**OPS102 – Week 3 – File Systems - Sample Lab**

Student Name Aditya Mahesh Tambe

Student ID: 171969223

**Activity 1: File Globing**

When issuing Linux or Windows commands, it may be **more efficient** (less typing) to use **filename expansion symbols** also called **File Globing** to match files that share similar characteristics (e.g. same file extension) when issuing Linux commands.

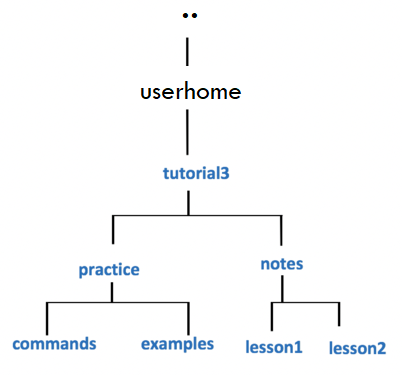
***Example****:* You can use a special character to indicate to the Bash shell to match all files that end with the extension ".txt" in your current working directory:

**ls \*.txt**  
a.txt b.txt c.txt 1.txt 2.txt 3.txt abc.txt work.txt

Below are the most common Filename Expansion symbols and how they are used for filename expansion:

|  |  |
| --- | --- |
| **Filename Expansion Symbol** | **Purpose** |
| **\*** | Asterisk (\*) to represent **0 or more characters** |
| **?** | Question mark (?) to represent **exactly one character (any character)** |
| **[ ]** | Square brackets ([ ]) to represent and match for the  **character enclosed within the square brackets**. It represents ONLY ONE character - it's like a **Question Mark (?)** but with **conditions or restrictions.** |
| **[! ]** | Square brackets containing an exclamation mark immediately after the open square bracket ([! ]) to represent and match and **OPPOSITE** character for the character enclosed within the square brackets. |

Consider following file hierarchy for the activities in this section. This applies to both of Linux and Windows.

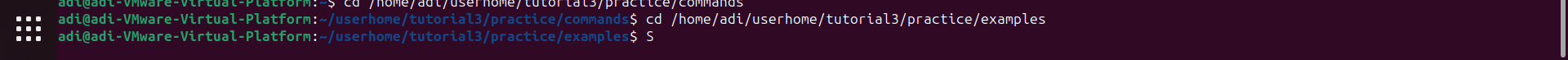


You will now get practice issuing file management commands using **filename expansion symbols**. We will be using the directory structure given above.

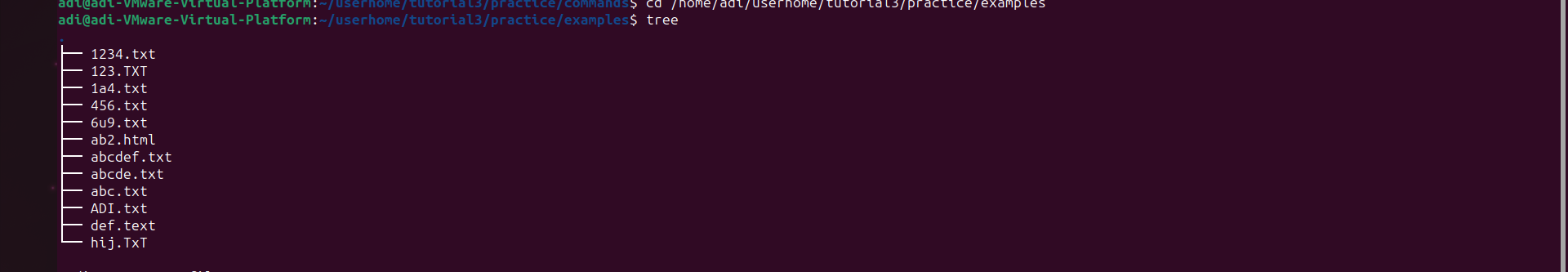
A great way to practice filename expansion, use the **touch** command on Linux to create a lot of empty filenames (for windows use any preferred way to create such files.), write the **ls/dir** commands that use **filename expansion**, predict the filenames that will be display, and finally run the command to check your work.

**Perform the following steps for Linux and repeat them for windows using equivalent commands learnt previously:**

1. Issue a Linux command to move to the **examples** directory  
   (i.e. under *practice* directory as shown in diagram to the right).

