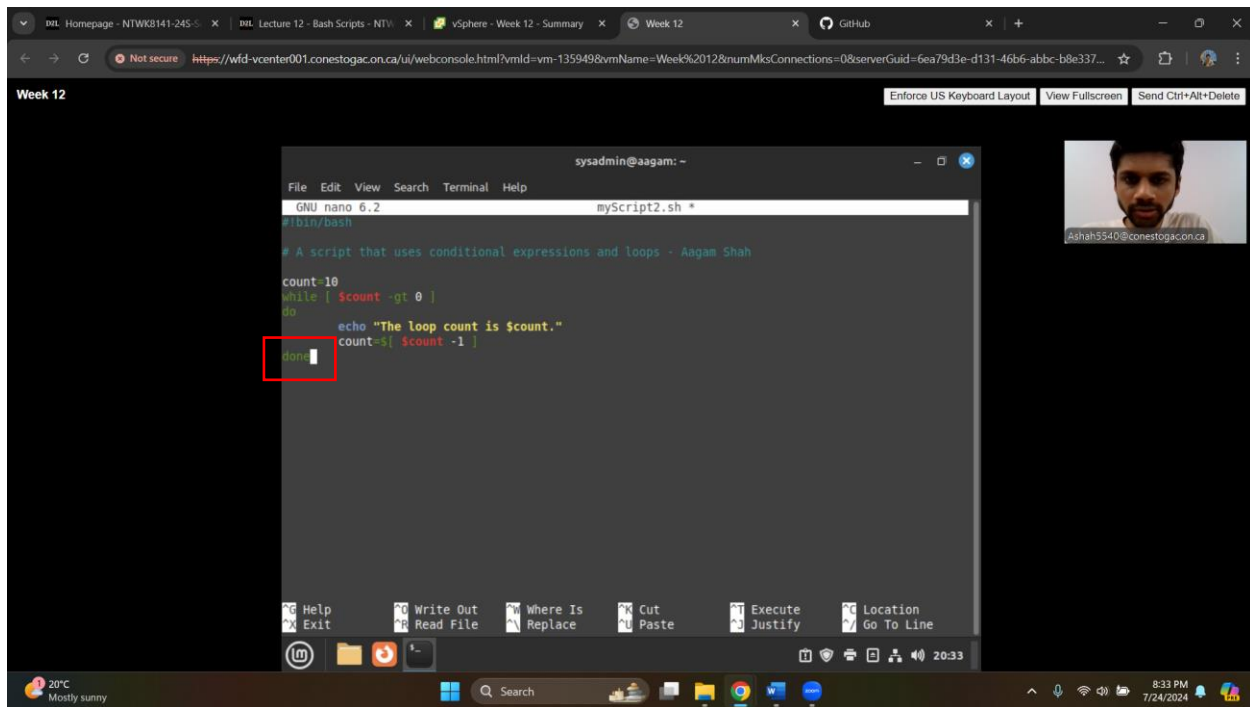
```
GNU nano 6.2 myScript2.sh *
#bin/bash

# A script that uses conditional expressions and loops - Aagam Shah

count=10
while [ $count -gt 0 ]
do
    echo "The loop count is $count."
    count=[ $count - 1 ]
done
```

The line `count=[$count - 1]` is highlighted with a red box. The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The bottom of the terminal window shows a status bar with "Help", "Exit", "Write Out", "Read File", "Where Is", "Replace", "Cut", "Paste", "Execute", "Justify", and "Location", "Go To Line". The bottom of the browser window shows a taskbar with a search bar and various icons. The system tray at the bottom right shows the date and time as "8:33 PM 7/24/2024".

10. End the loop by typing done and pressing Enter.



The screenshot shows a web browser window with a URL that includes 'wfd-vcenter001.conestogac.on.ca'. The main content area displays a terminal window titled 'sysadmin@aagam: -' running the 'nano myScript2.sh' editor. The script content is as follows:

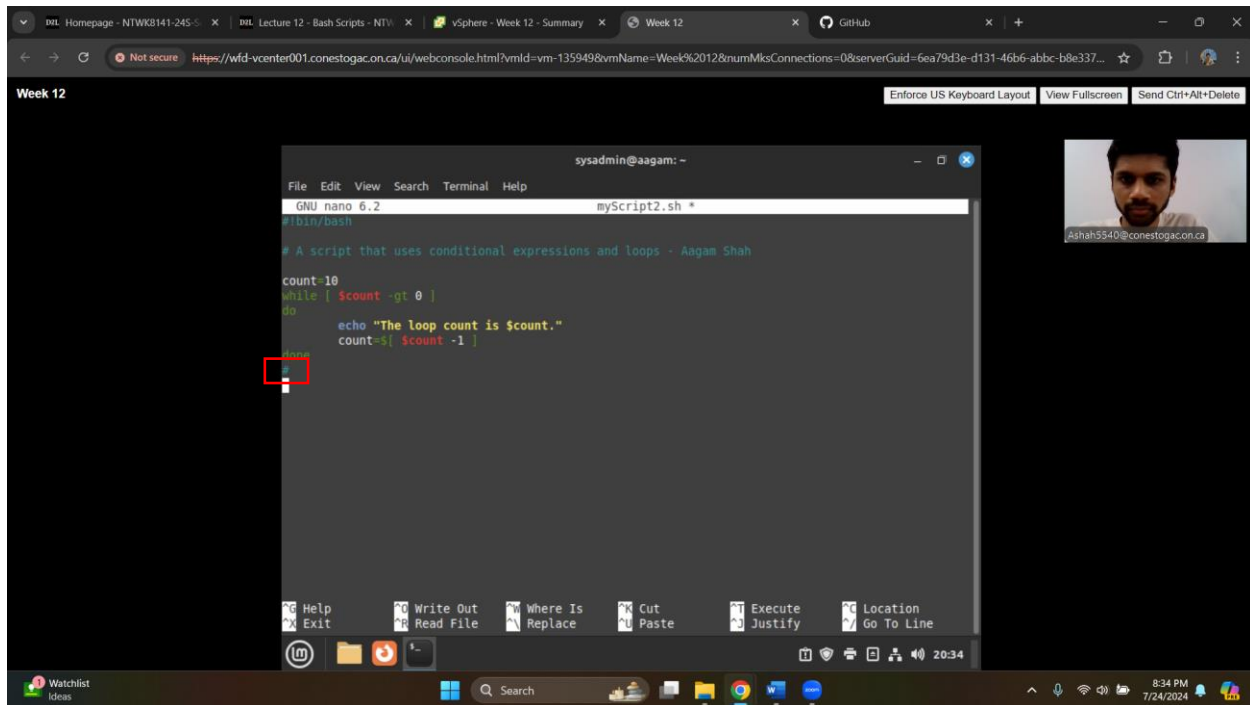
```
GNU nano 6.2 myScript2.sh *
#bin/bash

# A script that uses conditional expressions and loops - Aagam Shah

count=10
while [ $count -gt 0 ]
do
    echo "The loop count is $count."
    count=$((count - 1))
done
```

The word 'done' on the last line is highlighted with a red rectangular box. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. At the bottom of the terminal, there are buttons for 'Help', 'Exit', 'Write Out', 'Read File', 'Where Is', 'Replace', 'Cut', 'Paste', 'Execute', 'Justify', and 'Location Go To Line'. The system tray at the bottom of the browser shows the date and time as '8:33 PM 7/24/2024'.

