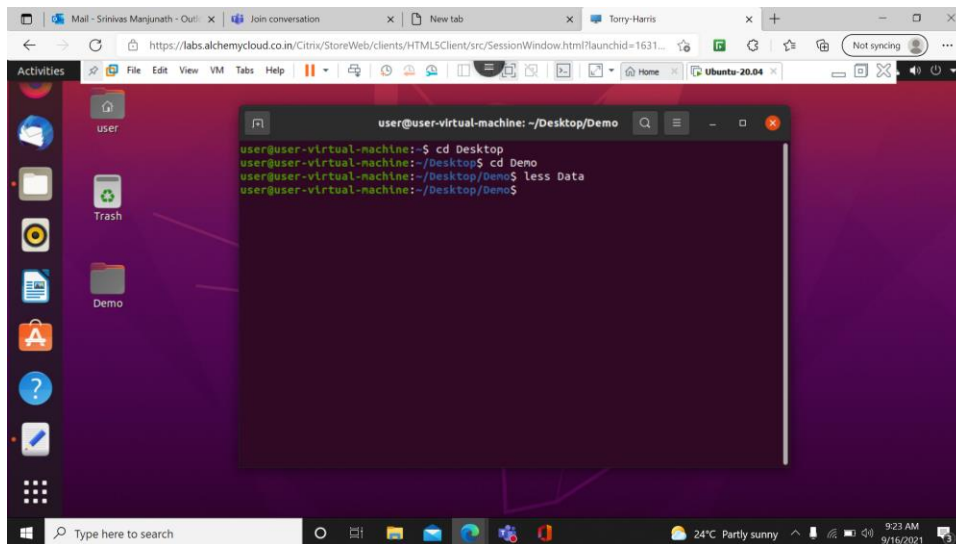
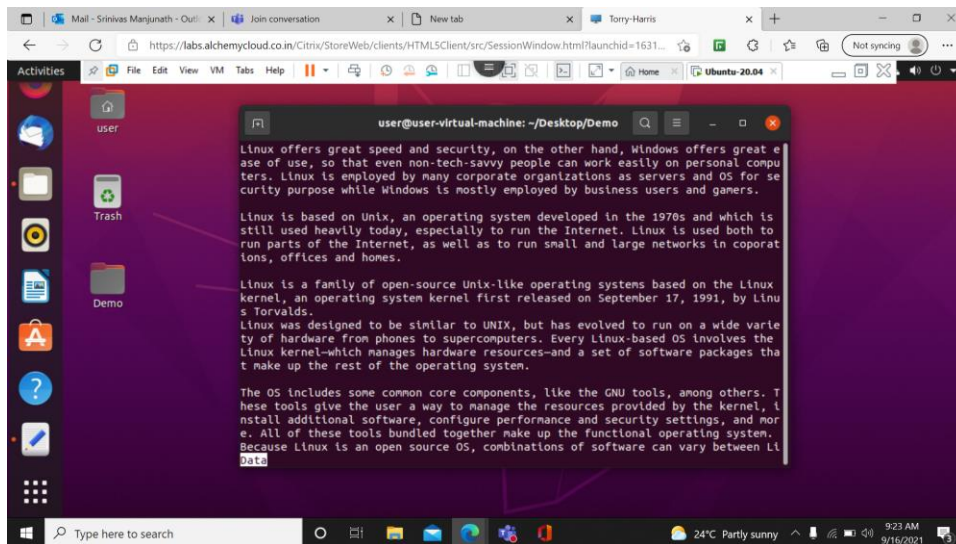


Day 04 Assignment 04

1>Less: Less command is a Linux utility that can be used to read the contents of a text file one page(one screen) at a time. It has faster access because if file is large it doesn't access the complete file, but accesses it page by page.



```
user@user-virtual-machine: ~/Desktop/Demo
user@user-virtual-machine:~$ cd Desktop
user@user-virtual-machine:~/Desktop$ cd Demo
user@user-virtual-machine:~/Desktop/Demo$ less Data
user@user-virtual-machine:~/Desktop/Demo$
```



```
user@user-virtual-machine: ~/Desktop/Demo
Linux offers great speed and security, on the other hand, Windows offers great ease of use, so that even non-tech-savvy people can work easily on personal computers. Linux is employed by many corporate organizations as servers and OS for security purpose while Windows is mostly employed by business users and gamers.

Linux is based on Unix, an operating system developed in the 1970s and which is still used heavily today, especially to run the Internet. Linux is used both to run parts of the Internet, as well as to run small and large networks in corporations, offices and homes.

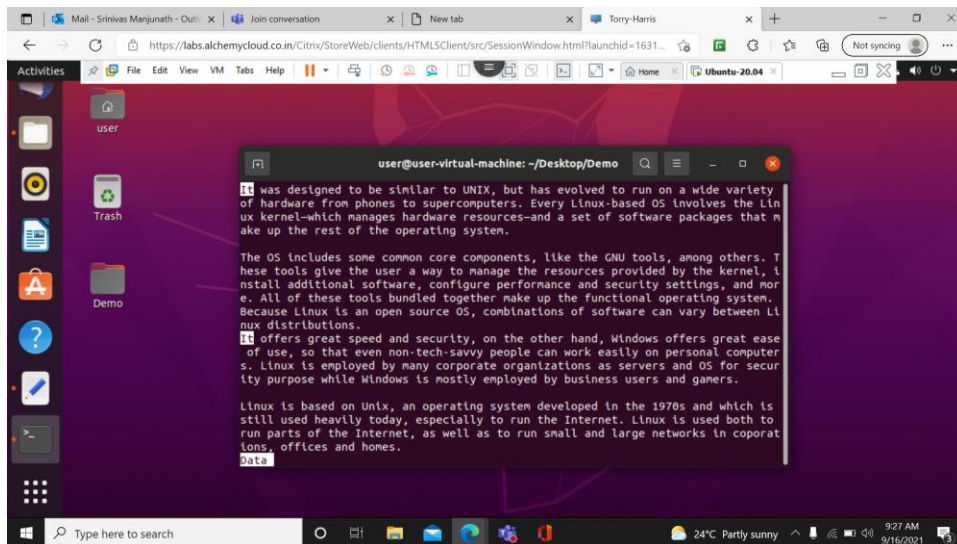
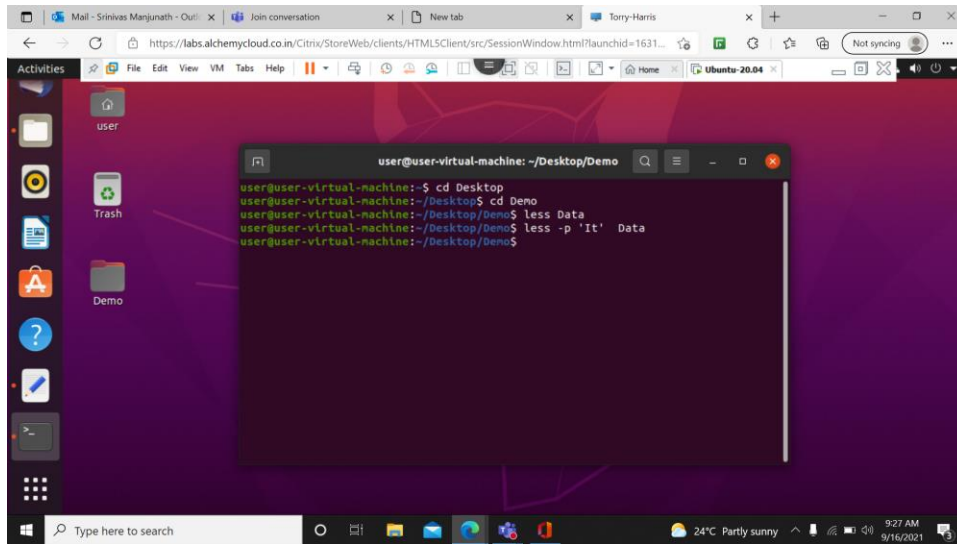
Linux is a family of open-source Unix-like operating systems based on the Linux kernel, an operating system kernel first released on September 17, 1991, by Linus Torvalds.

Linux was designed to be similar to UNIX, but has evolved to run on a wide variety of hardware from phones to supercomputers. Every Linux-based OS involves the Linux kernel-which manages hardware resources-and a set of software packages that make up the rest of the operating system.

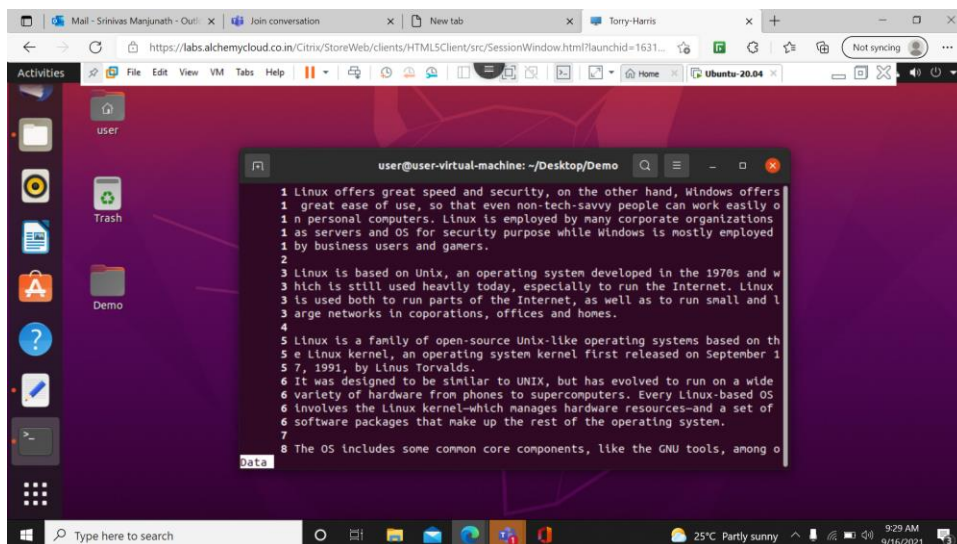
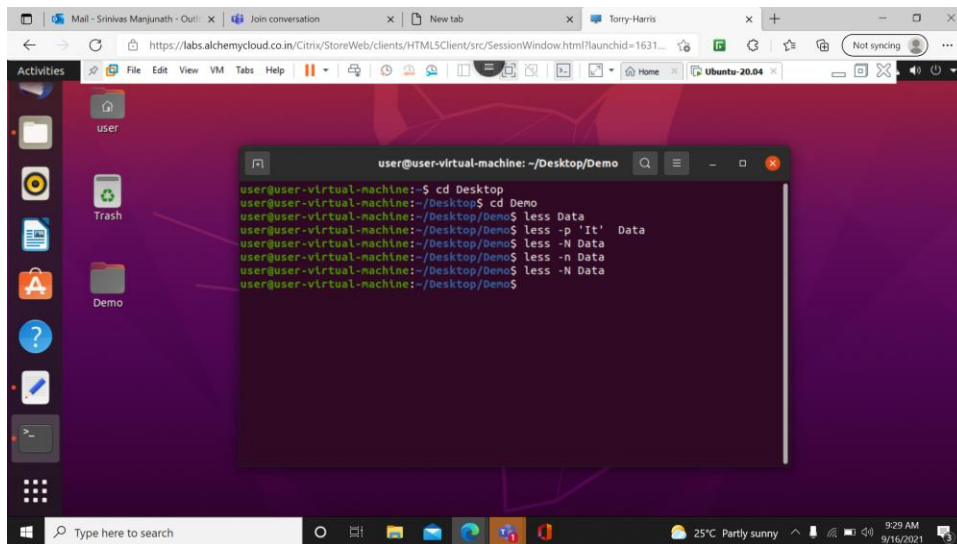
The OS includes some common core components, like the GNU tools, among others. These tools give the user a way to manage the resources provided by the kernel, to install additional software, configure performance and security settings, and more. All of these tools bundled together make up the functional operating system. Because Linux is an open source OS, combinations of software can vary between Linux distributions.
Data
```