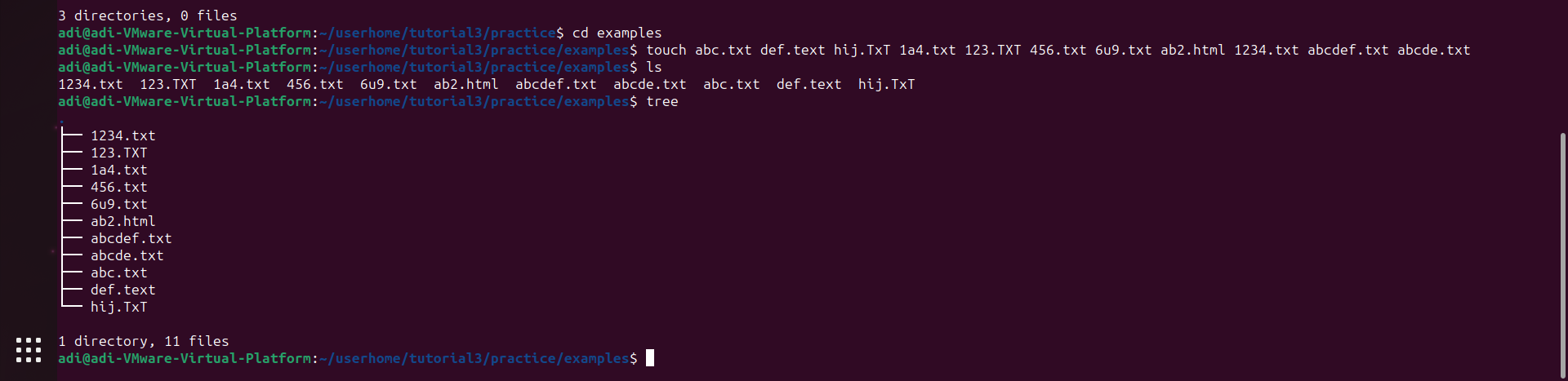


1. Issue a Linux command to confirmed that you have moved to the **examples** directory.



1. Issue the **touch** command to create the following empty text files in the *examples* directory:  
   (note *upper* and *lowercase* letters)

**abc.txt  
def.text  
hij.TxT  
1a4.txt  
123.TXT  
456.txt  
6u9.txt  
ab2.html  
1234.txt  
abcdef.txt  
abcde.txt**

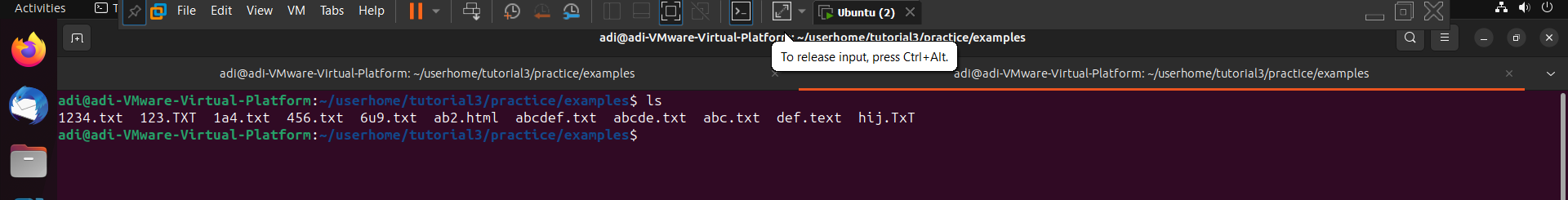
****

1. If you encounter errors, then make corrections (eg. **viewing directory contents**, **check for correct filename syntax**, **case sensitivity**, **missing files**, **files in the wrong location**, etc.)

no error was seen in the linux ubuntu as afterwards in the listing scenario, it shows the files not sure if it will work the same as the extensions I have provided with the questions.

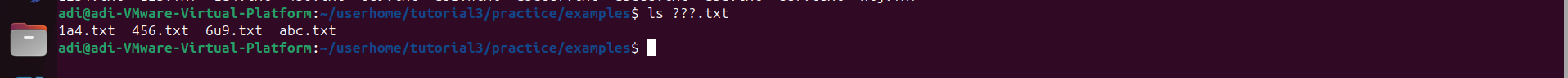
1. Issue the **ls** command to get a listing of files in your *examples* directory.  
     
