

Process Text Streams using Filters

- **Introduction**
 - **Lab Topology**
 - **Exercise 1 - Process Text Streams using Filters**
 - **Review**
-

Introduction

Welcome to the **Process Text Streams using Filters** Practice Lab. In this module you will be provided with the instructions and devices needed to develop your hands-on skills.

Text Streams

Filters

Processing

Learning Outcomes

In this module, you will complete the following exercise:

- Exercise 1 - Process Text Streams using Filters

After completing this lab, you will be able to:

- Process Text Files Using Text Utility Filters

Exam Objectives

The following exam objectives are covered in this lab:

- **LPI: 109.3 Basic network troubleshooting**
- **LPI: 109.2 Persistent network configuration**
- **CompTIA: 1.3 Given a scenario, configure and verify network connection parameters.**

Note: Our main focus is to cover the practical, hands-on aspects of the exam objectives. We recommend referring to course material or a search engine to research theoretical topics in more detail.

Lab Duration

It will take approximately **1 hour** to complete this lab.

Help and Support

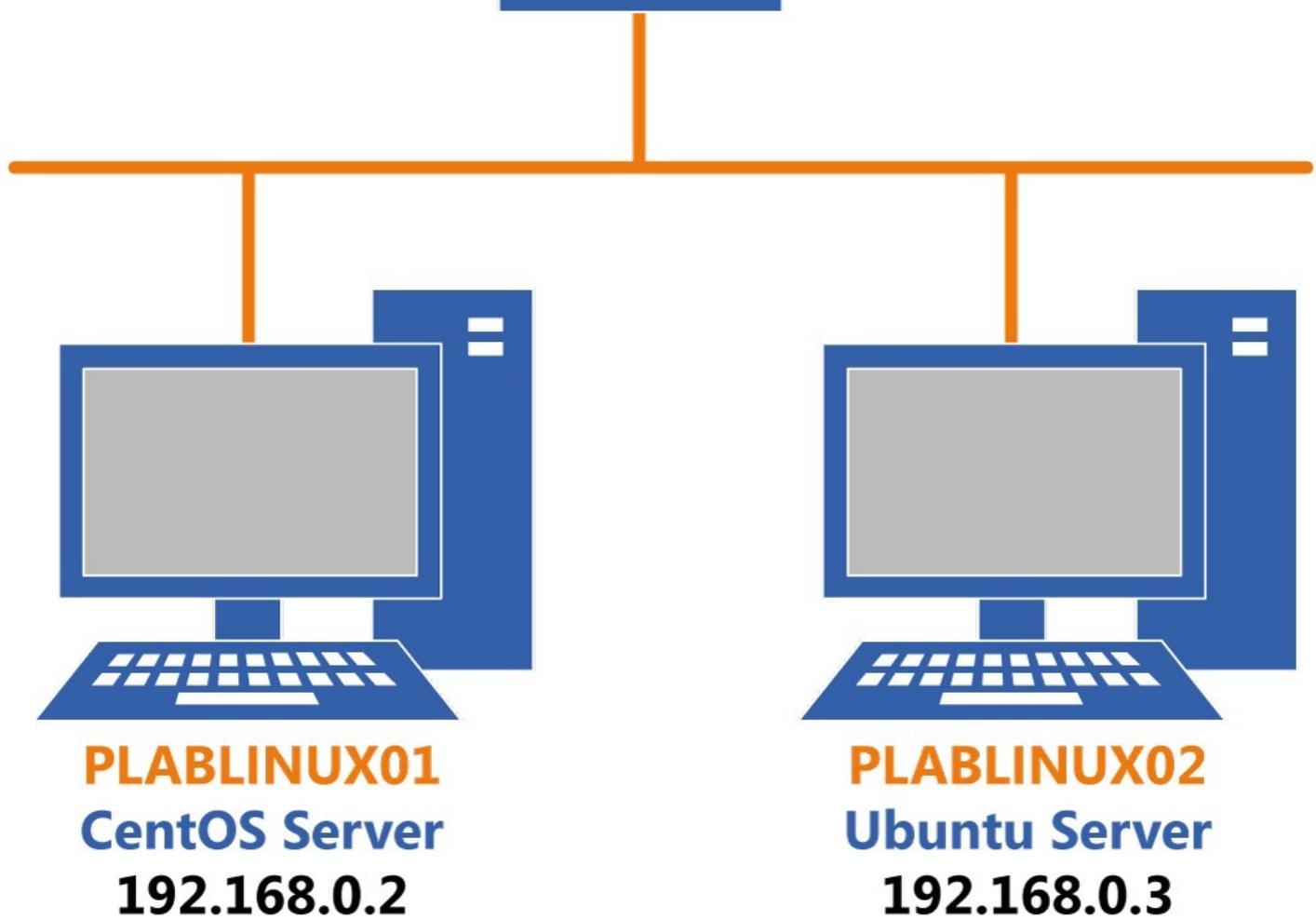
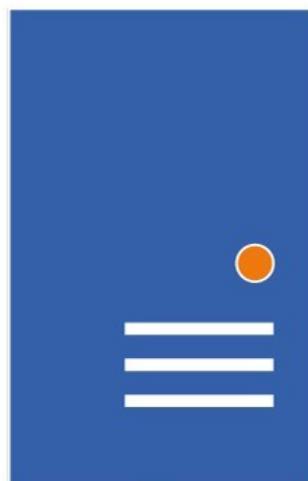
For more information on using Practice Labs, please see our **Help and Support** page. You can also raise a technical support ticket from this page.

Click Next to view the Lab topology used in this module.

Lab Topology

During your session, you will have access to the following lab configuration.

PLABSA01
Windows Server 2016
192.168.0.1



Depending on the exercises you may or may not use all of the devices, but they are shown here in the layout to get an overall understanding of the topology of the lab.

- **PLABSA01** (Windows Server 2016)
- **PLABLINUX01** (CentOS Server)
- **PLABLINUX02** (Ubuntu Server)

Click Next to proceed to the first exercise.

Exercise 1 - Process Text Streams using Filters

Linux OS provides commands that can be used as filters. These filters take the text from the specified source file, re-format the text as required, and store the output in a new file. Applying multiple commands can create text streams that are re-formatted at each command in the pipeline.