2> less -p: pattern it tells less to start at the first occurrence of pattern in the file.

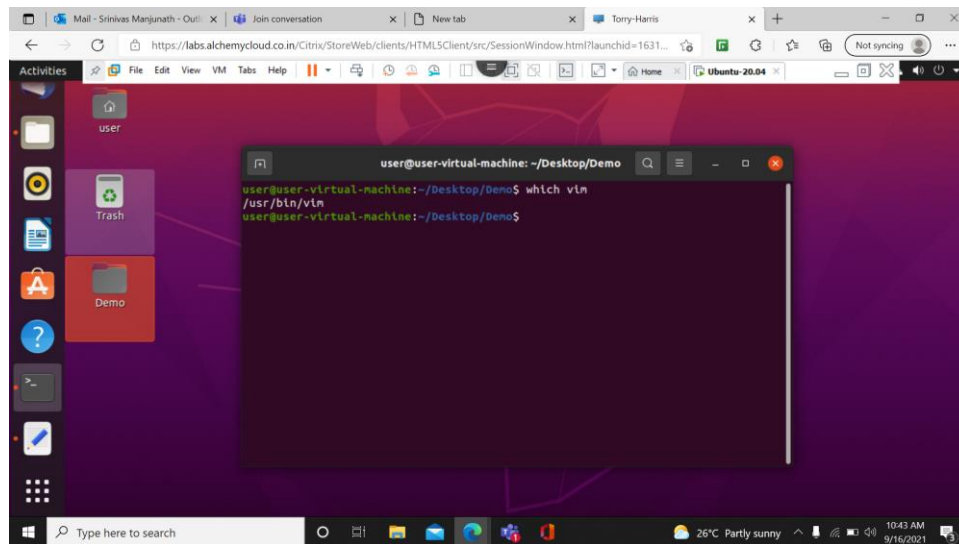
Note: press enter for next line , Space bar for next paragraph, p for previous page and q for quit.



3> less -N: Display the line number.

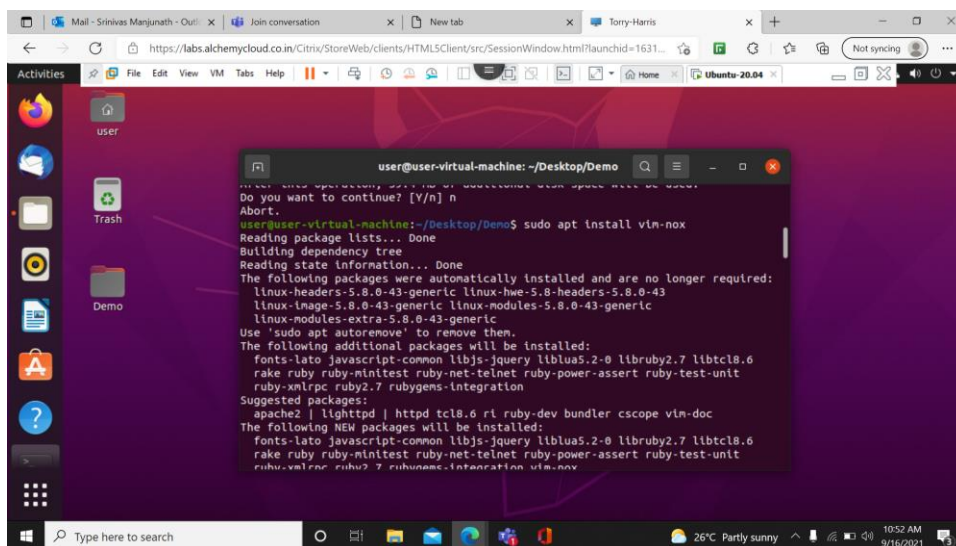


4> which: Used to locate the executable file associated with the given command by searching it in the path environment variable.

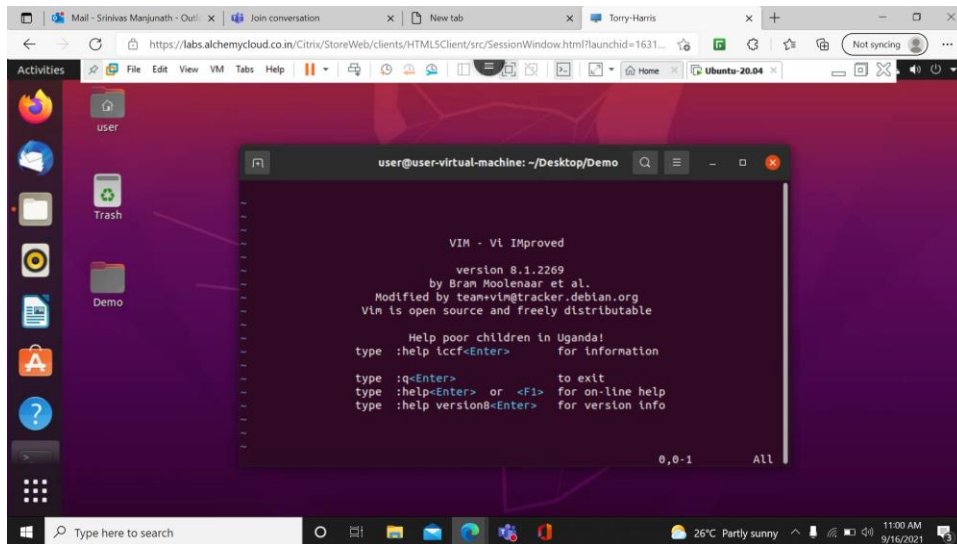
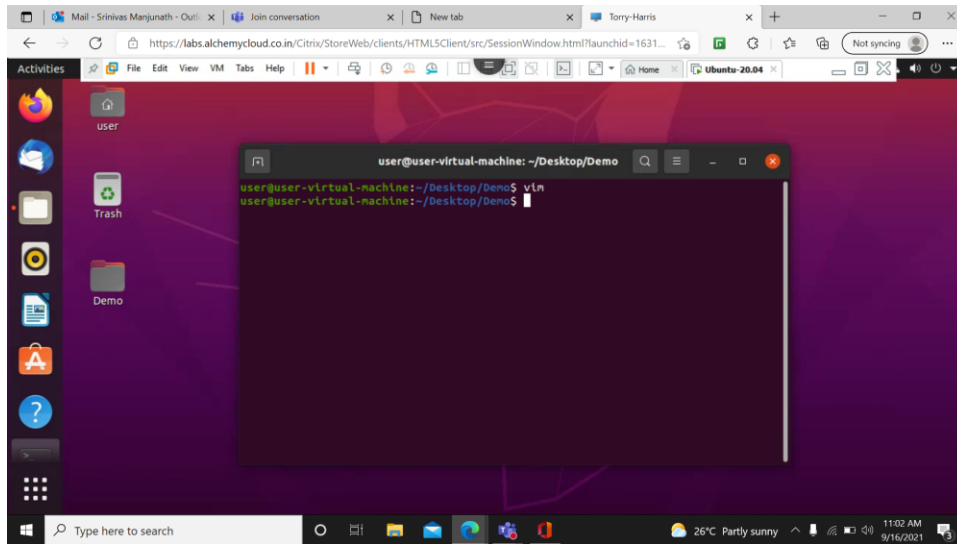