11. Add a blank comment line by typing # and pressing Enter. This will help the readability of the script.



This screenshot is similar to the previous one, but the script in the nano editor now includes an additional blank comment line at the end:

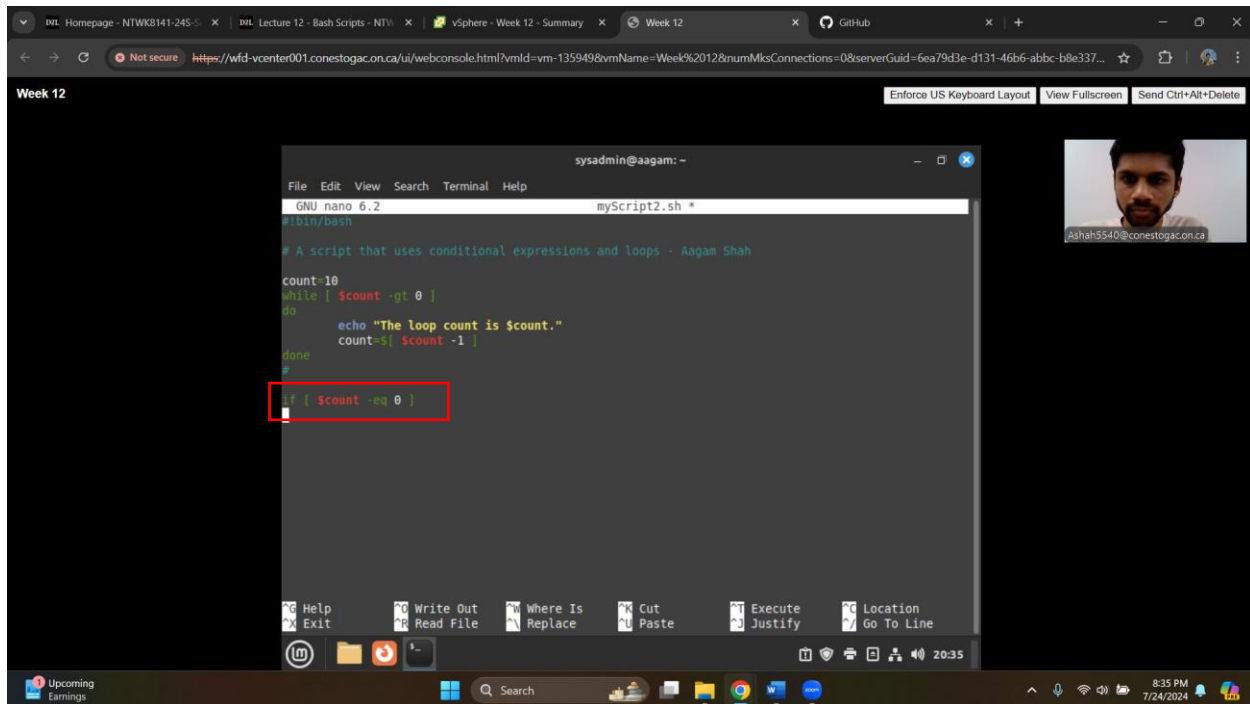
```
GNU nano 6.2 myScript2.sh *
#bin/bash

# A script that uses conditional expressions and loops - Aagam Shah

count=10
while [ $count -gt 0 ]
do
    echo "The loop count is $count."
    count=$((count - 1))
done
#
```

The word 'done' is still highlighted with a red rectangular box. The rest of the terminal window and browser interface are identical to the previous screenshot, with the date and time now showing '8:34 PM 7/24/2024'.

12. On the file's next line, start adding a conditional expression by typing `if [$count -eq 0]` and pressing Enter.



The screenshot shows a terminal window titled "sysadmin@aagam: ~" with a nano editor open. The editor is editing a file named "myScript2.sh". The script content is as follows:

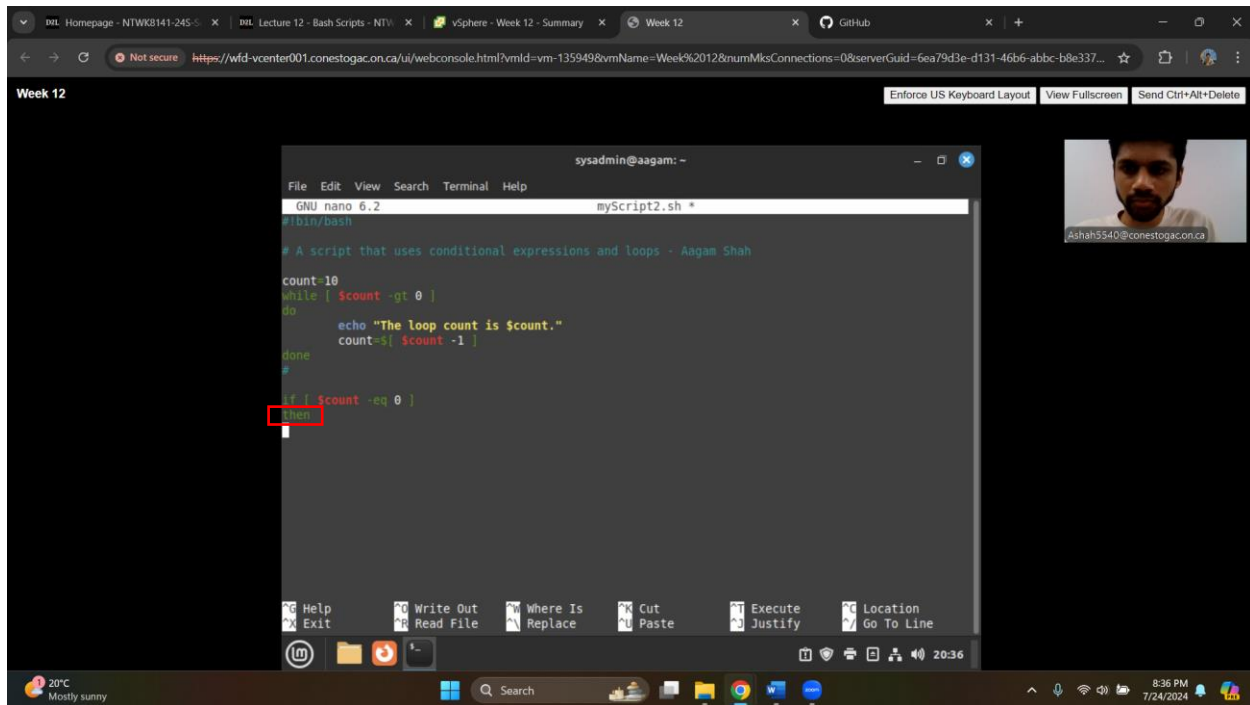
```
#!/bin/bash

# A script that uses conditional expressions and loops - Aagam Shah

count=10
while [ $count -gt 0 ]
do
    echo "The loop count is $count."
    count=$((count - 1))
done
#
if [ $count -eq 0 ]
```

The line `if [$count -eq 0]` is highlighted with a red box. The terminal window is part of a virtual machine interface, and a video feed of the user is visible in the top right corner.