   The output should look identical to the diagram displayed below.  
   You can refer to this listing to see all files so you can then predict the output from Linux commands that use filename expansion symbols.

[Listing-1.png](https://wiki.cdot.senecacollege.ca/wiki/File:Listing-1.png)



1. What do you think the output will be from the following Linux command?  
   **ls ???.txt**  
   **Write down the expected output** on paper, then **issue the command** to check your answer.

ls command helps us to list the directories, folders, sub-folders, files present in a particular directory or root folder.



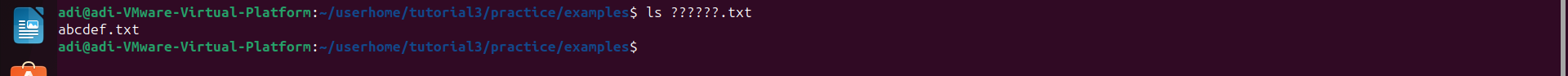
In this very case it is listing the .txt file only which as three characters before the .txt as we gave a command to do so.

1. What do you think the output will be from the following Linux command?  
   **ls ?????.txt**  
   **Write down the expected output** on paper, then **issue the command** to check your answer.



In this case the .txt file which has file character before it is needed to listed not the others.

1. What do you think the output will be from the following Linux command?  
   **ls ??????.txt**  
   **Write down the expected output** on paper, then **issue the command** to check your answer.



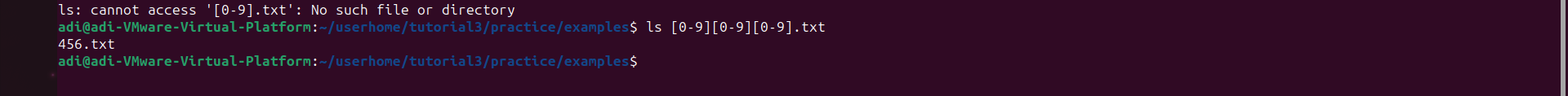
In this it is listing a file which has 6 characters before the .txt extension.

1. What do you think the output will be from the following Linux command?  
   **ls [0-9].txt**  
   **Write down the expected output** on paper, then **issue the command** to check your answer.br>Did the command work?  
   What does this teach you about the character class [ ] symbol?

It doesn’t work, I expected that it will list the files in one to nine order, but it didn’t.

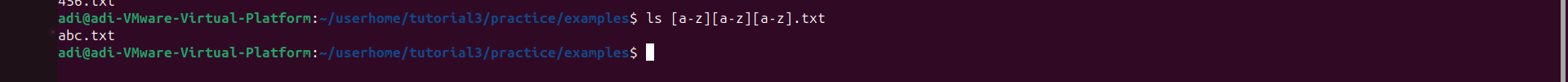


1. What do you think the output will be from the following Linux command?  
   **ls [0-9][0-9][0-9].txt**  
   **Write down the expected output** on paper, then **issue the command** to check your answer.



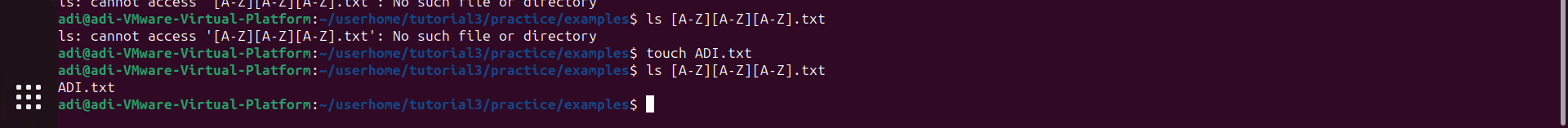
Yeah it did work, it listed 456.txt.

1. What do you think the output will be from the following Linux command?  
   **ls [a-z][a-z][a-z].txt**  
   **Write down the expected output** on paper, then **issue the command** to check your answer.



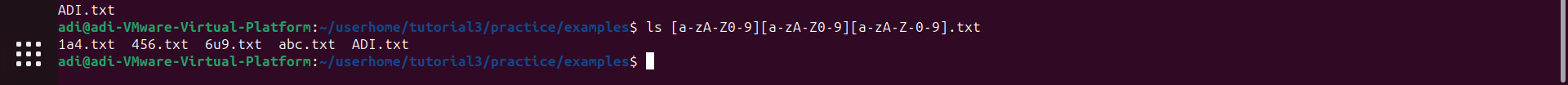
It did listed the lowercase file name which consists a-z characters.