5> Install vim: `$sudo apt-get install vim` or `$sudo apt install vim-nox`

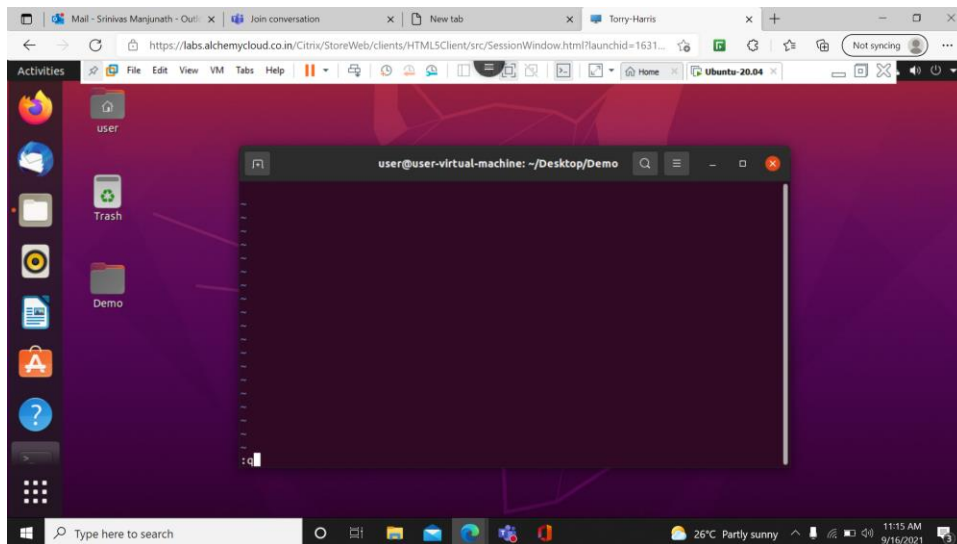
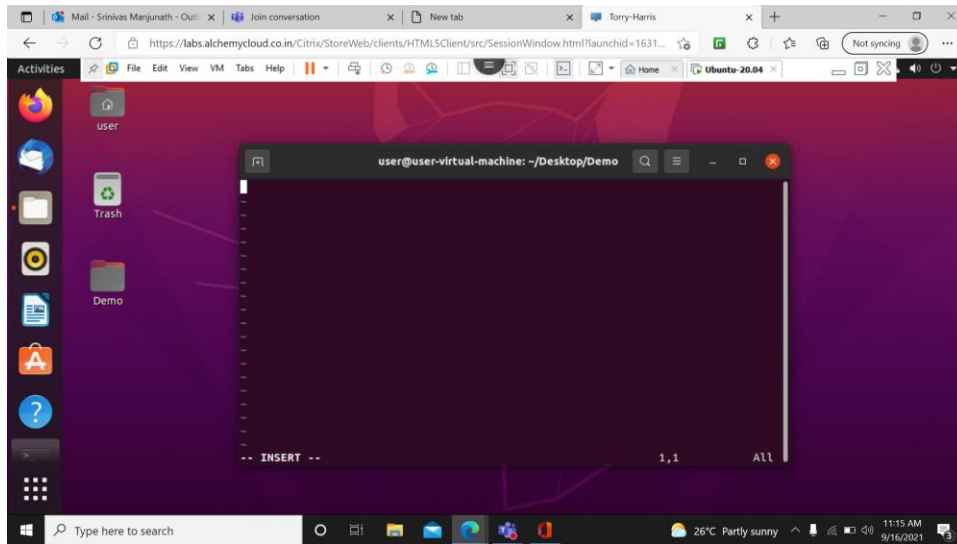
Vim is an advanced and highly configurable text editor built to enable efficient text editing. Vim text editor is developed by Bram Moolenaar. It supports most file types and vim editor is also known as a programmer's editor. We can use with its plugin based on our needs.



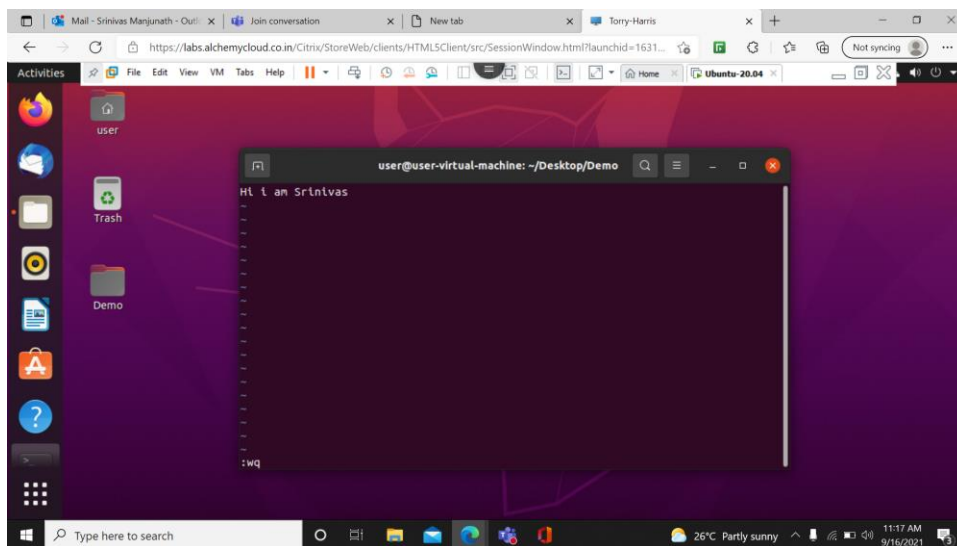
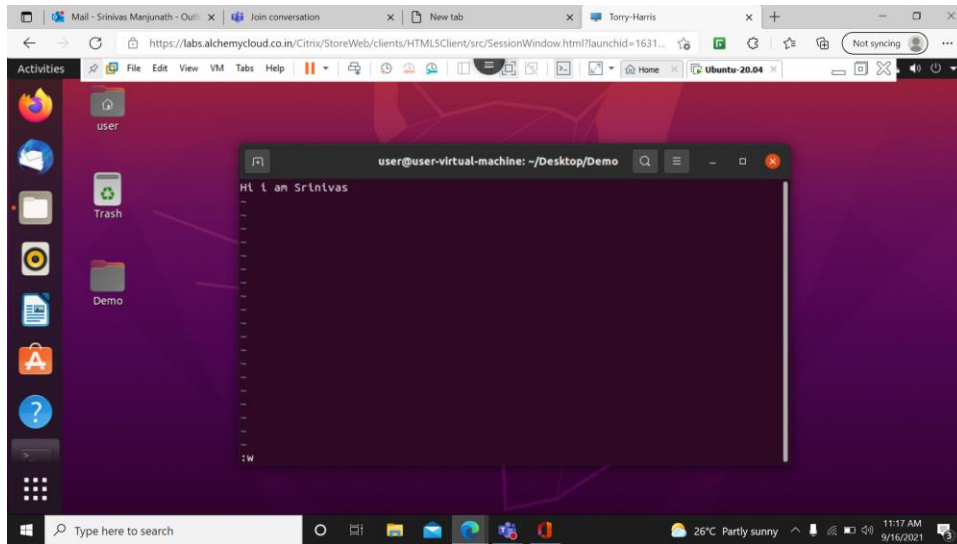
6> Vim: is a free and open-source, screen-based text editor program for Unix.