13. Type `then` and press Enter.



The screenshot shows the same terminal window as before, but now the script content is:

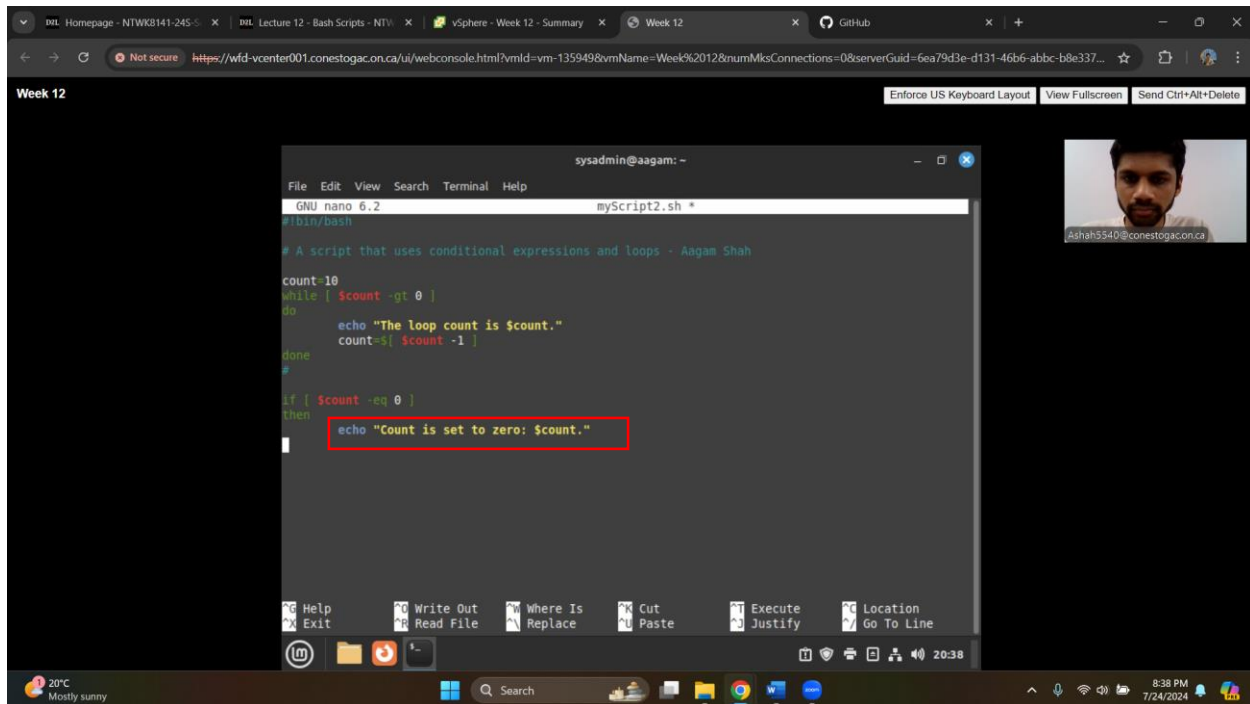
```
#!/bin/bash

# A script that uses conditional expressions and loops - Aagam Shah

count=10
while [ $count -gt 0 ]
do
    echo "The loop count is $count."
    count=$((count - 1))
done
#
if [ $count -eq 0 ]
then
```

The word `then` is highlighted with a red box. The terminal window is part of a virtual machine interface, and a video feed of the user is visible in the top right corner.

14. Add a command to execute, if the test returns a zero exit status (true), by pressing Tab, typing echo "Count is set to zero: \$count", and pressing Enter.



The screenshot shows a terminal window titled 'sysadmin@aagam: ~' with a nano editor open. The editor is editing a file named 'myScript2.sh'. The script content is as follows:

```
#!/bin/bash

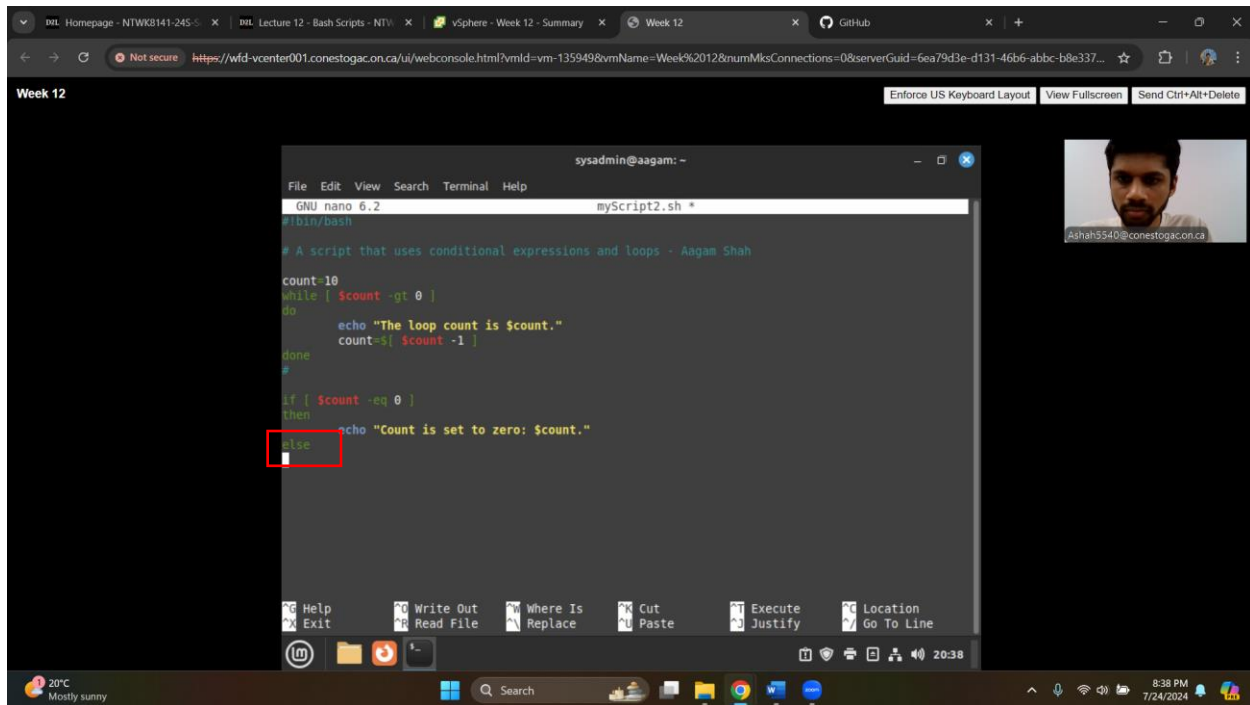
# A script that uses conditional expressions and loops - Aagam Shah

count=10
while [ $count -gt 0 ]
do
    echo "The loop count is $count."
    count=$((count - 1))
done

if [ $count -eq 0 ]
then
    echo "Count is set to zero: $count."
fi
```

The command 'echo "Count is set to zero: \$count."' is highlighted with a red box. The terminal window is part of a web browser interface, and a video feed of a person is visible in the top right corner.