1. What do you think the output will be from the following Linux command (using character class with UPPERCASE letters)?:  
   **ls [A-Z][A-Z][A-Z].txt**  
   **Write down the expected output** on paper, then **issue the command** to check your answer.



Firstly, it did not listed anything as no UPPERCASE character file was present, but then I did created a file which had UPPERCASE character and it did listed it.

1. What do you think the output will be from the following Linux command (using character class using alpha-numeric characters)?  
   **ls [a-zA-Z0-9][a-zA-Z0-9][a-zA-Z0-9].txt**  
   **Write down the expected output** on paper, then **issue the command** to check your answer.



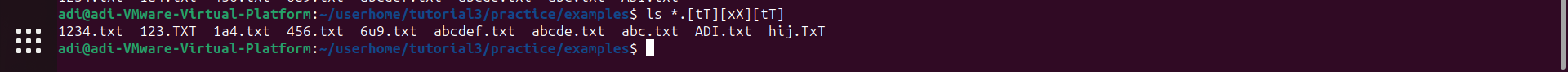
It did mention the files which had numbers, characters & mixture of it so that was listed.

1. What do you think the output will be from the following Linux command?  
   **ls \*.txt**  
   **Write down the expected output** on paper, then **issue the command** to check your answer. Did ALL text files get listed? Why not?



Yes all the files get listed and all of them were txt only.

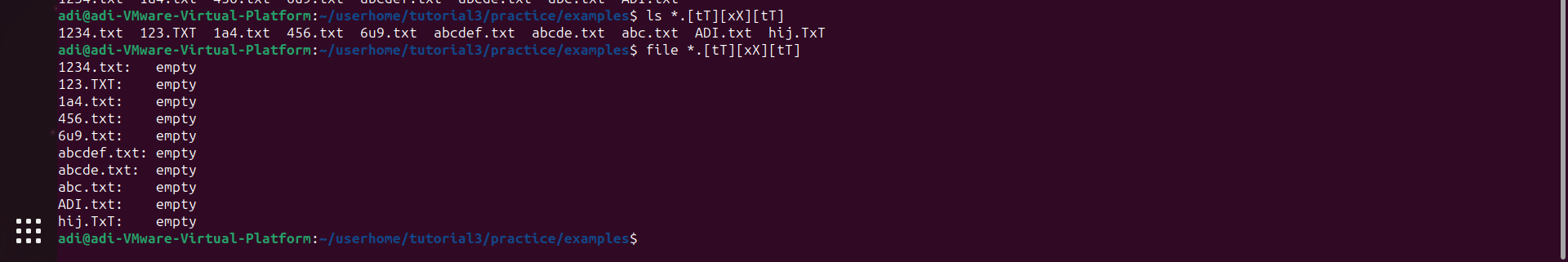
1. What do you think the output will be from the following Linux command?  
   **ls \*.[tT][xX][tT]**  
   **Write down the expected output** on paper, then **issue the command** to check your answer. Did ALL text files get listed this time? If so, why?



Yes it did, listed because after the dot we asked to list those file which have .txt or .TXT so the ones which are listed have them .

**NOTE:** We have just been using filename expansion symbols just with the ls command.  
Filename expansion symbols can be used for ANY Linux file management command (e.g. **cat**, **more**, **less**, **cp**, **mv**, **rm**, **ls**, etc.).  
Let's get some practice issuing these other Linux file management commands.

1. Issue the following Linux command: **file \*.[tT][xX][tT]**  
   What is the purpose of this command? Which files are contained in this output?

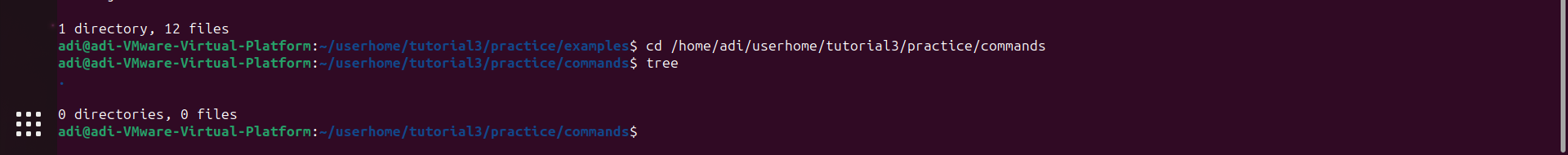


Right now, the files doesn’t contain any data in it since we have just created and not inserted any data in it. But this helps us to know if the file has data in it or not if not it will display empty.