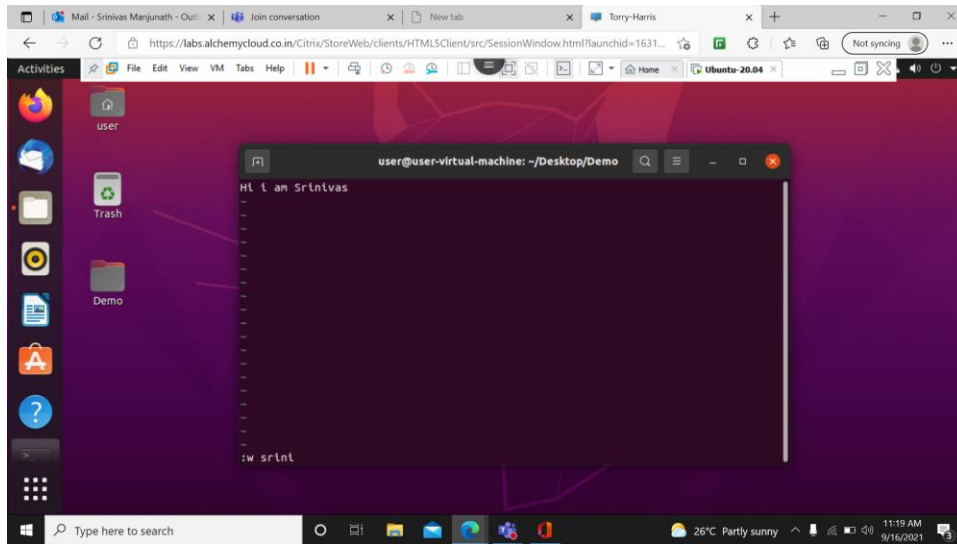
7> write data we need to go in insert mode. To go into write mode type i. And also we can type A or a or I for inserting. To exit press esc and type command :q.



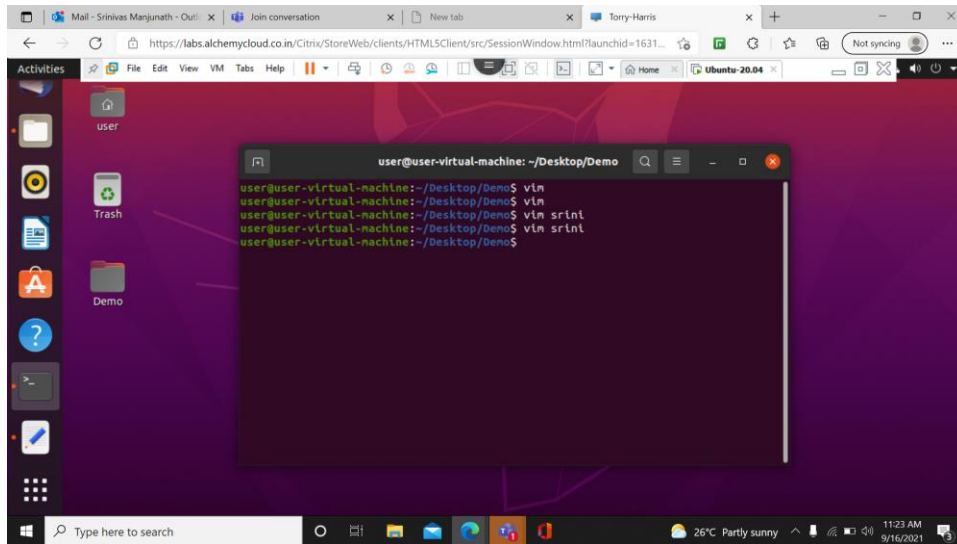
8>To save the file use :w or :wq to save and exit.



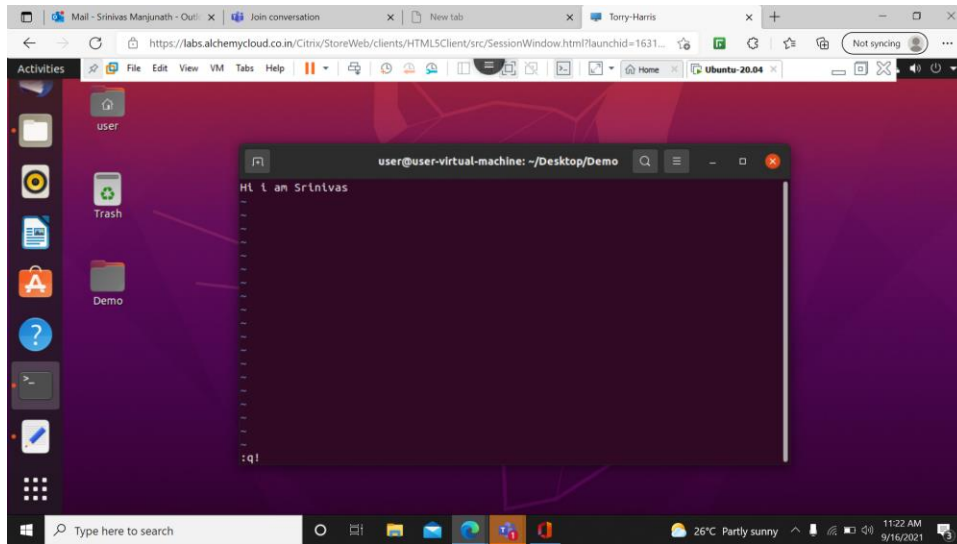
9>To save the file with file name use :w filename.



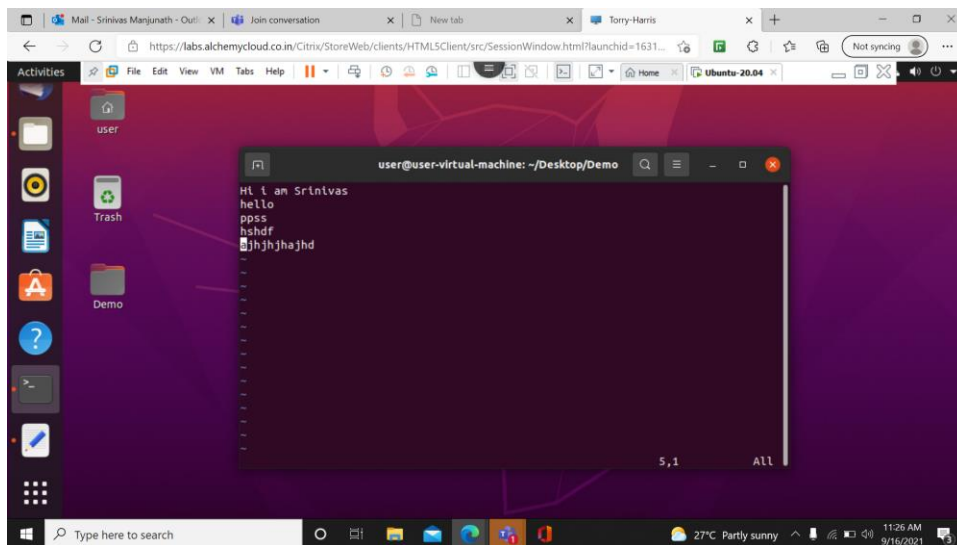
10> To open the existing file \$vim filename.



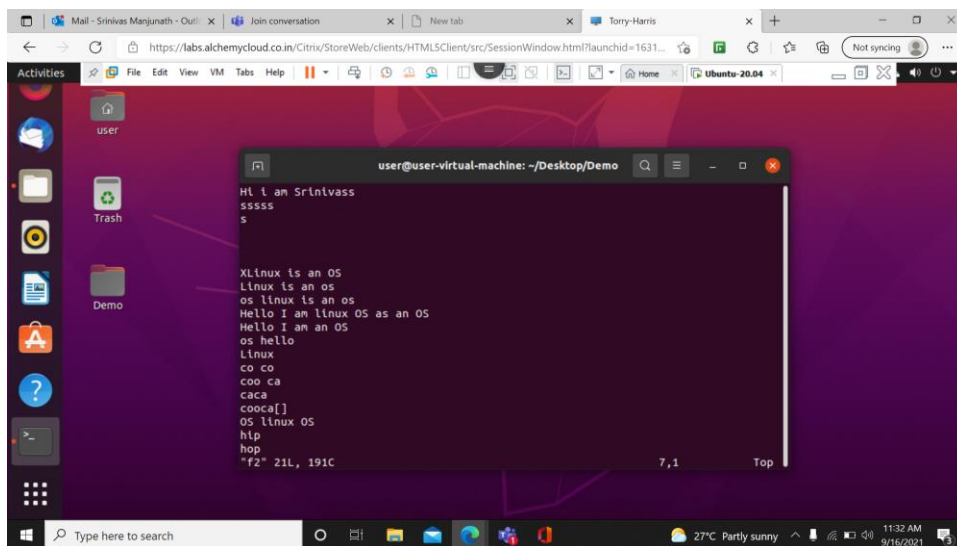
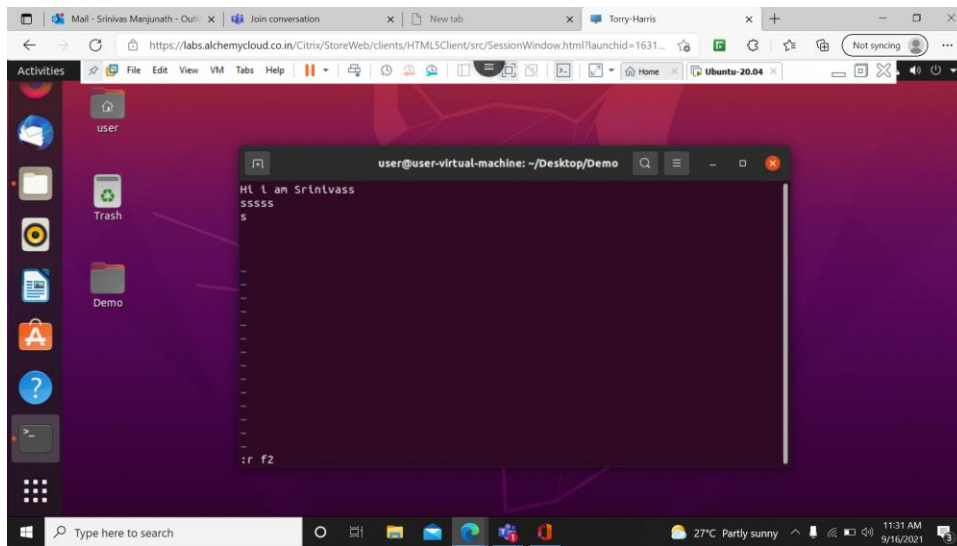
11> to quite the file without saving it :q!