15. Type else and press Enter.



The screenshot shows the same terminal window as before, but now the script has been updated to include an 'else' clause. The script content is as follows:

```
#!/bin/bash

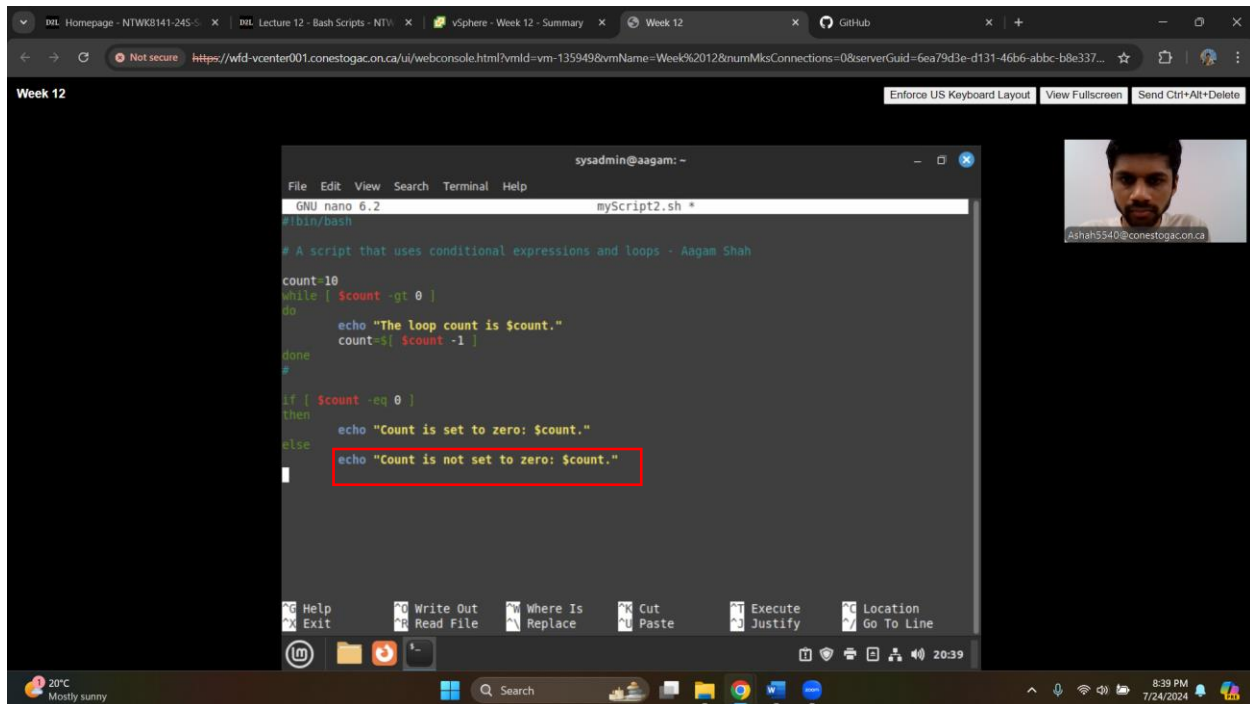
# A script that uses conditional expressions and loops - Aagam Shah

count=10
while [ $count -gt 0 ]
do
    echo "The loop count is $count."
    count=$((count - 1))
done

if [ $count -eq 0 ]
then
    echo "Count is set to zero: $count."
else
    echo "Count is set to zero: $count."
fi
```

The 'else' keyword is highlighted with a red box. The terminal window is part of a web browser interface, and a video feed of a person is visible in the top right corner.

16. Add a command to execute if the test returns a nonzero exit status (false), by pressing Tab, typing `echo "Count is not set to zero: $count"`, and pressing Enter.



The screenshot shows a terminal window titled 'sysadmin@aagam: ~' with a nano editor open. The editor shows a script named 'myScript2.sh'. The script content is as follows:

```
#!/bin/bash

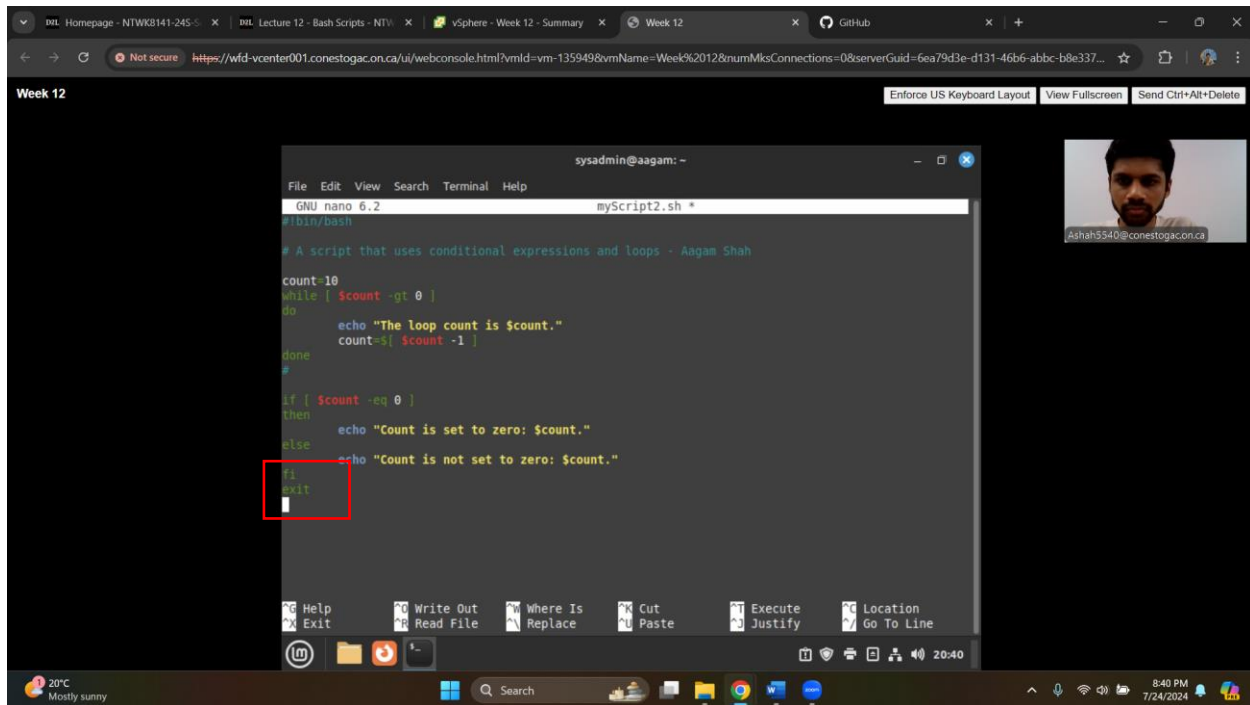
# A script that uses conditional expressions and loops - Aagam Shah

count=10
while [ $count -gt 0 ]
do
    echo "The loop count is $count."
    count=$((count - 1))
done

if [ $count -eq 0 ]
then
    echo "Count is set to zero: $count."
else
    echo "Count is not set to zero: $count."
fi
```

The line `echo "Count is not set to zero: $count."` is highlighted with a red box. The terminal window is part of a web browser interface, and a video feed of a person is visible in the top right corner.