In this exercise, you will understand how to use filters to work with multiple commands.

Learning Outcomes

After completing this exercise, you will be able to:

- Log into a Linux System
- Process Text Files Using Text Utility Filters

Your Devices

You will be using the following device in this lab. Please power these on now.

- **PLABLINUX01** (CentOS Server)



Task 1 - Process Text Files Using Text Utility Filters

Some examples of text utility filters include **cat**, **cut**, **tr**, **pr**, **join**, **paste**, **split**, **od**, and more.

To use the filters, perform the following steps:

Step 1

On the desktop, right-click and select **Open Terminal**.

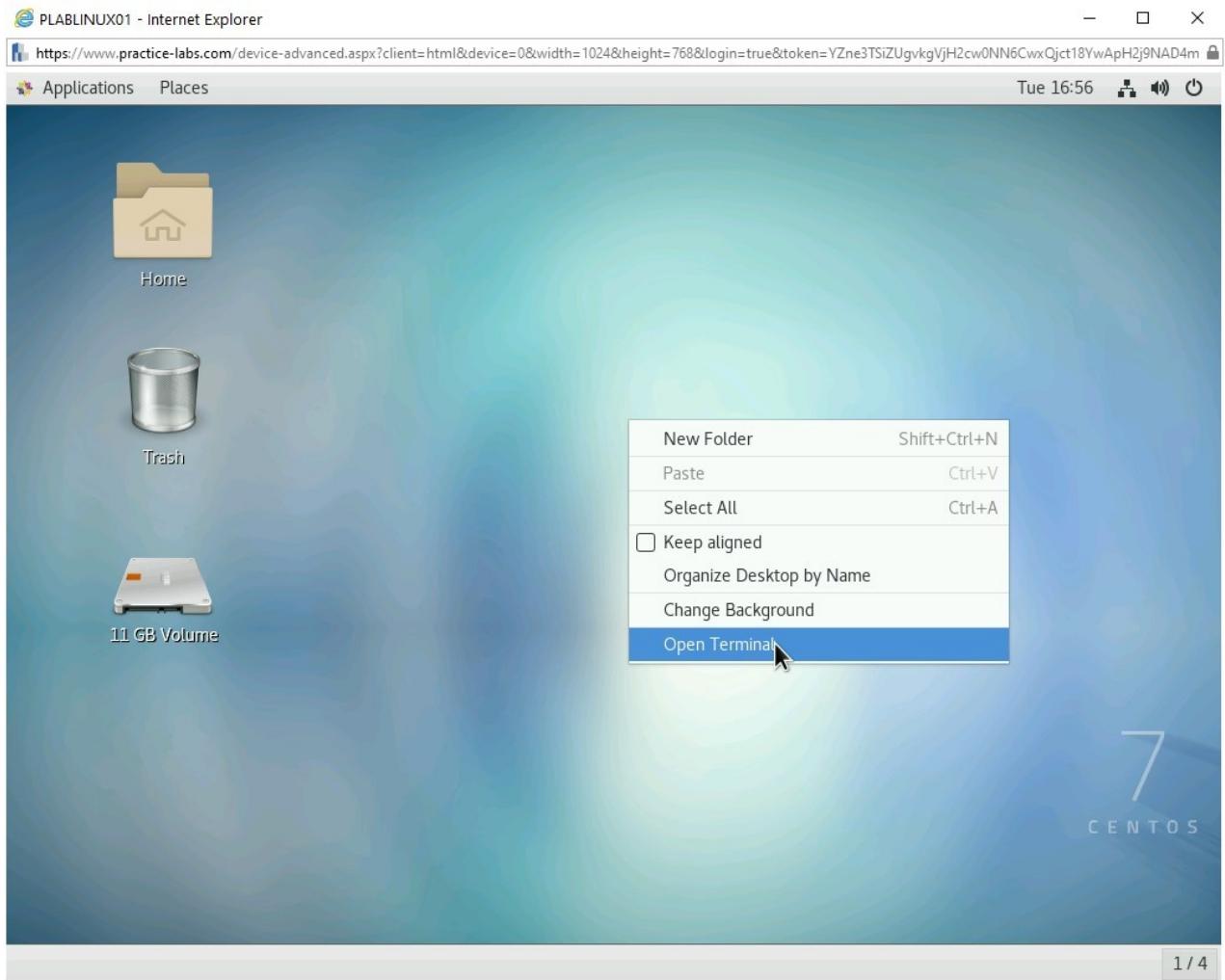


Figure 1.1 Screenshot of PLABLINUX01: Selecting the Open Terminal option from the context menu.

Step 2

The command prompt window is displayed. Type the following command:

```
su -
```

Press **Enter**.

At the **Password** prompt, type the following password:

Passw0rd

Press **Enter**.