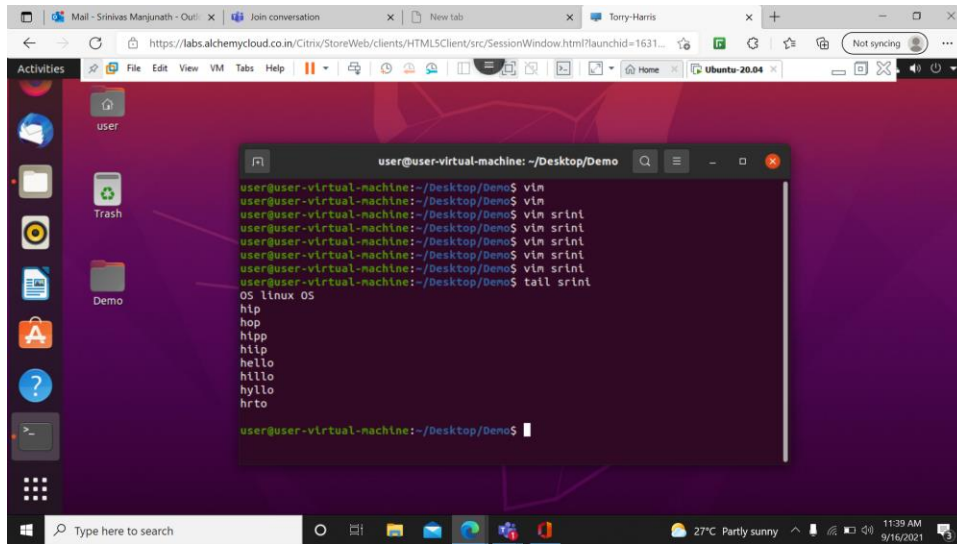
12> Delete commands: we exit from insert mode and press d for line-by-line deletion, u for undo, x for word by word.



13> to read the existing file in vim :r filename



14> tail: The tail command, as the name implies, print the last N number of data of the given input. By default it prints the last 10 lines of the specified files.

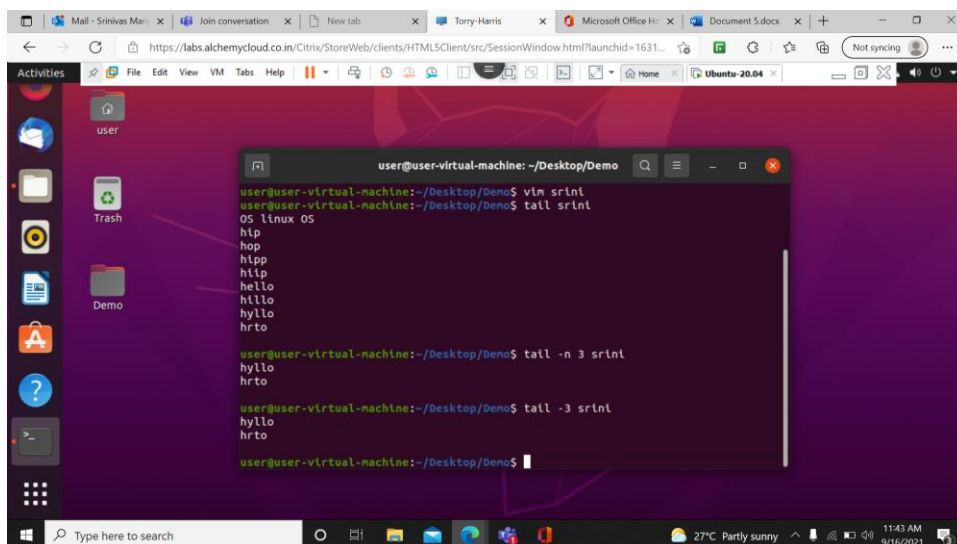


The screenshot shows a virtual machine desktop with a purple background. A terminal window titled 'user@user-virtual-machine: ~/Desktop/Demo' is open. The terminal shows the following commands and output:

```
user@user-virtual-machine:~/Desktop/Demo$ vim
user@user-virtual-machine:~/Desktop/Demo$ vim
user@user-virtual-machine:~/Desktop/Demo$ vim srint
user@user-virtual-machine:~/Desktop/Demo$ vim srint
user@user-virtual-machine:~/Desktop/Demo$ vim srint
user@user-virtual-machine:~/Desktop/Demo$ vim srint
user@user-virtual-machine:~/Desktop/Demo$ vim srint
user@user-virtual-machine:~/Desktop/Demo$ tail srint
05 linux 05
hip
hop
hipp
hiip
hello
hylo
hylo
hrto

user@user-virtual-machine:~/Desktop/Demo$
```

15> tail -n: Prints the last 'num' lines instead of last 10 lines. num is mandatory to be specified in command otherwise it displays an error. **\$ tail -n 3 filename** or **\$ tail -3 filename**.



The screenshot shows the same virtual machine desktop. The terminal window now shows the following commands and output:

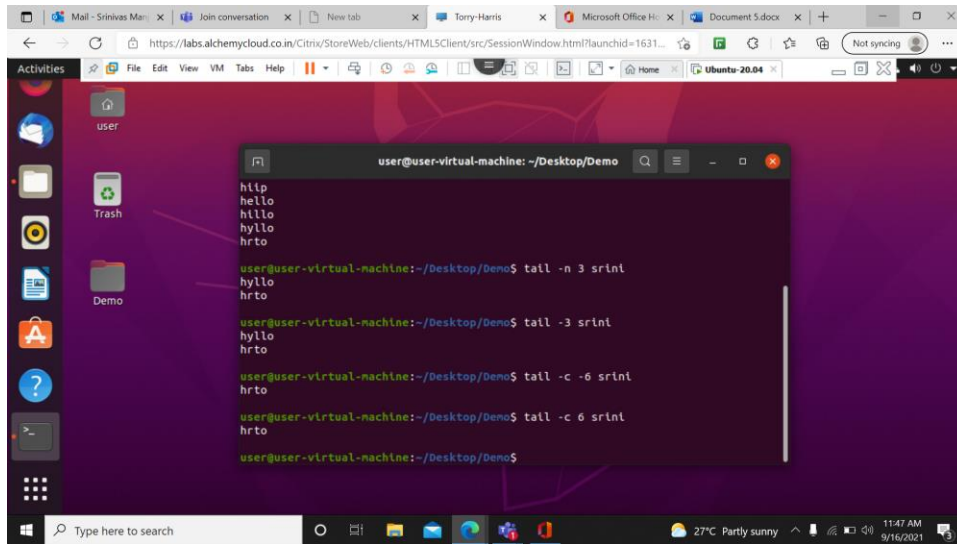
```
user@user-virtual-machine:~/Desktop/Demo$ vim srint
user@user-virtual-machine:~/Desktop/Demo$ tail srint
05 linux 05
hip
hop
hipp
hiip
hello
hylo
hylo
hrto

user@user-virtual-machine:~/Desktop/Demo$ tail -n 3 srint
hylo
hrto

user@user-virtual-machine:~/Desktop/Demo$ tail -3 srint
hylo
hrto

user@user-virtual-machine:~/Desktop/Demo$
```