17. On the file's last line, type `exit`.



The screenshot shows the same terminal window as before, but now the script has been updated. The content is as follows:

```
#!/bin/bash

# A script that uses conditional expressions and loops - Aagam Shah

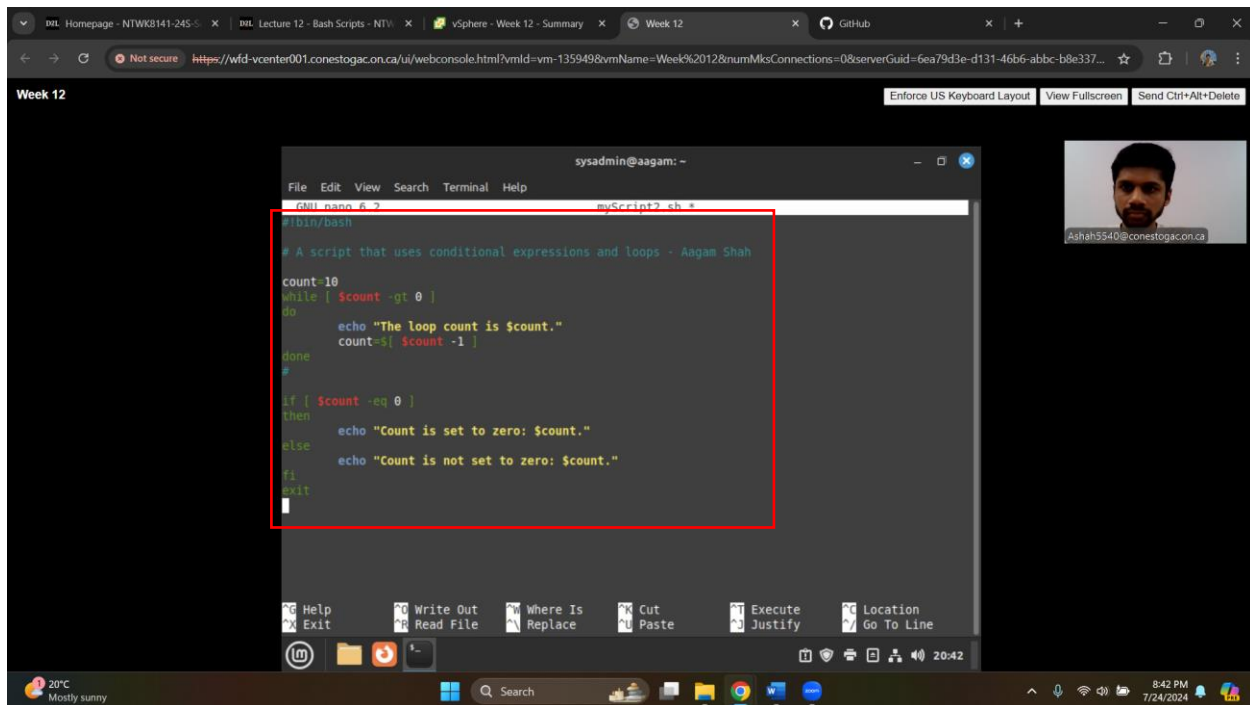
count=10
while [ $count -gt 0 ]
do
    echo "The loop count is $count."
    count=$((count - 1))
done

if [ $count -eq 0 ]
then
    echo "Count is set to zero: $count."
else
    echo "Count is not set to zero: $count."
fi
exit
```

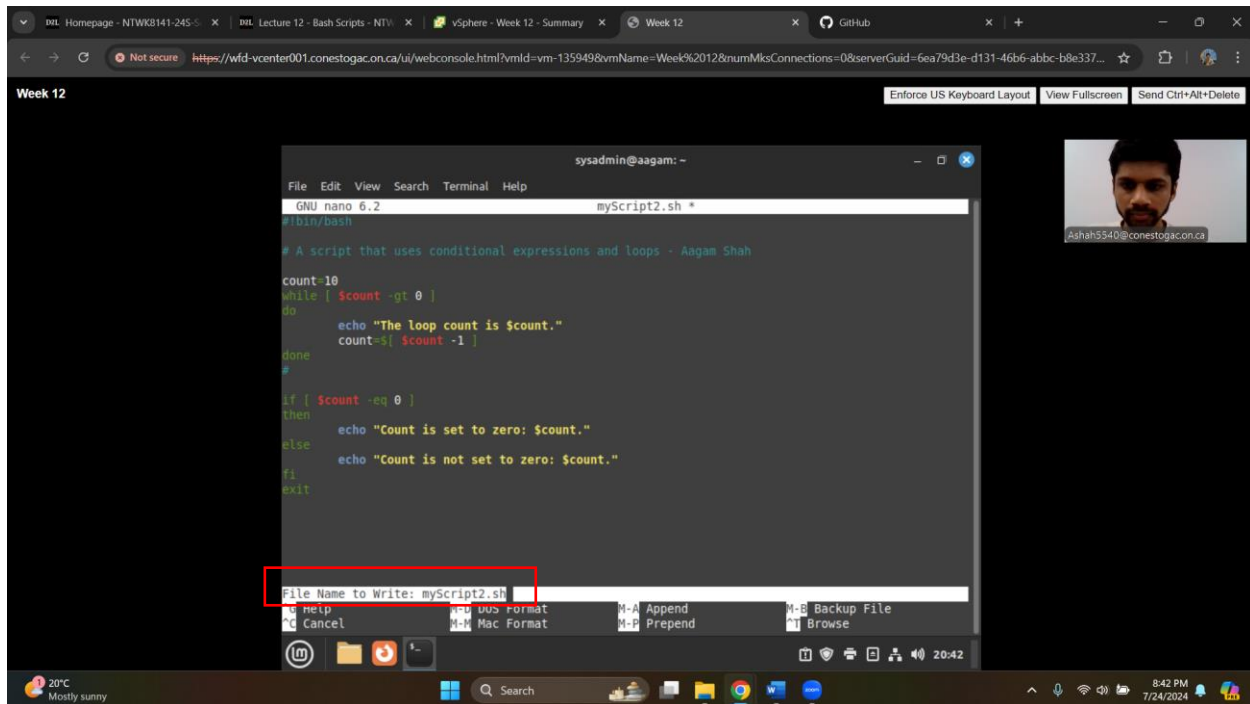
The lines `fi` and `exit` are highlighted with a red box. The terminal window is part of a web browser interface, and a video feed of a person is visible in the top right corner.

18. Before saving your shell script file, review the following code and make sure your code is the same (though your comment lines do not have to match). Make any needed changes.

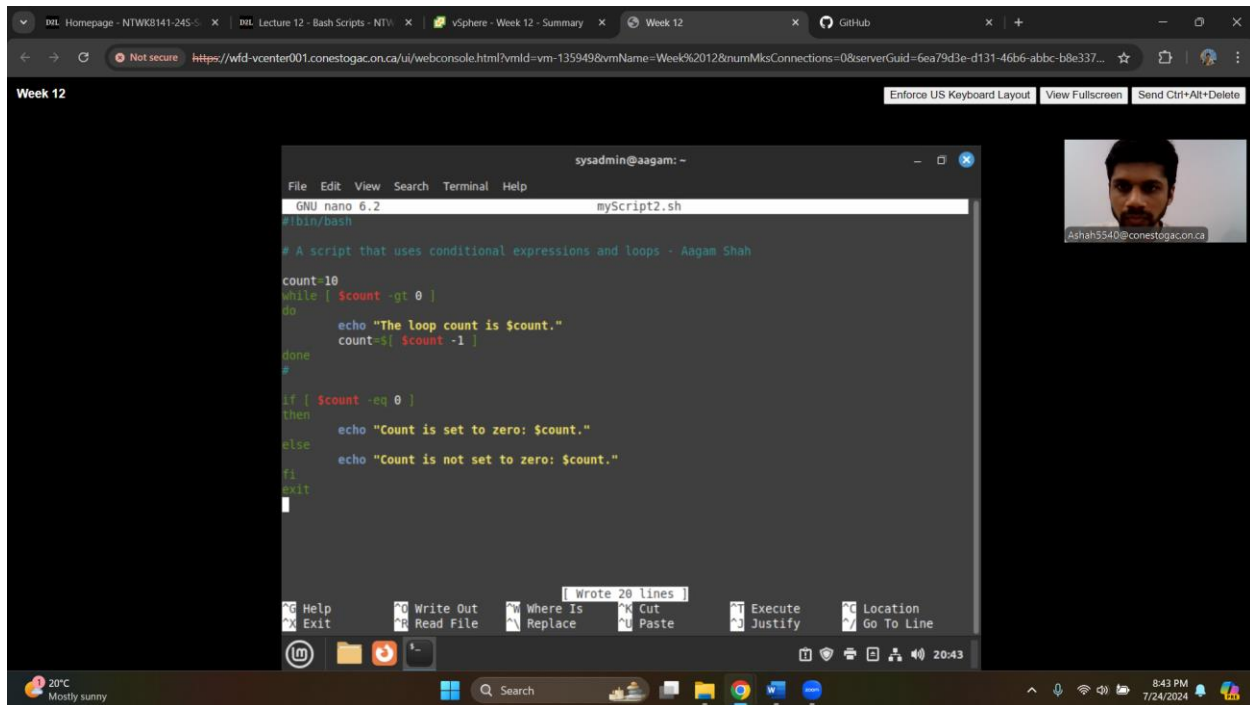
```
#!/bin/bash
# A script that uses conditional expressions and loops
count=10
while [ $count -gt 0 ]
do
    echo "The loop count is $count."
    count=$(( $count - 1 ))
done
#
if [ $count -eq 0 ]
then
    echo "Count is set to zero: $count"
else
    echo "Count is not set to zero: $count"
fi
exit
```