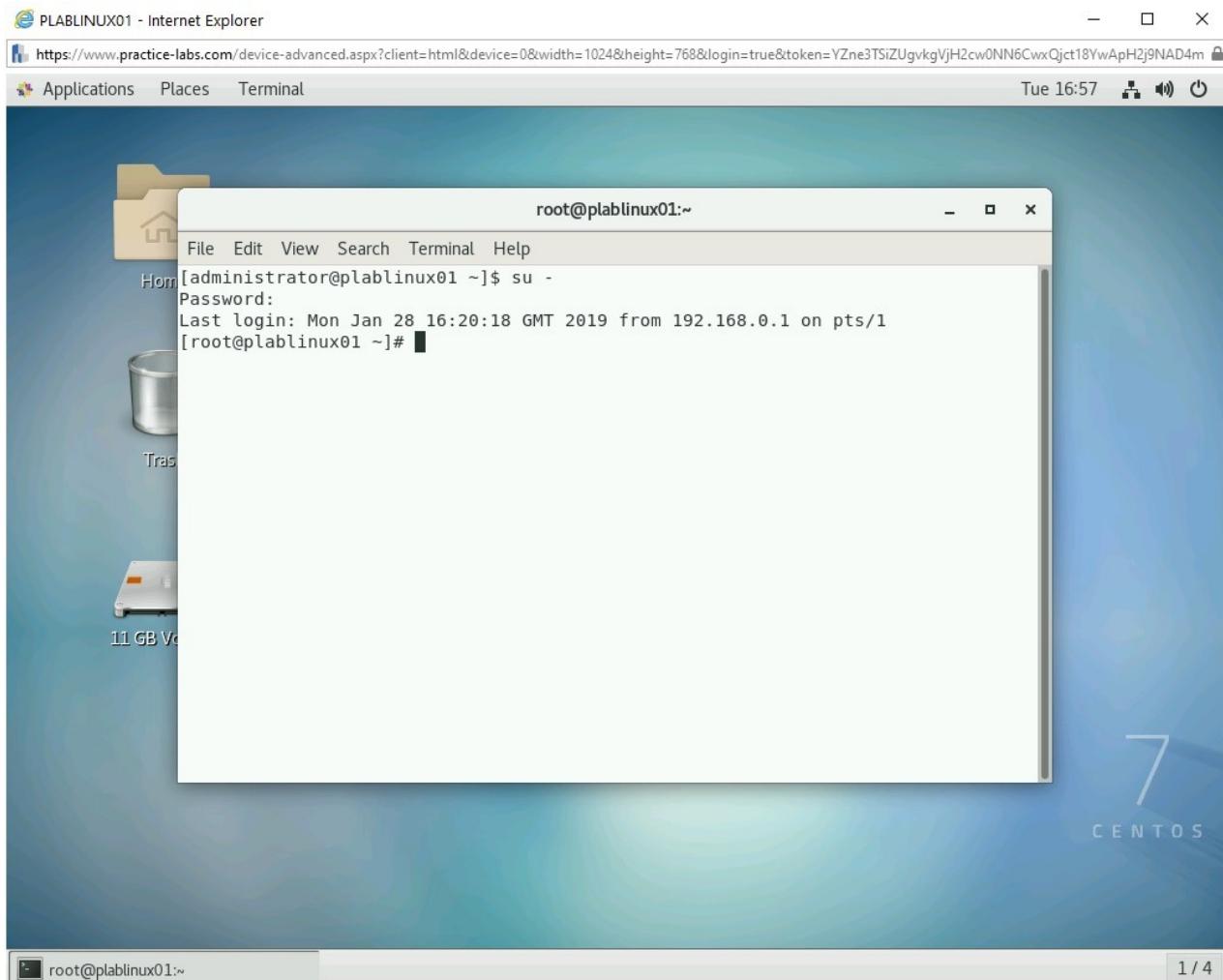


Figure 1.2 Screenshot of PLABLINUX01: Changing the account to the root account with the su command.

Step 3

Clear the screen by entering the following command:

```
clear
```

The cat command is used to print the output of a file onto the screen. You can also use this command to send the content of one file to another.

To print the output of the **/etc/yum.conf** file onto the screen, type the following command:

```
cat /etc/yum.conf
```

Press **Enter**.

Note: For this task, the **yum.conf** file is being referenced. You can choose any file you may want.

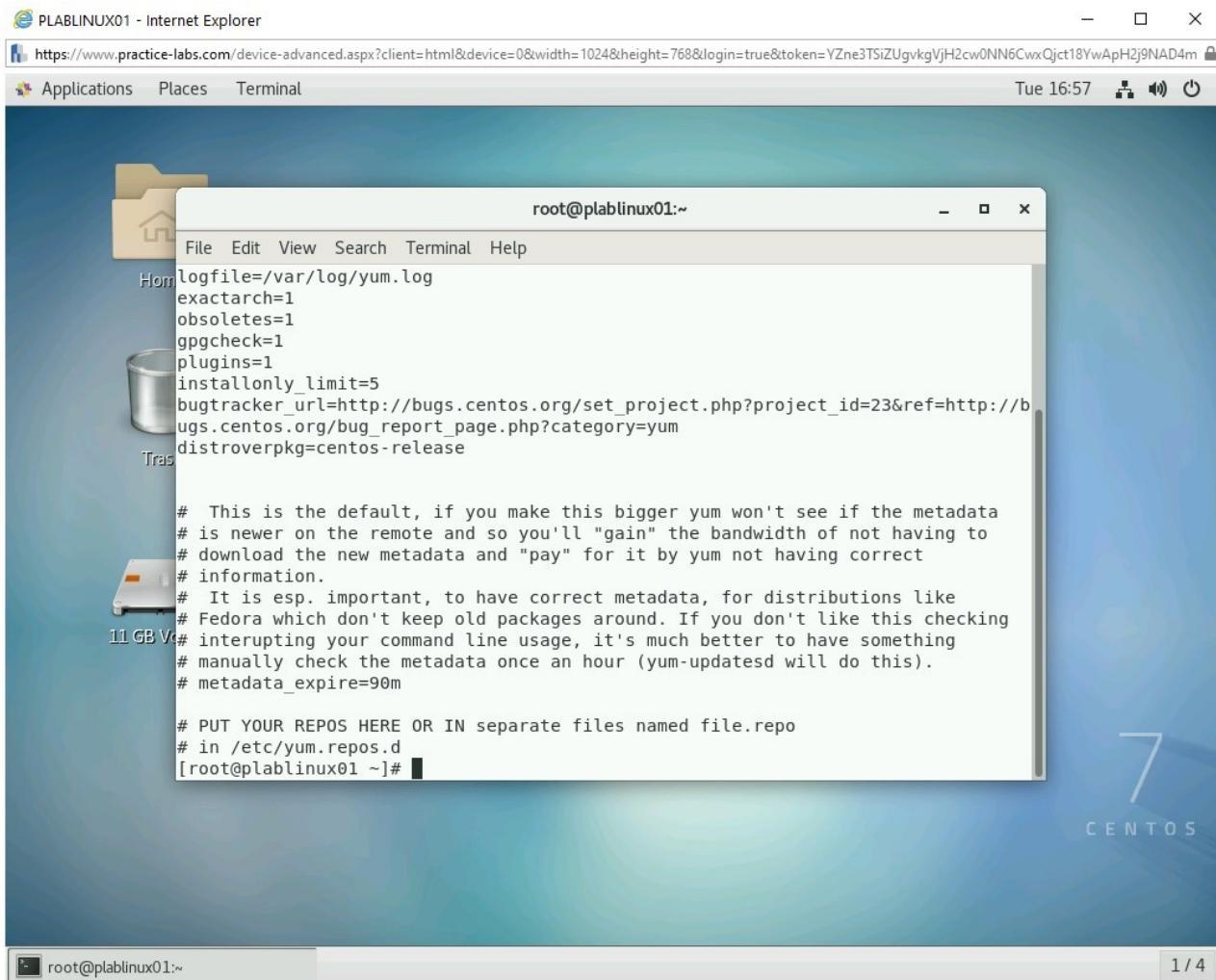


Figure 1.3 Screenshot of PLABLINUX01: Printing the output of the /etc/yum.conf file onto the screen.