16> tail -c : Prints the last 'num' bytes from the file specified. Newline count as a single character, so if tail prints out a newline, it will count it as a byte. **\$ tail -c -6 filename** or **\$ tail -c 6 filename**.



```
user@user-virtual-machine: ~/Desktop/Demo
http
hello
hillo
hylllo
hrto

user@user-virtual-machine:~/Desktop/Demo$ tail -n 3 srini
hylllo
hrto

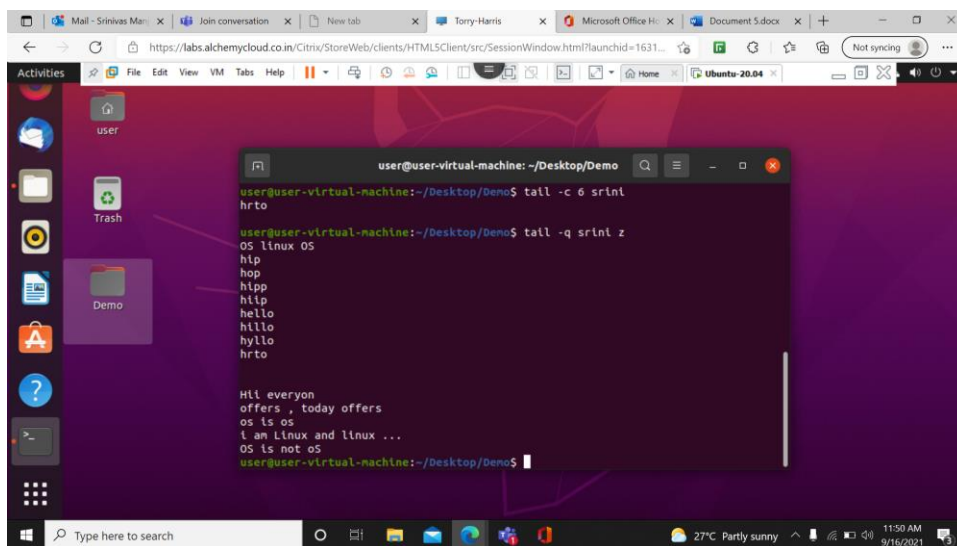
user@user-virtual-machine:~/Desktop/Demo$ tail -3 srini
hylllo
hrto

user@user-virtual-machine:~/Desktop/Demo$ tail -c -6 srini
hrto

user@user-virtual-machine:~/Desktop/Demo$ tail -c 6 srini
hrto

user@user-virtual-machine:~/Desktop/Demo$
```

17> **tail -q**: It is used if more than 1 file is given. Because of this command, data from each file is not precedes by its file name.

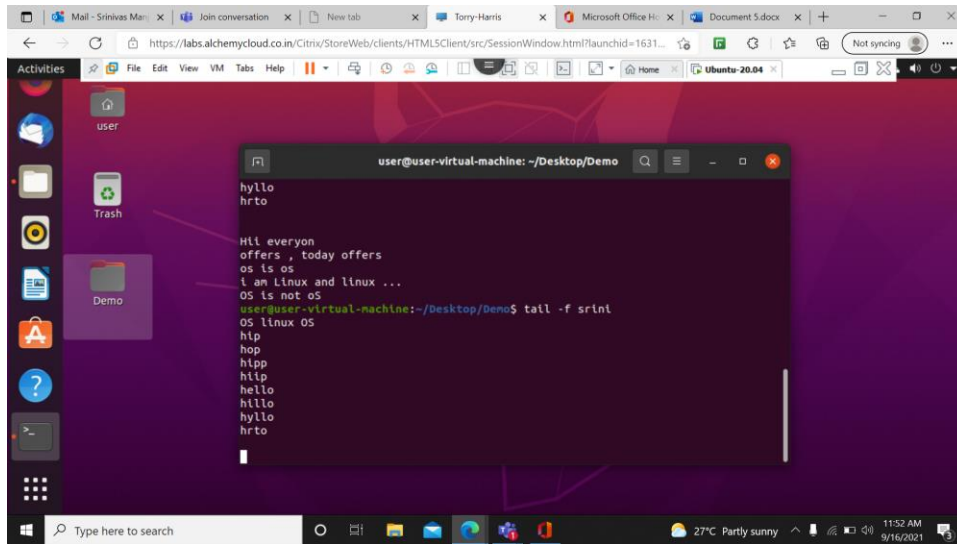


```
user@user-virtual-machine:~/Desktop/Demo$ tail -c 6 srini
hrto

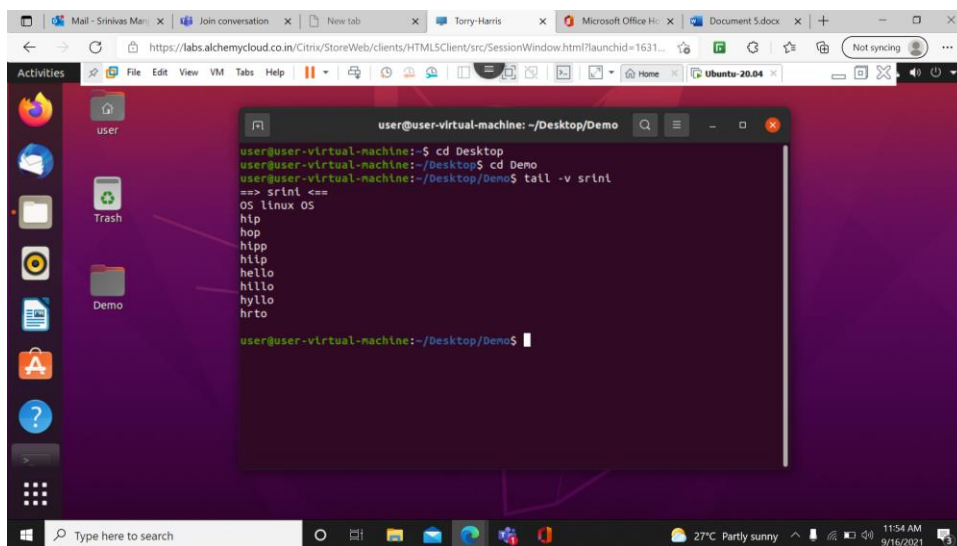
user@user-virtual-machine:~/Desktop/Demo$ tail -q srini z
OS linux OS
hip
hop
hipp
htip
hello
hillo
hylllo
hrto

Hii everyon
offers , today offers
os is os
i am linux and linux ...
OS is not os
user@user-virtual-machine:~/Desktop/Demo$
```

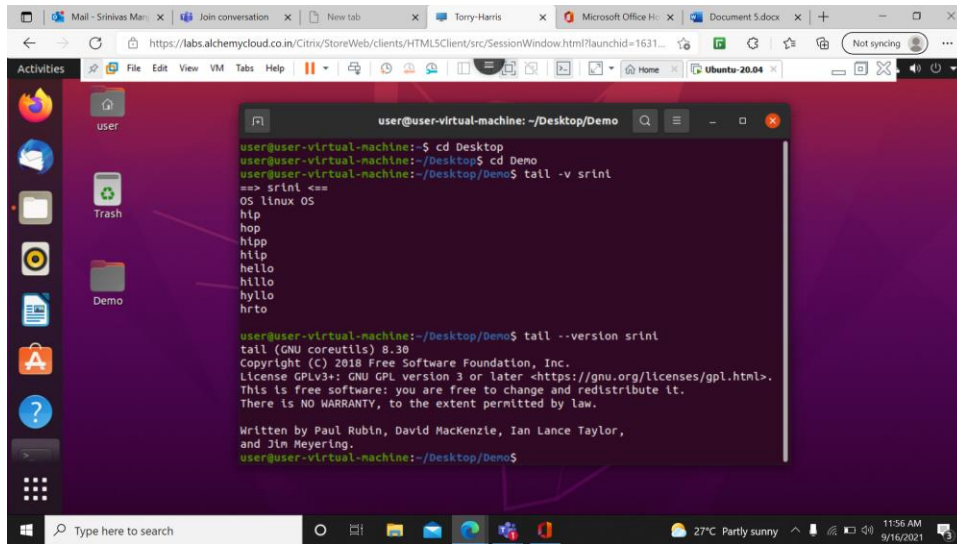
18> **tail -f**: This option is mainly used by system administration to monitor the growth of the log files written by many Unix program as they are running.



19> tail -v : -v: By using this option, data from the specified file is always preceded by its file name.



20> tail- -version: This option is used to display the version of tail which is currently running on your system.