17.Change to the **commands** directory using an **absolute** pathname  
 (use the diagram on right-side for reference).



1. Issue a Linux command to confirm that you are now in the **commands** directory.



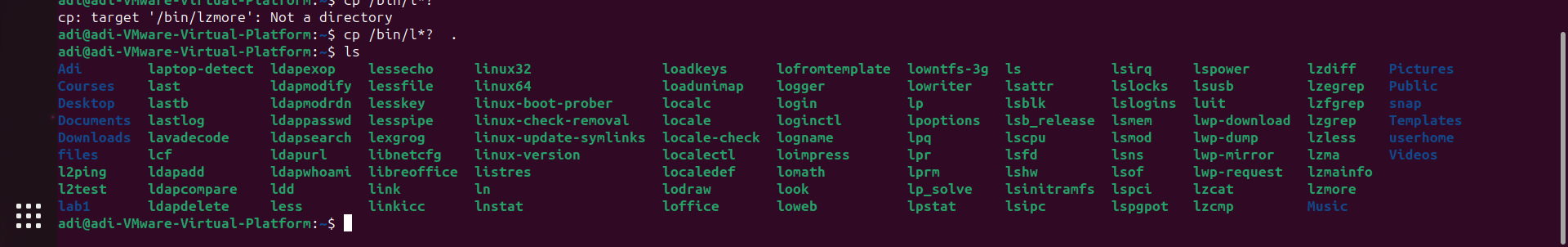
1. Issue the following Linux command (lowercase "l" NOT the number "1"):  
   **cp /bin/l\*   .**  
   View the contents of the contents directory. What did this command do?
2. Issue the following Linux command: **rm \***  
     
   View the contents of the contents directory. What did this command do?



Basically it says that it cannot remove listed files. And rm on the other hand we use it for removing file or directory.

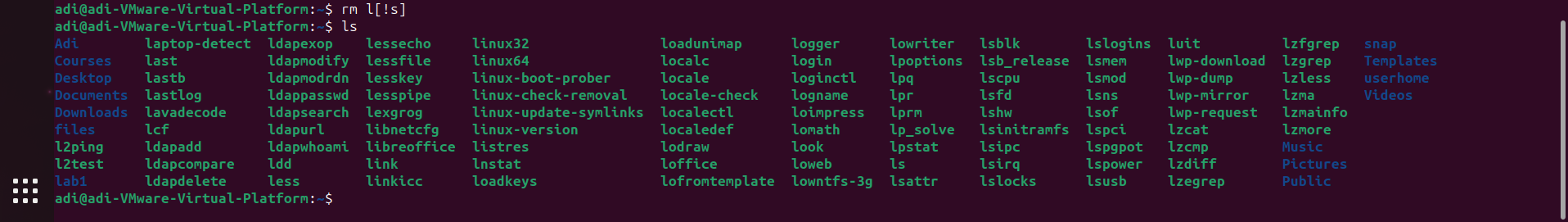
1. Issue the following Linux command (lowercase "l" NOT the number "1"):  
   **cp /bin/l?   .**  
   View the contents of the contents directory. What did this command do?



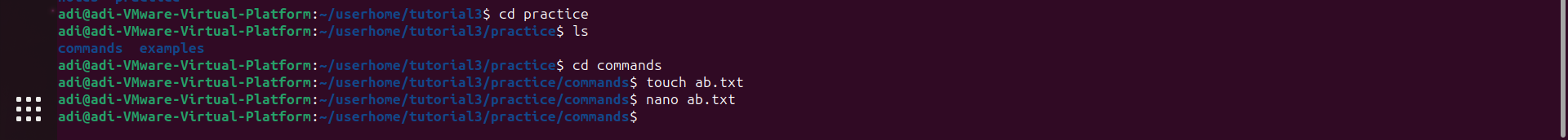


Basically it did take the command but don’t know what it did.