Step 4

Clear the screen by entering the following command:

```
clear
```

To send the content of one file to another file, you can use the > redirector. Assuming that you want to redirect the content of **yum.conf** into another file **yumtemp.conf**, type the following command:

```
cat /etc/yum.conf > /tmp/yumtemp.conf
```

Press **Enter**.

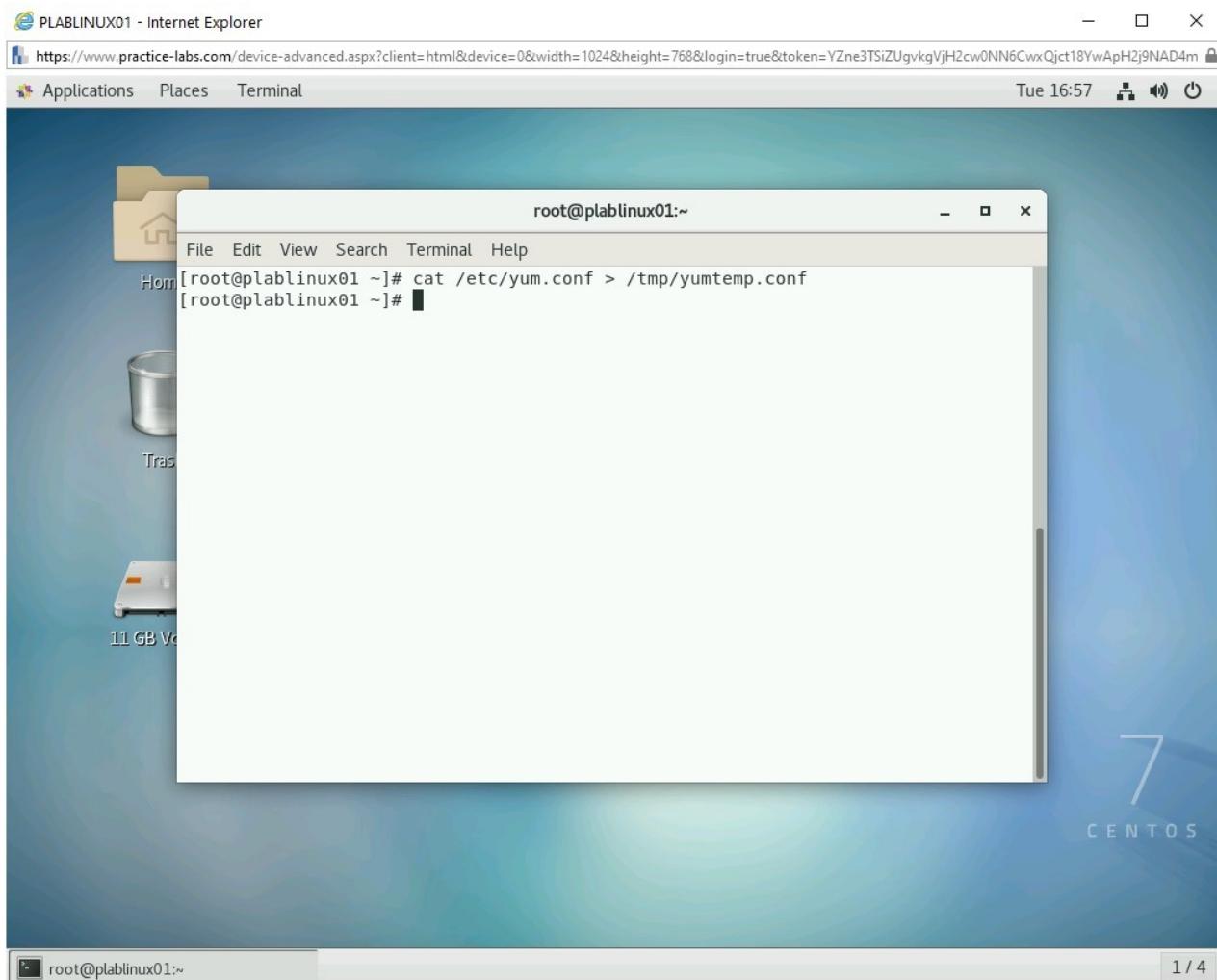


Figure 1.4 Screenshot of PLABLINUX01: Sending the content of one file to another file.

Step 5

Clear the screen by entering the following command:

```
clear
```

The cut command is another filter that is used to print the selected output of a file onto the screen. However, the source file remains as is and no modifications are made to it.

For example, to display only the first column of the **passwd** file, type the following command:

```
cut -d: -f 1 /etc/passwd
```

Press **Enter**.