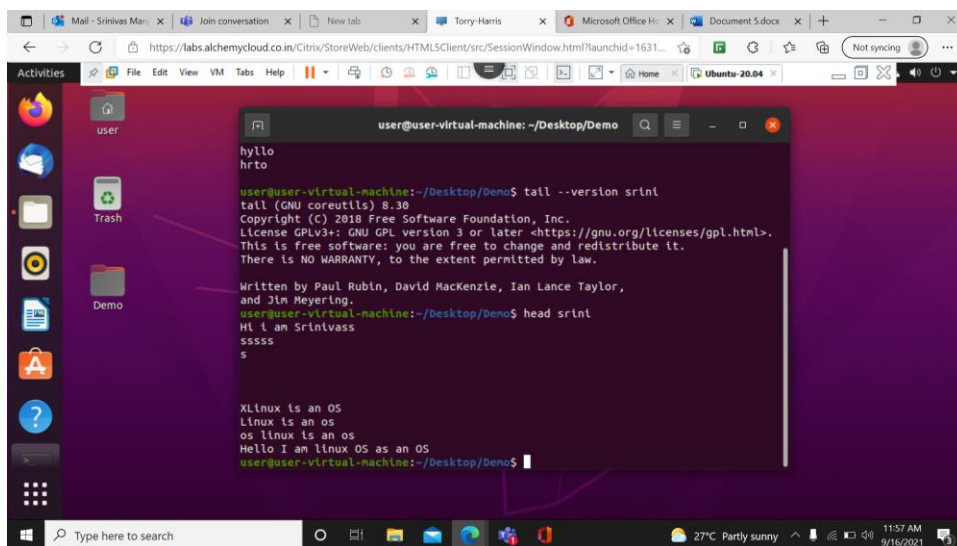
The screenshot shows a virtual machine desktop with a purple background. A terminal window is open, displaying the output of the `tail -v srini` command. The output lists the contents of the `srini` file, which includes the OS name, a list of names, the GNU coreutils version, copyright information, and the GNU GPL license text. The terminal prompt is `user@user-virtual-machine: ~/Desktop/Demo`.

```
user@user-virtual-machine: ~/Desktop/Demo
user@user-virtual-machine:~$ cd Desktop
user@user-virtual-machine:~/Desktop$ cd Demo
user@user-virtual-machine:~/Desktop/Demo$ tail -v srini
==> srini <==
OS linux OS
hlp
hop
hipp
hllp
hello
hillo
hylo
hrto

user@user-virtual-machine:~/Desktop/Demo$ tail --version srini
tail (GNU coreutils) 8.30
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by Paul Rubin, David MacKenzie, Ian Lance Taylor,
and Jim Meyering.
user@user-virtual-machine:~/Desktop/Demo$
```

21> head- The head command, as the name implies, print the top N number of data of the given input. By default, it prints the first 10 lines of the specified files.



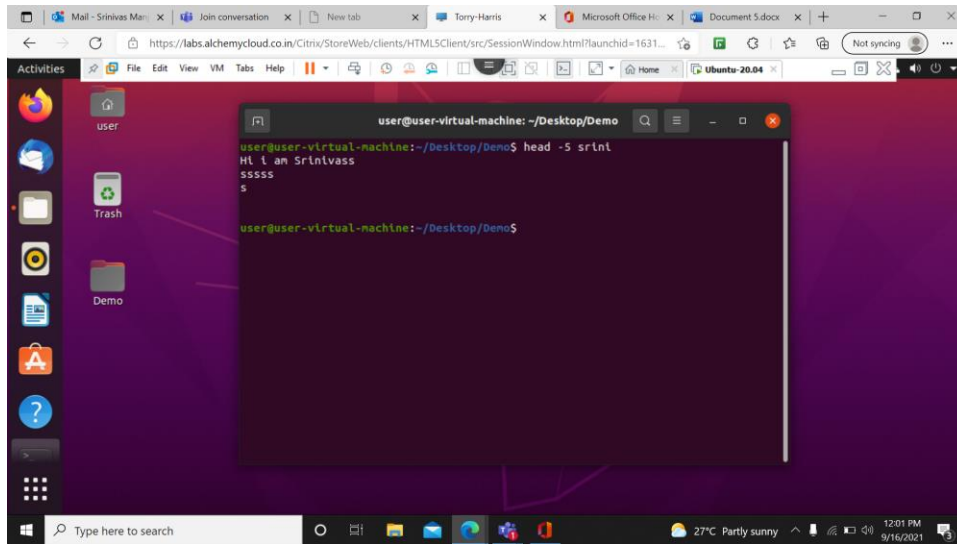
The screenshot shows the same virtual machine desktop. The terminal window now displays the output of the `head srini` command. The output shows the first 10 lines of the `srini` file, which are the last two lines of the previous output, followed by the GNU coreutils version, copyright information, and the GNU GPL license text. The terminal prompt is `user@user-virtual-machine: ~/Desktop/Demo`.

```
user@user-virtual-machine: ~/Desktop/Demo
hylo
hrto

user@user-virtual-machine:~/Desktop/Demo$ head srini
HI I am Srinivass
sssss
s

XLinux is an OS
Linux is an os
os linux is an os
Hello I am linux OS as an OS
user@user-virtual-machine:~/Desktop/Demo$
```

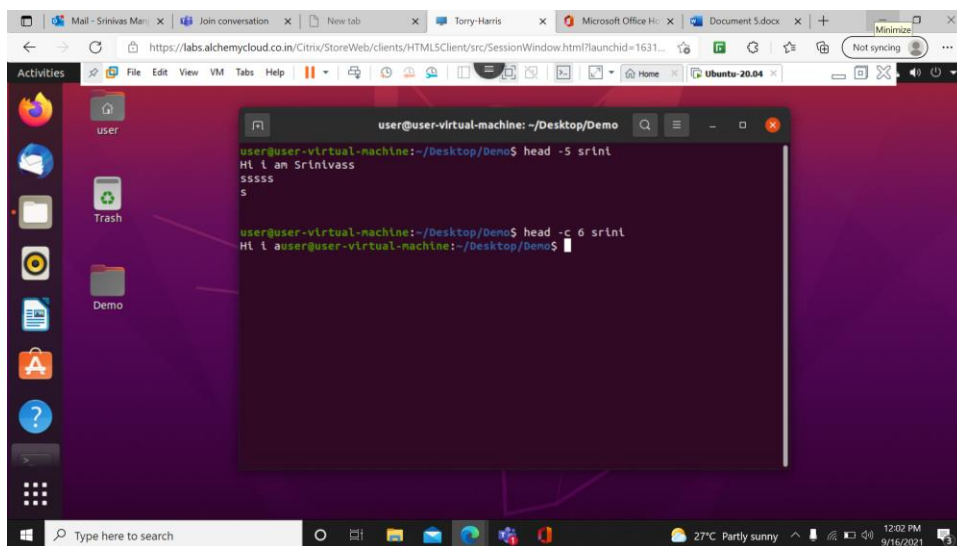
22> head -n : Prints the first 'num' lines instead of first 10 lines. **num** is mandatory to be specified in command otherwise it displays an error.



The screenshot shows a virtual machine desktop with a purple background. A terminal window titled 'user@user-virtual-machine: ~/Desktop/Demo' is open. The terminal shows the command `head -5 srintl` being executed, resulting in the output: `Hi I am Srinivass`, `sssss`, and `s`. The desktop has icons for 'user', 'Trash', and 'Demo'. The system tray at the bottom shows the date and time as 12:01 PM on 9/16/2021.

```
user@user-virtual-machine: ~/Desktop/Demo
user@user-virtual-machine:~/Desktop/Demo$ head -5 srintl
Hi I am Srinivass
sssss
s
user@user-virtual-machine:~/Desktop/Demo$
```

23> **head -c** : Prints the first 'num' bytes from the file specified. Newline count as a single character, so if head prints out a newline, it will count it as a byte.



The screenshot shows the same virtual machine desktop. The terminal window now shows the command `head -c 6 srintl` being executed, resulting in the output: `Hi I`. The desktop and system tray are the same as in the previous screenshot.

```
user@user-virtual-machine: ~/Desktop/Demo
user@user-virtual-machine:~/Desktop/Demo$ head -c 6 srintl
Hi I
user@user-virtual-machine:~/Desktop/Demo$
```

24> **head -q**: It is used if more than 1 file is given. Because of this command, data from each file is not preceded by its file name.

```
user@user-virtual-machine: ~/Desktop/Demo
Hi I am Srinivass
sssss
s

user@user-virtual-machine:~/Desktop/Demo$ head -c 6 srint
Hi I auser@user-virtual-machine:~/Desktop/Demo$ head -q srint z
Hi I am Srinivass
sssss
s

Xlinux is an OS
Linux is an os
os linux is an os
Hello I am linux OS as an OS

Hll everyon
offers , today offers
os ls os
i am linux and linux ...
OS ls not os
user@user-virtual-machine:~/Desktop/Demo$
```