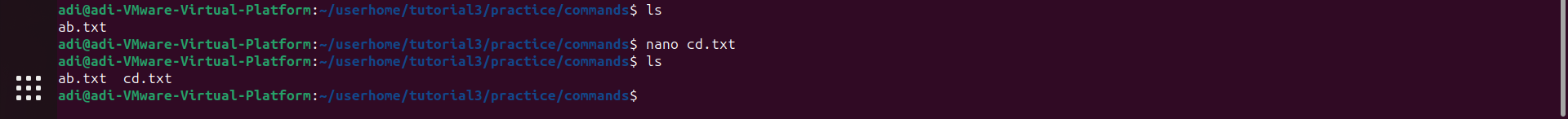
1. Issue the following Linux command: **rm l[!s]**  
   View the contents of the contents directory. What did this command do?



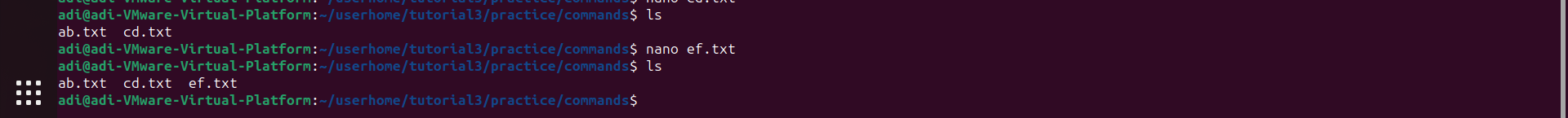
1. Use a text editor (nano or vi) to create the file called **ab** in the **commands** directory that contains the line of text below,  
   and then save editing changes to this file:  
   This is file ab



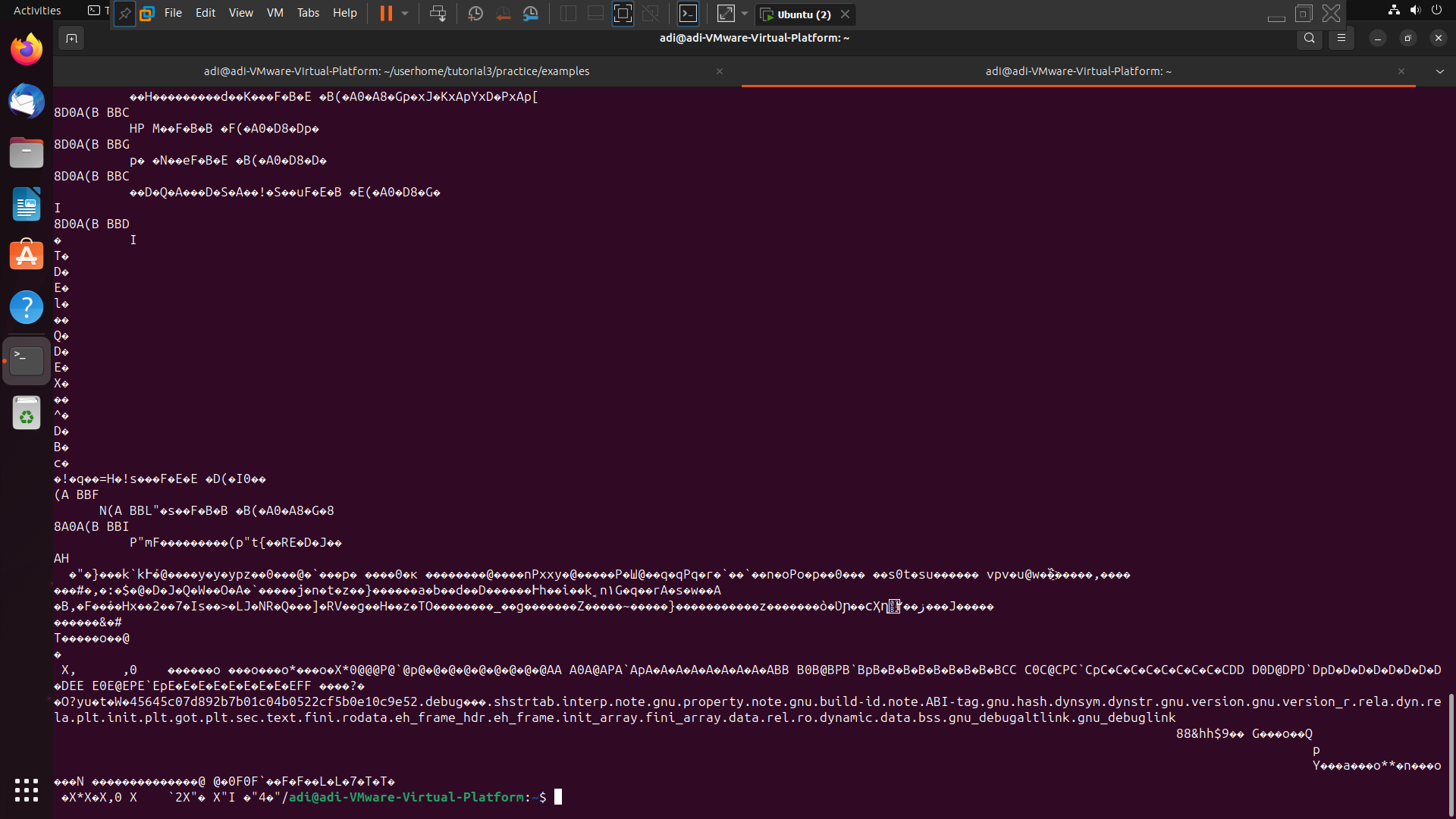
1. Use a text editor (nano or vi) to create the file called **cd** in the **commands** directory that contains the line of text below,  
   and then save editing changes to this file:  
   This is file cd