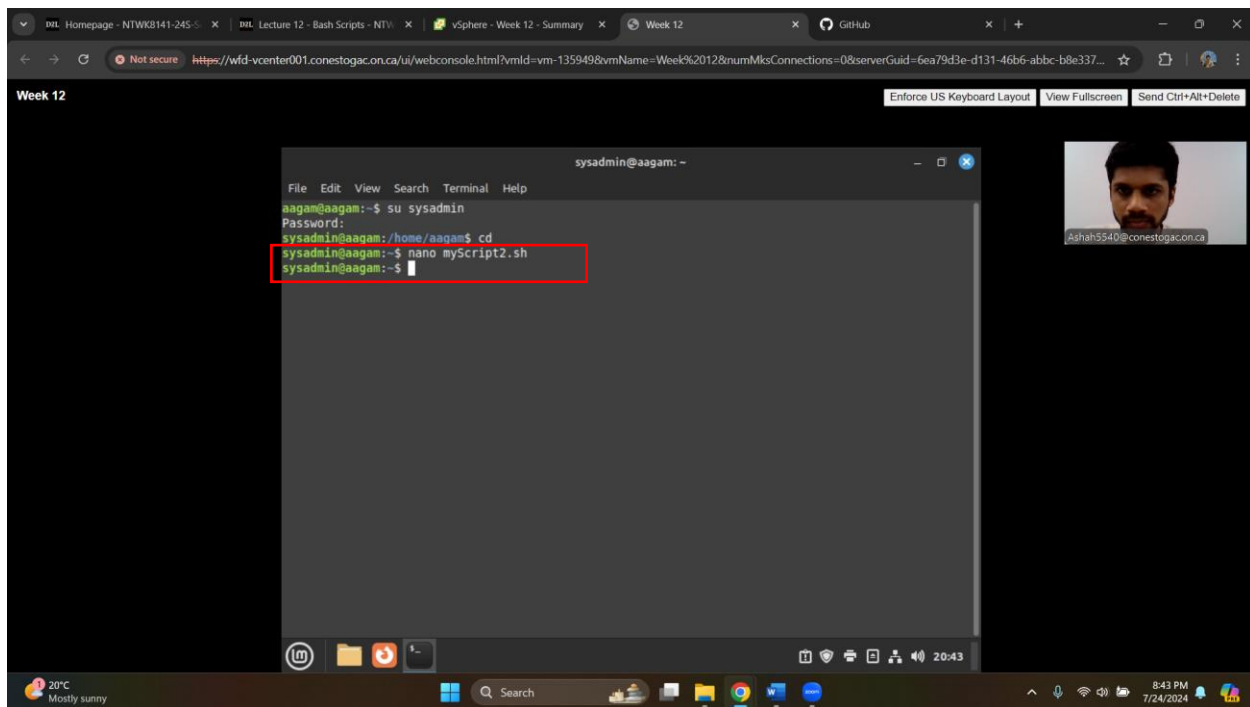


19. Save the entered text to the script file by pressing Ctrl+O and pressing Enter when the file's name displays on the text editor's status line.