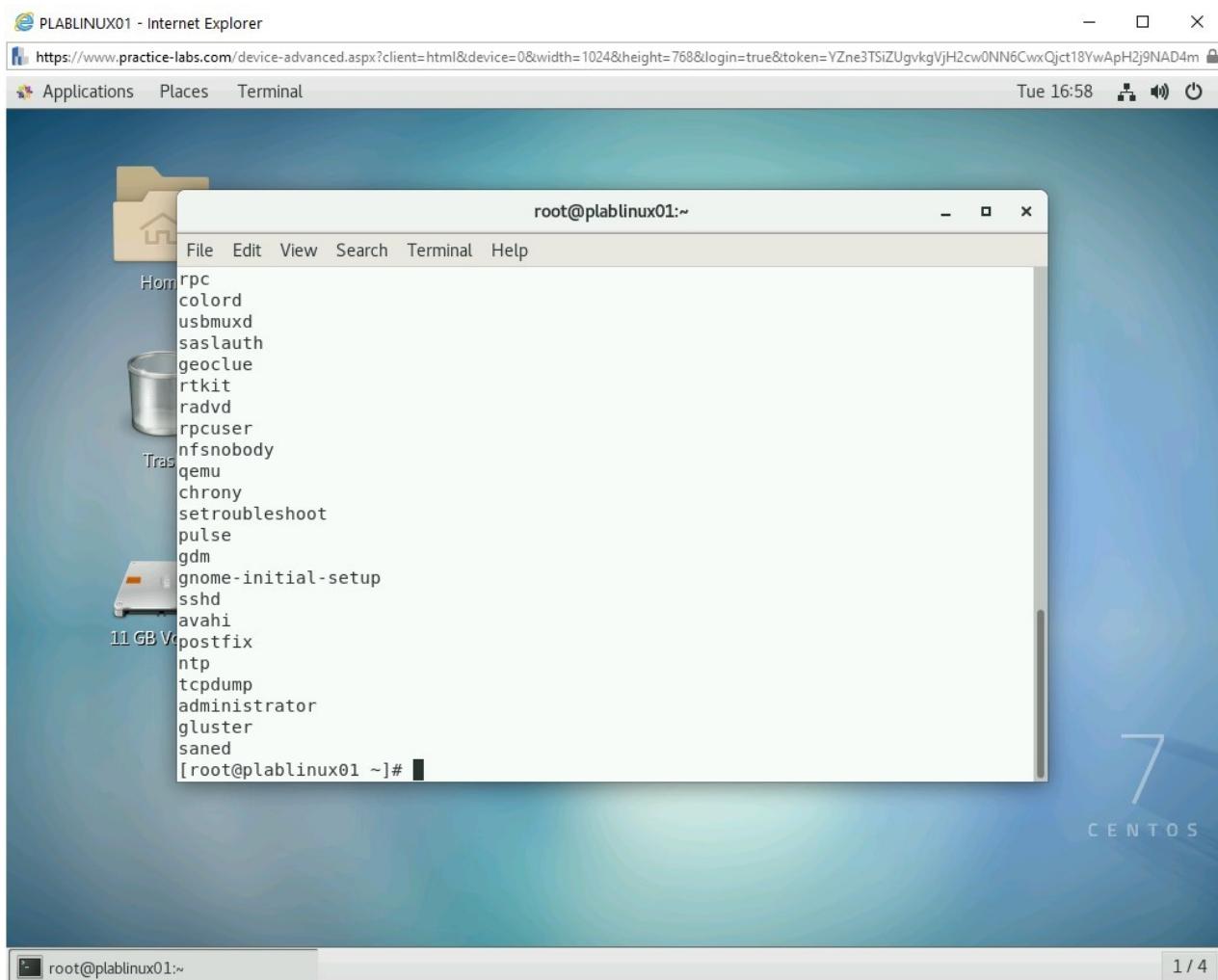


Figure 1.5 Screenshot of PLABLINUX01: Displaying only the first column of the passwd file.

Step 6

The **passwd** file has six columns. You can again use the cut command to view, say the last column in this file.

Type the following command:

```
cut -d: -f 6 /etc/passwd
```

Press **Enter**.

Notice that only the last column is displayed.