1. Use a text editor (nano or vi) to create the file called **ef** in the **commands** directory that contains the line of text below,  
   and then save editing changes to this file:  
   This is file ef

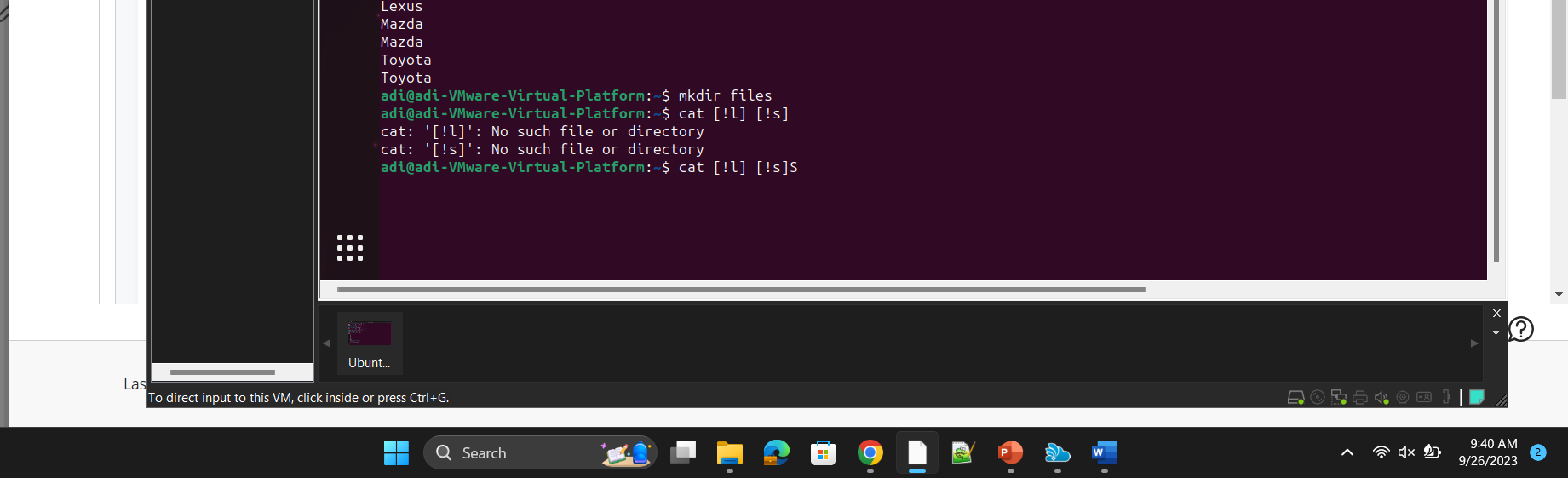


1. Issue the following Linux command: **cat ??**  
     
   View the contents of the contents directory. What did this command do? Why does the output look strange?



The cat command basically helps to view the content within the file and displays it on the terminal.  
  
**NOTE:** Press the keys **ctrl-c** to return to the shell prompt.

1. Issue the following Linux command: **cat [!l][!s]**  
     
   View the contents of the contents directory. What did this command do? Does the output look better? If so, why?



Basically this will display the directories or files which have l and s characters in it.