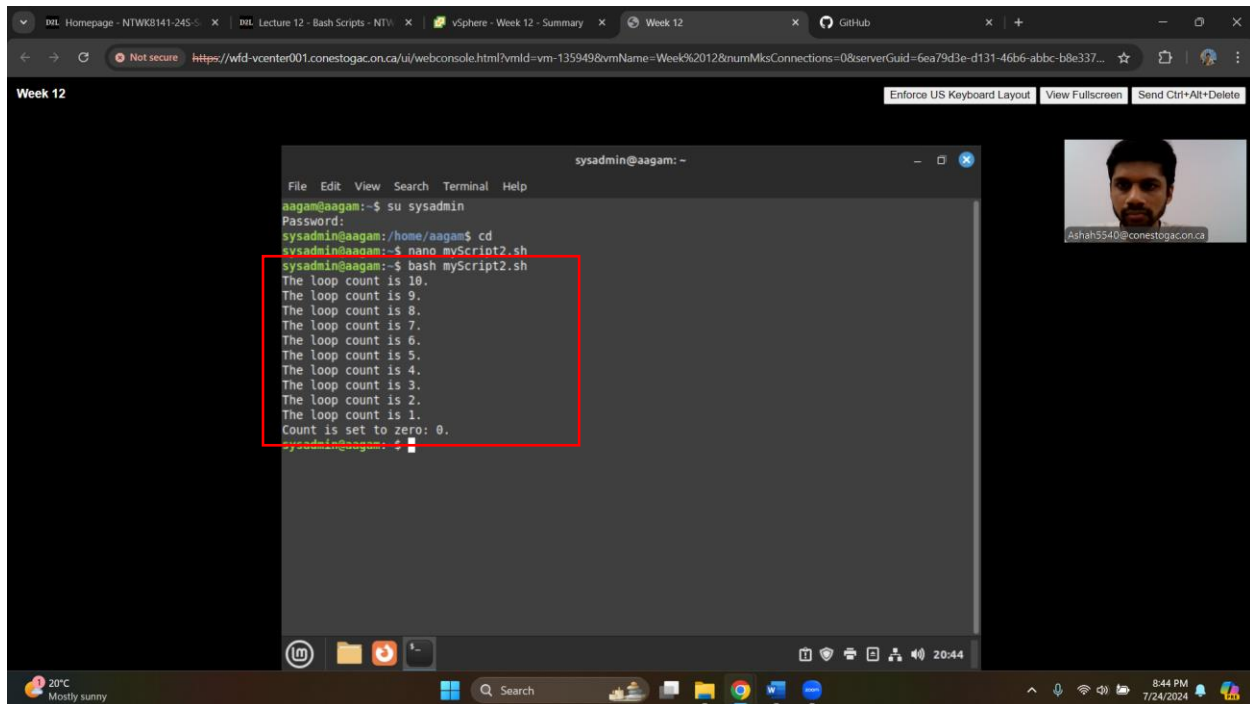


20. Exit the nano text editor by pressing Ctrl+X.





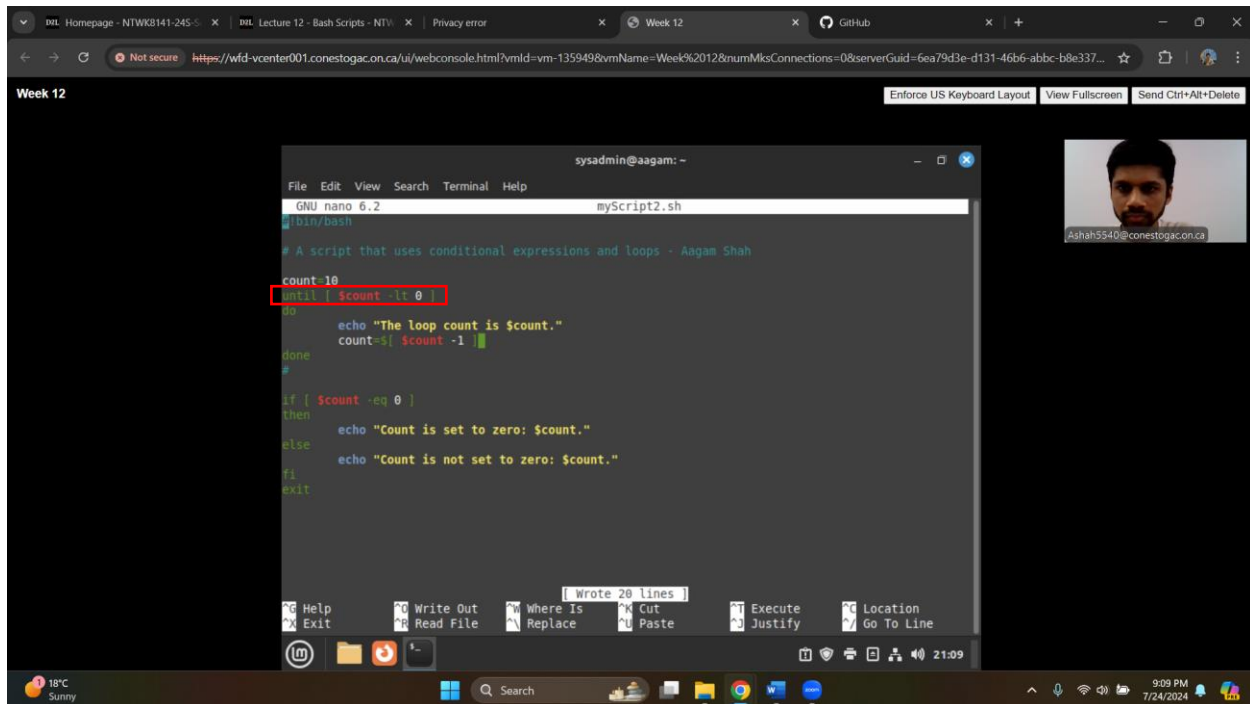
21. Try running the script by typing `bash myScript2.sh` and pressing Enter.



22. If the script did not run successfully, go back and edit the file, making any needed corrections.

➔ Script is running successfully. No need to change the file.

23. Once you have the script running correctly, try turning your while loop into an until loop, making changes to the count variable's original definition and math expression as needed.



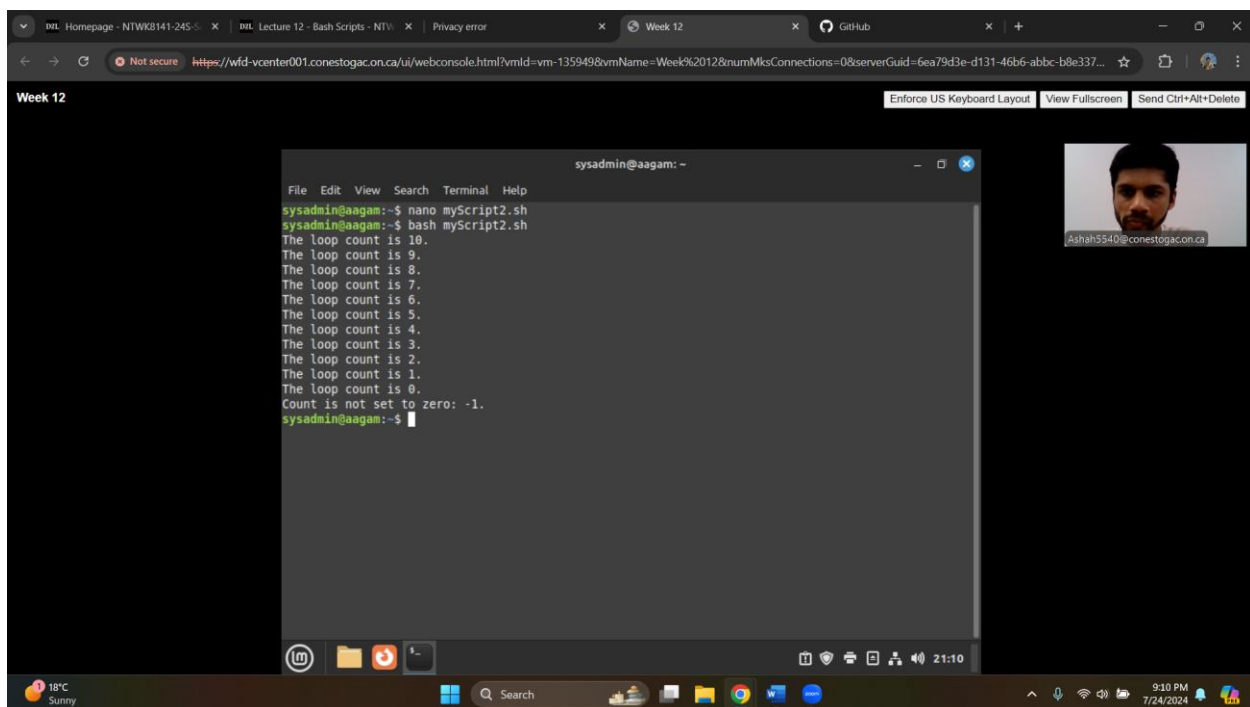
The screenshot shows a web browser window with a URL that includes 'Week12'. The main content area displays a terminal window titled 'sysadmin@aagam: ~'. Inside the terminal, a nano editor is open, editing a file named 'myScript2.sh'. The script content is as follows:

```
GNU nano 6.2 myScript2.sh
#bin/bash

# A script that uses conditional expressions and loops - Aagam Shah

count=10
until [ $count -lt 0 ]
do
    echo "The loop count is $count."
    count=$((count - 1))
done
#
if [ $count -eq 0 ]
then
    echo "Count is set to zero: $count."
else
    echo "Count is not set to zero: $count."
fi
exit
```

The terminal window has a status bar at the bottom indicating 'Wrote 20 lines'. The browser's address bar shows a 'Not secure' warning and a long URL. The browser's top bar shows 'Week 12' and 'GitHub' tabs. The browser's bottom bar shows 'Enforce US Keyboard Layout', 'View Fullscreen', and 'Send Ctrl+Alt+Delete' buttons. The browser's right sidebar shows a video feed of a person with the name 'Ashah5540@conestogac.on.ca'.