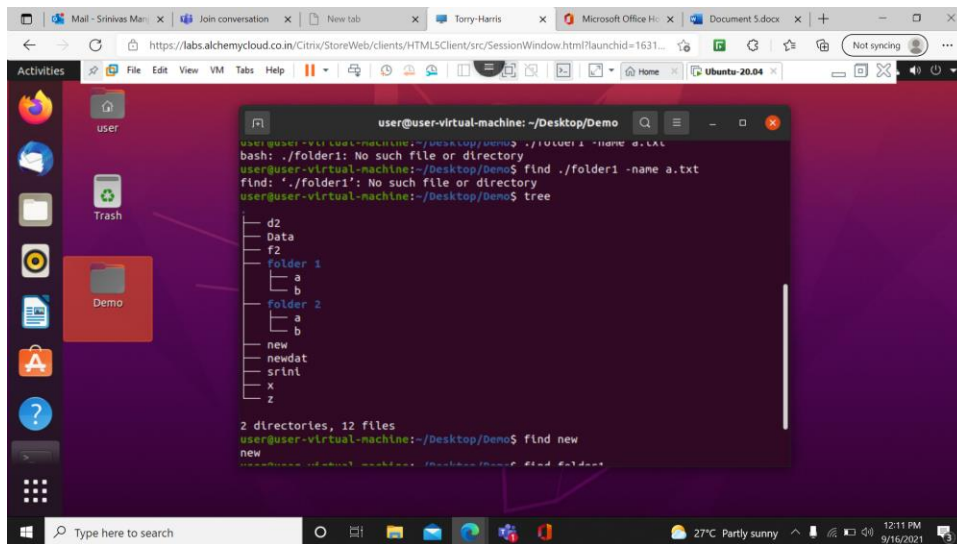
25> **head-v:** By using this option, data from the specified file is always preceded by its file name.

```
user@user-virtual-machine: ~/Desktop/Demo
Xlinux is an OS
Linux is an os
os linux is an os
Hello I am linux OS as an OS

Hll everyon
offers , today offers
os ls os
i am linux and linux ...
OS ls not os
user@user-virtual-machine:~/Desktop/Demo$ head -v srint
==> srint <==
Hi I am Srinivass
sssss
s

Xlinux is an OS
Linux is an os
os linux is an os
Hello I am linux OS as an OS
user@user-virtual-machine:~/Desktop/Demo$
```

26> **find:** It can be used to find files and directories and perform subsequent operations on them. It supports searching by file, folder, name, creation date, modification date, owner and permissions.



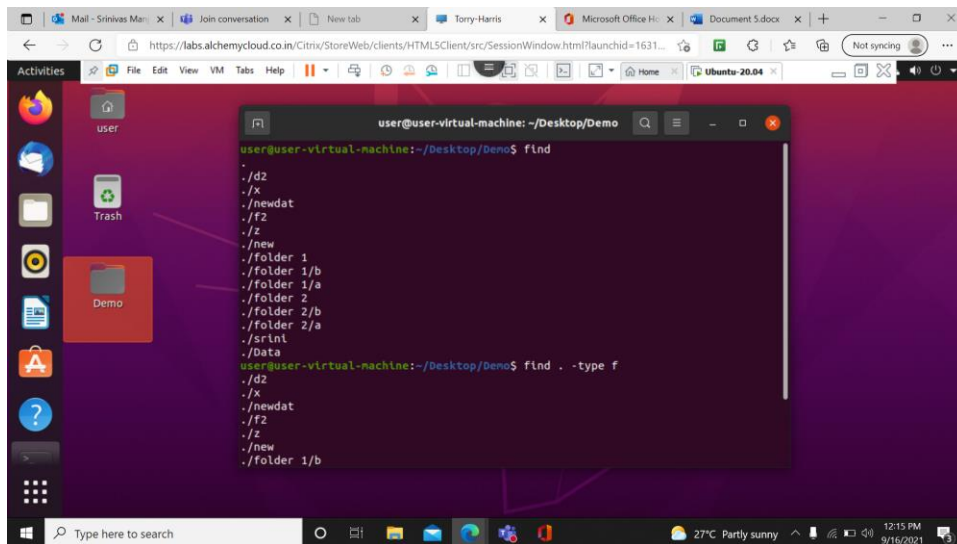
The screenshot shows a virtual machine desktop with a purple background. A terminal window titled 'user@user-virtual-machine: ~/Desktop/Demo' is open, displaying the following commands and output:

```
user@user-virtual-machine:~/Desktop/Demo$ ./folder1 -name a.txt
bash: ./folder1: No such file or directory
user@user-virtual-machine:~/Desktop/Demo$ find ./folder1 -name a.txt
find: './folder1': No such file or directory
user@user-virtual-machine:~/Desktop/Demo$ tree
.
├── d2
├── Data
├── f2
├── folder 1
│   ├── a
│   └── b
├── folder 2
│   ├── a
│   └── b
├── new
├── newdat
├── srini
├── x
└── z

2 directories, 12 files
user@user-virtual-machine:~/Desktop/Demo$ find new
new
user@user-virtual-machine:~/Desktop/Demo$ find -type f
```

A file explorer window titled 'user' is open, showing the contents of the 'Demo' folder. The files listed are 'd2', 'Data', 'f2', 'folder 1', 'folder 2', 'new', 'newdat', 'srini', 'x', and 'z'. The 'folder 1' and 'folder 2' folders are expanded, showing their sub-files 'a' and 'b'.

27> \$ find or \$find . - type f : to name the files in the folder.

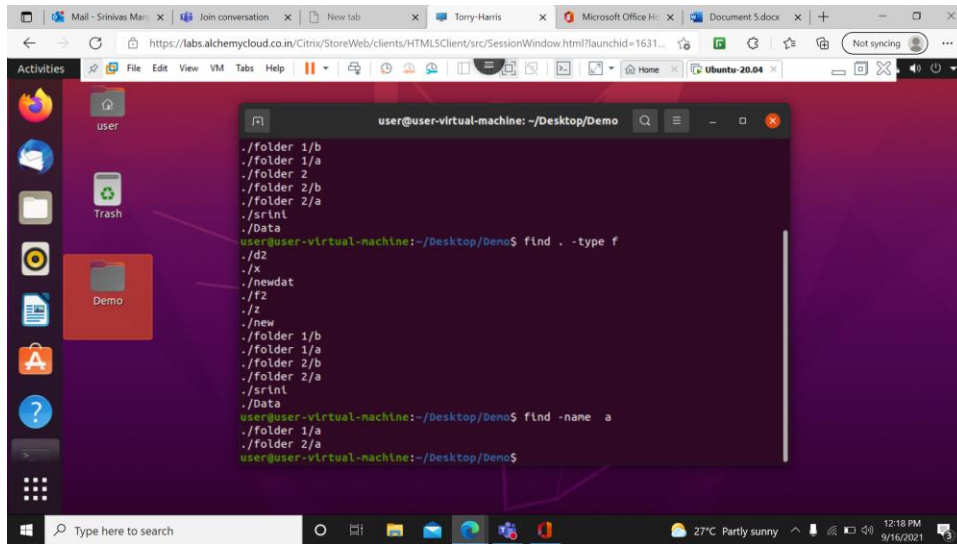


The screenshot shows the same virtual machine desktop as the previous one. The terminal window now displays the following commands and output:

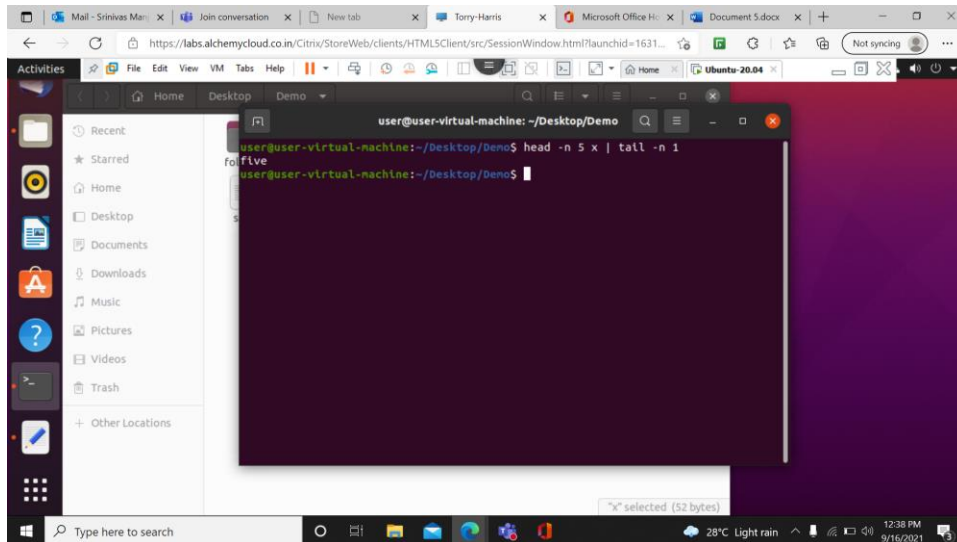
```
user@user-virtual-machine:~/Desktop/Demo$ find
./d2
./x
./newdat
./f2
./z
./new
./folder 1
./folder 1/b
./folder 1/a
./folder 2
./folder 2/b
./folder 2/a
./srini
./Data
user@user-virtual-machine:~/Desktop/Demo$ find . -type f
./d2
./x
./newdat
./f2
./z
./new
./folder 1/b
```

The file explorer window remains open, showing the same contents as before.

28> \$find -name filename : used to find the file in the folder, if the file is present in sub-folders means it shows that also.



29> To find the particular line from a file. `$head -n 5 num | tail -n 1`.



30> `$ls -t | head -n 3`