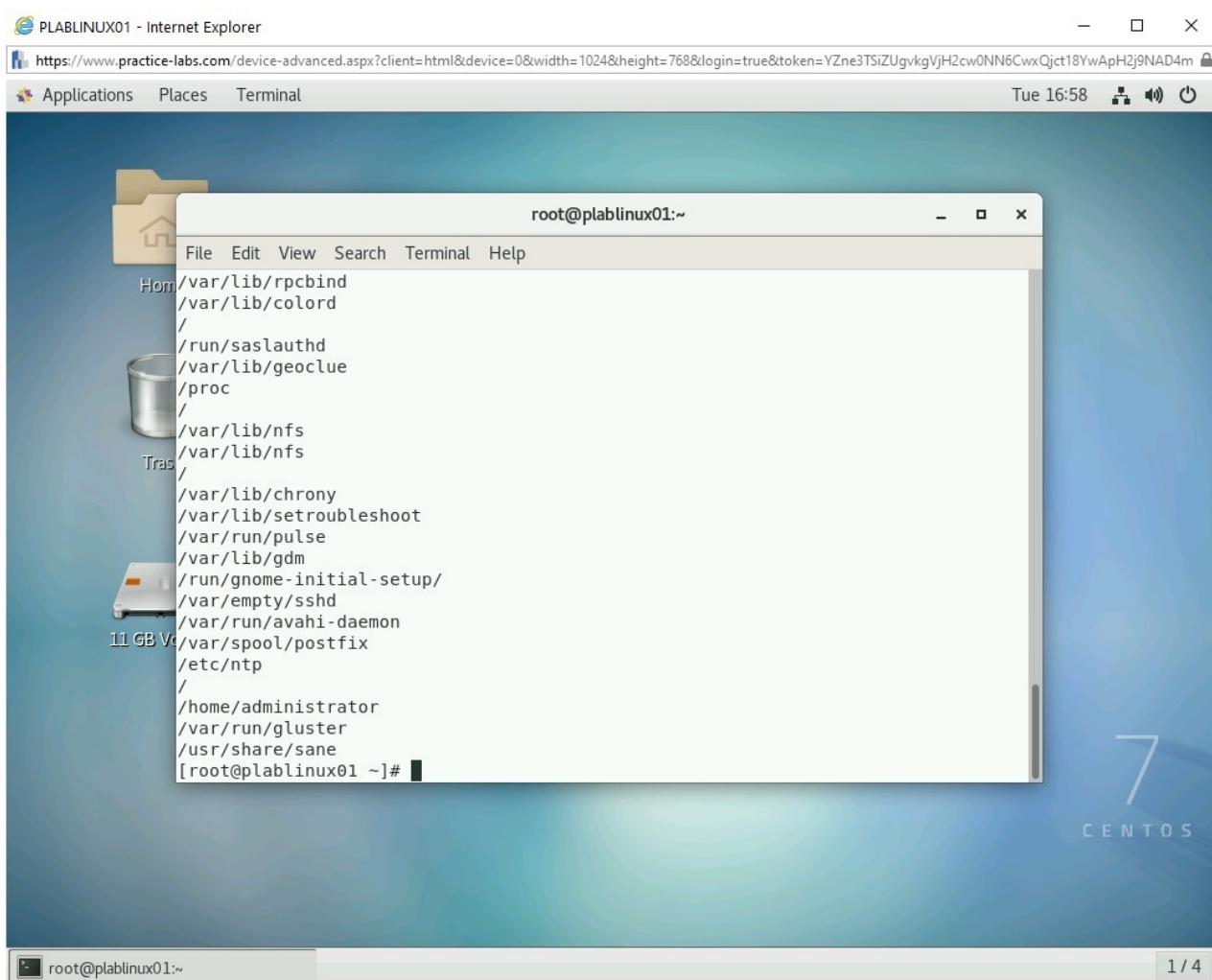


Figure 1.6 Screenshot of PLABLINUX01: Displaying only the last column of the passwd file.

Step 7

Let's now view the original file, **/etc/passwd**.

Type the following command:

```
cat /etc/passwd
```

Press **Enter**.

Notice that the output displays the complete **/etc/passwd** file. Therefore, the source file is not modified in any way due to the cut command.

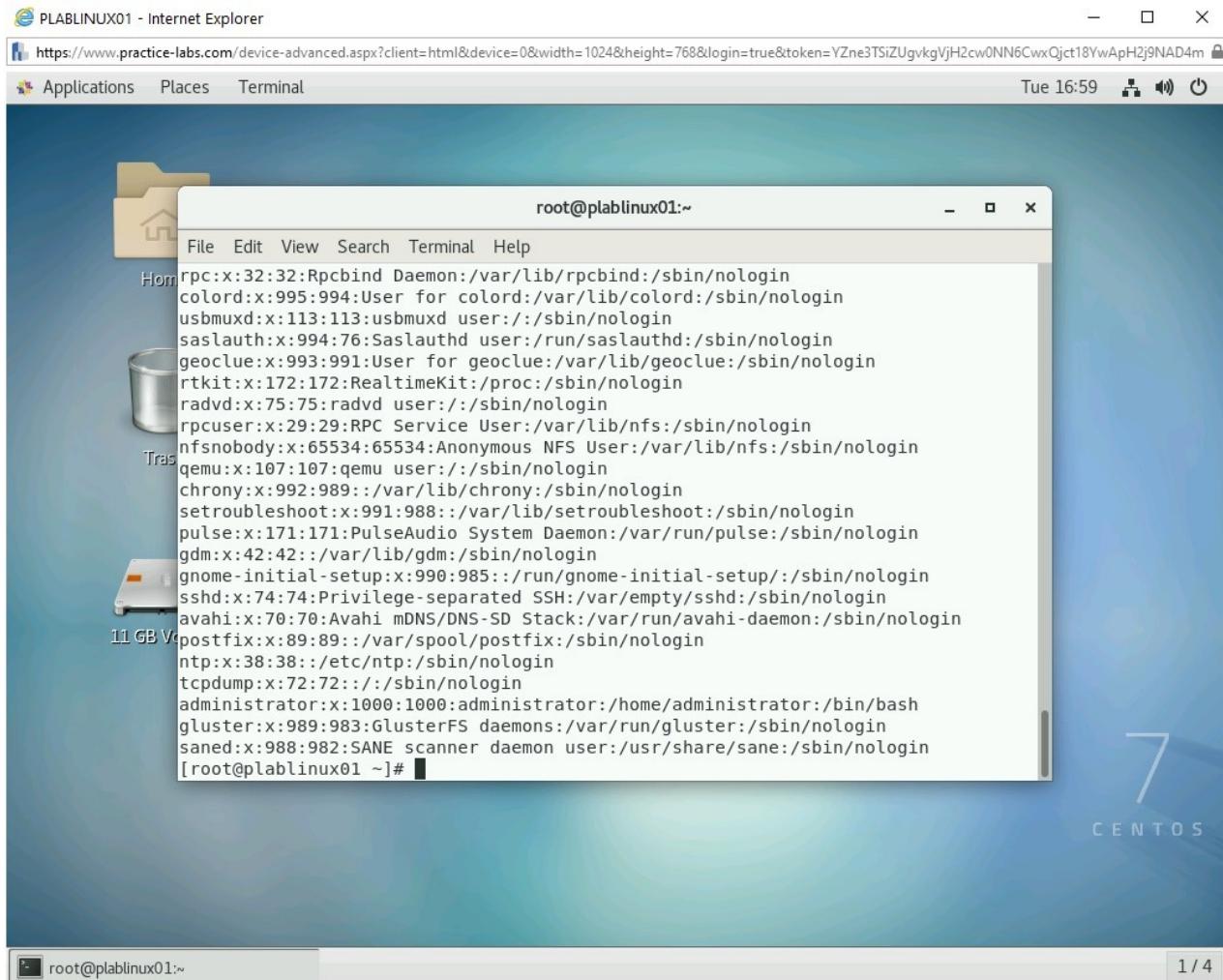


Figure 1.7 Screenshot of PLABLINUX01: Displaying the complete /etc/passwd file.

Step 8

Clear the screen by entering the following command:

```
clear
```

The expand command is another filter that converts tabs to spaces, without affecting the source file. By default, a tab consists of eight spaces.

Before using this command, let's create a text file with tabs. Type the following command:

```
vi
```

Press **Enter**.

