

**Assignment 3: Create a function that takes a filename as an argument and prints the number of lines in the file. Call this function from your script with different filenames.**

**Ans:** Step1:

nano count\_line.sh

this command is use for the creating script file

after writing the script

use CTRL+O to save the file

press ENTER

For exit from the file PRESS

CTRL+X

The Script is:

```
#!/bin/bash
```

```
count_lines() {
```

```
    local filename=$1
```

```
    if [ -f "$filename" ]; then
```

```
        local line_count=$(wc -l < "$filename")
```

```
        echo "The file '$filename' has $line_count lines."
```

```
    else
```

```
        echo "Error: The file '$filename' does not exist."
```

```
    fi
```

```
}
```

```
count_lines "file1.txt"
```

```
count_lines "file2.txt"
```

```
count_lines "file3.txt"
```

Step3 :

Use command

```
chmod +x count_line.sh
```

This command is essential because it sets the necessary permissions on the file to allow it to be run as a program in your Linux environment

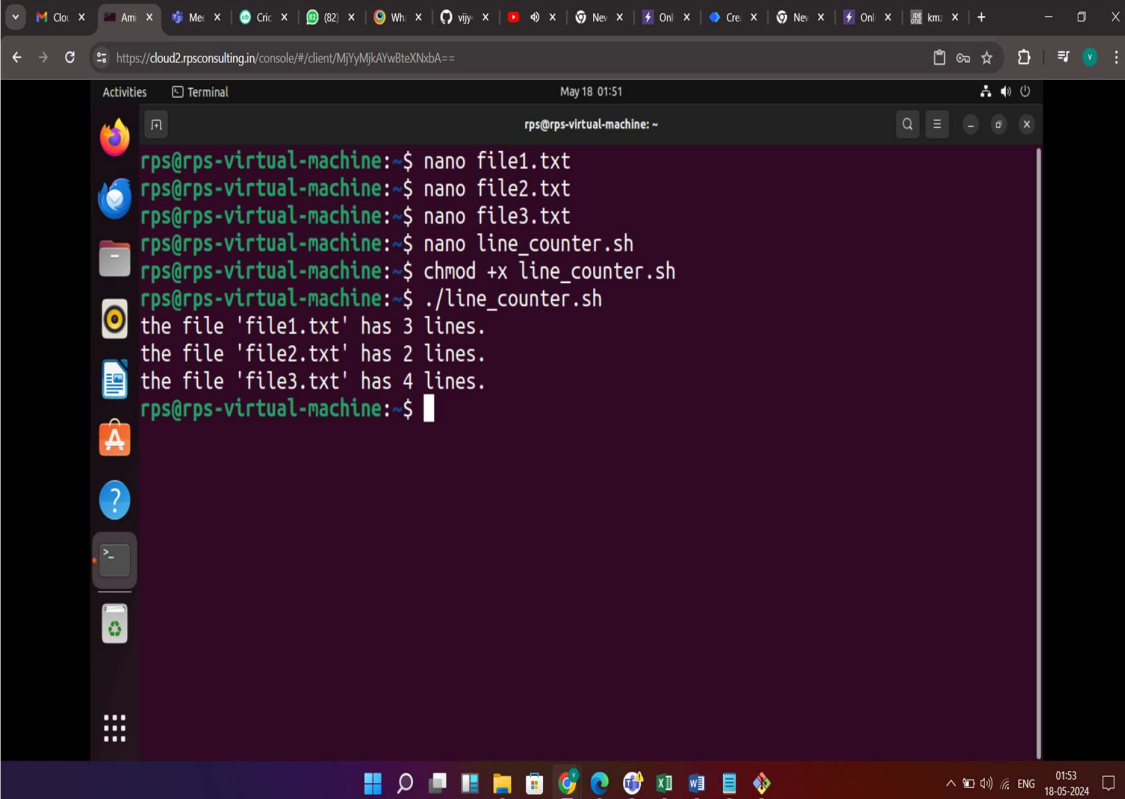
Step4:

Use `./count_line.sh`

To excute the script

Following is output snapshots:

output:

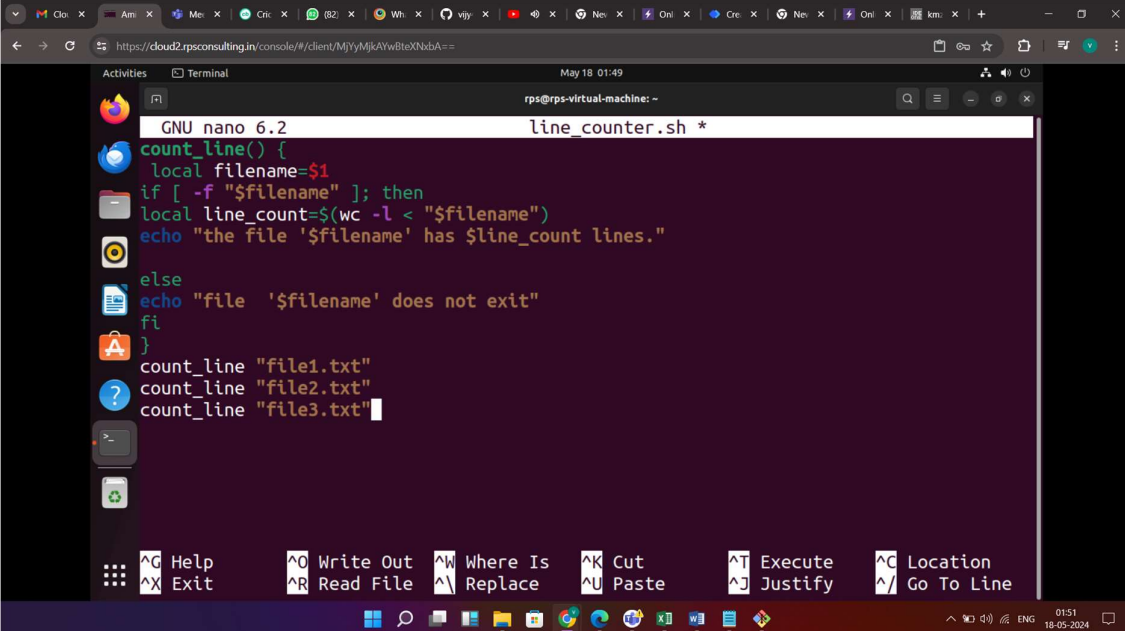


The screenshot shows a terminal window titled "rps@rps-virtual-machine: ~" with a timestamp of "May 18 01:51". The terminal displays the following commands and output:

```
rps@rps-virtual-machine:~$ nano file1.txt
rps@rps-virtual-machine:~$ nano file2.txt
rps@rps-virtual-machine:~$ nano file3.txt
rps@rps-virtual-machine:~$ nano line_counter.sh
rps@rps-virtual-machine:~$ chmod +x line_counter.sh
rps@rps-virtual-machine:~$ ./line_counter.sh
the file 'file1.txt' has 3 lines.
the file 'file2.txt' has 2 lines.
the file 'file3.txt' has 4 lines.
rps@rps-virtual-machine:~$
```

The terminal window is part of a desktop environment with a taskbar at the bottom showing various application icons and system status indicators (time: 01:53, date: 18-05-2024).

Script:



```
GNU nano 6.2 line_counter.sh *
count_line() {
  local filename=$1
  if [ -f "$filename" ]; then
    local line_count=$(wc -l < "$filename")
    echo "the file '$filename' has $line_count lines."
  else
    echo "file '$filename' does not exist"
  fi
}
count_line "file1.txt"
count_line "file2.txt"
count_line "file3.txt"
```

Terminal window showing the nano 6.2 editor editing the script `line_counter.sh`. The script defines a function `count_line()` that takes a filename as an argument and prints the number of lines in the file. The script is then executed with three arguments: `file1.txt`, `file2.txt`, and `file3.txt`.