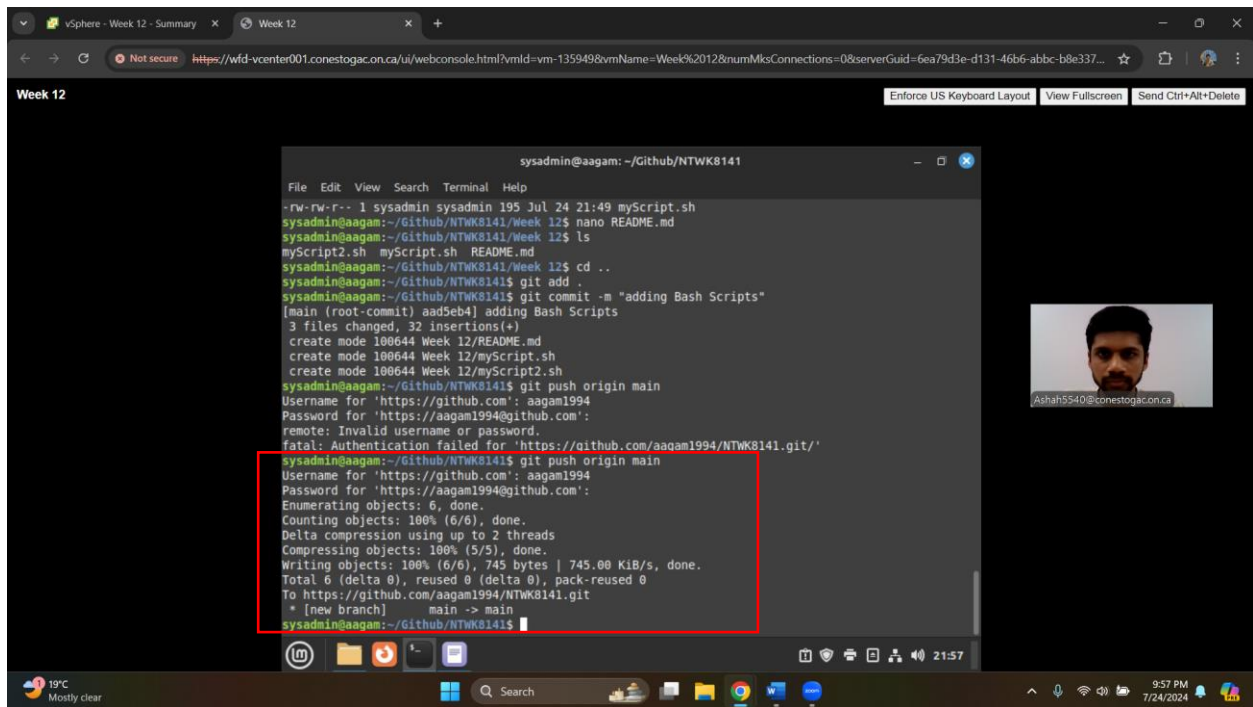


The screenshot shows the same web browser window as the previous one, but the terminal window now displays the output of the script. The output is as follows:

```
sysadmin@aagam: ~
sysadmin@aagam:~$ nano myScript2.sh
sysadmin@aagam:~$ bash myScript2.sh
The loop count is 10.
The loop count is 9.
The loop count is 8.
The loop count is 7.
The loop count is 6.
The loop count is 5.
The loop count is 4.
The loop count is 3.
The loop count is 2.
The loop count is 1.
The loop count is 0.
Count is not set to zero: -1.
sysadmin@aagam:~$
```

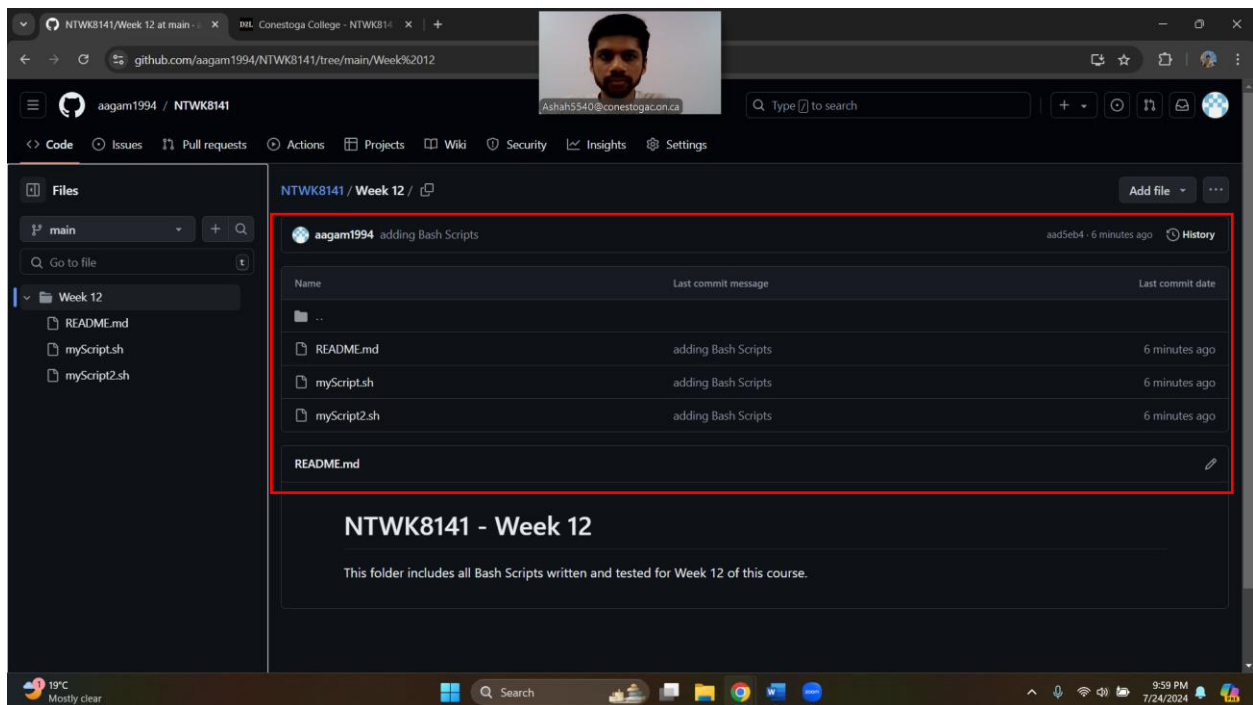
The terminal window now has a status bar at the bottom indicating '21:10'. The browser's address bar shows the same URL. The browser's top bar shows the same tabs. The browser's bottom bar shows the same buttons. The browser's right sidebar shows the same video feed.

GITHUB



```
sysadmin@aagam: ~/Github/NTWK8141
File Edit View Search Terminal Help
-rw-rw-r-- 1 sysadmin sysadmin 195 Jul 24 21:49 myScript.sh
sysadmin@aagam: ~/Github/NTWK8141/Week 12$ nano README.md
sysadmin@aagam: ~/Github/NTWK8141/Week 12$ ls
myScript2.sh myScript.sh README.md
sysadmin@aagam: ~/Github/NTWK8141/Week 12$ cd ..
sysadmin@aagam: ~/Github/NTWK8141$ git add .
sysadmin@aagam: ~/Github/NTWK8141$ git commit -m "adding Bash Scripts"
[main (root-commit) aad5eb4] adding Bash Scripts
3 files changed, 32 insertions(+)
create mode 100644 Week 12/README.md
create mode 100644 Week 12/myScript.sh
create mode 100644 Week 12/myScript2.sh
sysadmin@aagam: ~/Github/NTWK8141$ git push origin main
Username for 'https://github.com': aagam1994
Password for 'https://aagam1994@github.com':
remote: Invalid username or password.
fatal: Authentication failed for 'https://github.com/aagam1994/NTWK8141.git/'
sysadmin@aagam: ~/Github/NTWK8141$ git push origin main
Username for 'https://github.com': aagam1994
Password for 'https://aagam1994@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 2 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 745 bytes | 745.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/aagam1994/NTWK8141.git
 * [new branch] main -> main
sysadmin@aagam: ~/Github/NTWK8141$
```

Github Link: <https://github.com/aagam1994/NTWK8141/tree/main/Week%2012>



NTWK8141 / Week 12

Name	Last commit message	Last commit date
..		
README.md	adding Bash Scripts	6 minutes ago
myScript.sh	adding Bash Scripts	6 minutes ago
myScript2.sh	adding Bash Scripts	6 minutes ago
README.md		

NTWK8141 - Week 12

This folder includes all Bash Scripts written and tested for Week 12 of this course.