A new blank text file is opened. On the command line and put it into the insert mode by pressing:

```
i
```

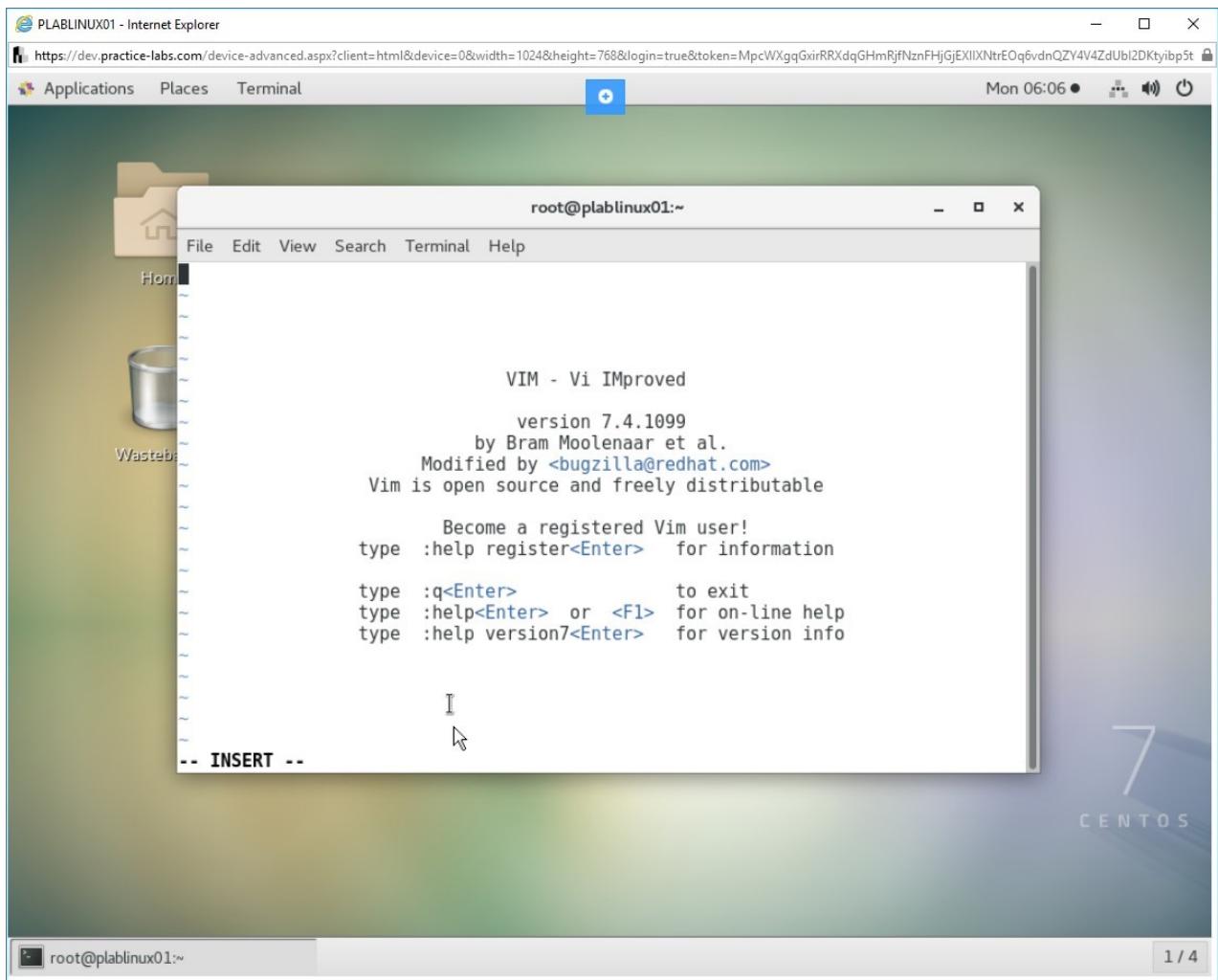


Figure 1.8 Screenshot of PLABLINUX01: Creating and entering the file into insert mode.

Step 9

Type the following text:

```
Test test
```

Press enter and the same text three more times.

Ensure that the words are entered, save the file by pressing **ESC** and then typing the following command:

```
:wq newfile.txt
```

Press **Enter**.

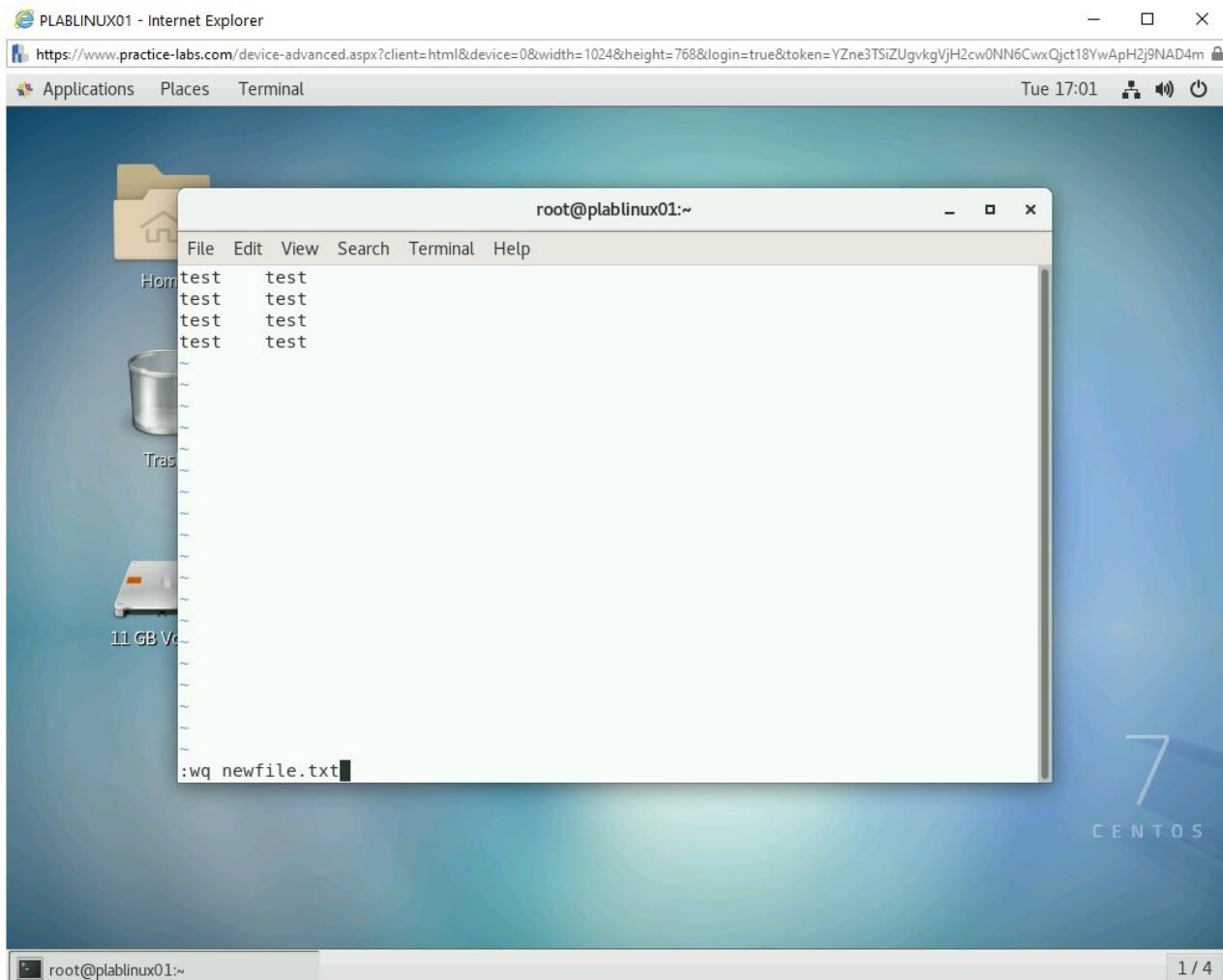


Figure 1.9 Screenshot of PLABLINUX01: Entering text in the file and then saving the file with the name newfile.txt.

Step 10

To view the saved file using the expand command, type the following command:

```
expand newfile.txt
```

Press **Enter**.

Note that the output is the default tab with eight spaces.

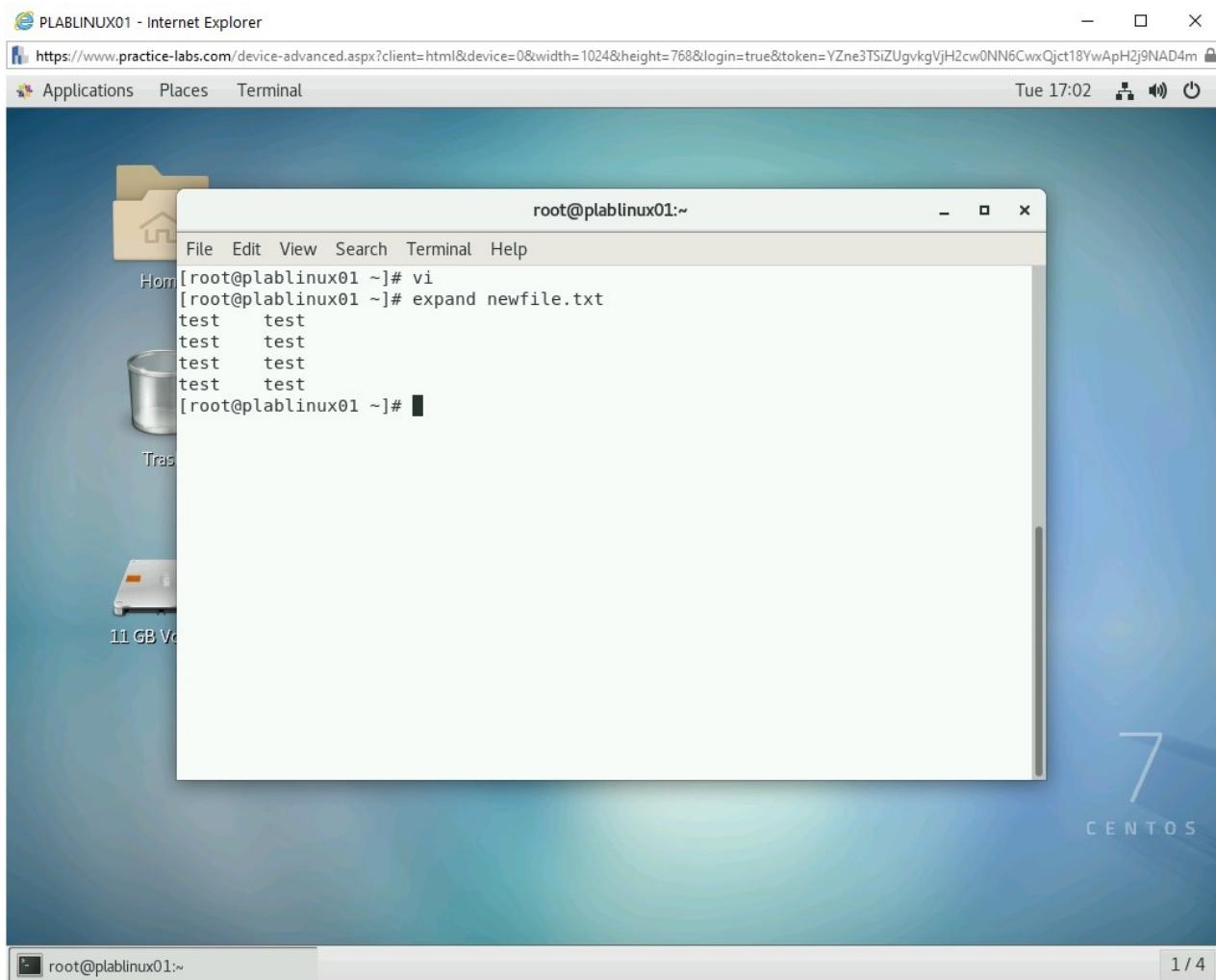


Figure 1.10 Screenshot of PLABLINUX01: Viewing the saved file using the expand command.

Step 11

Now, you can use the expand command to alter the tab spaces.

Let's increase the number of spaces.

Type the following command:

```
expand --tab=20 newfile.txt
```

Press **Enter**.

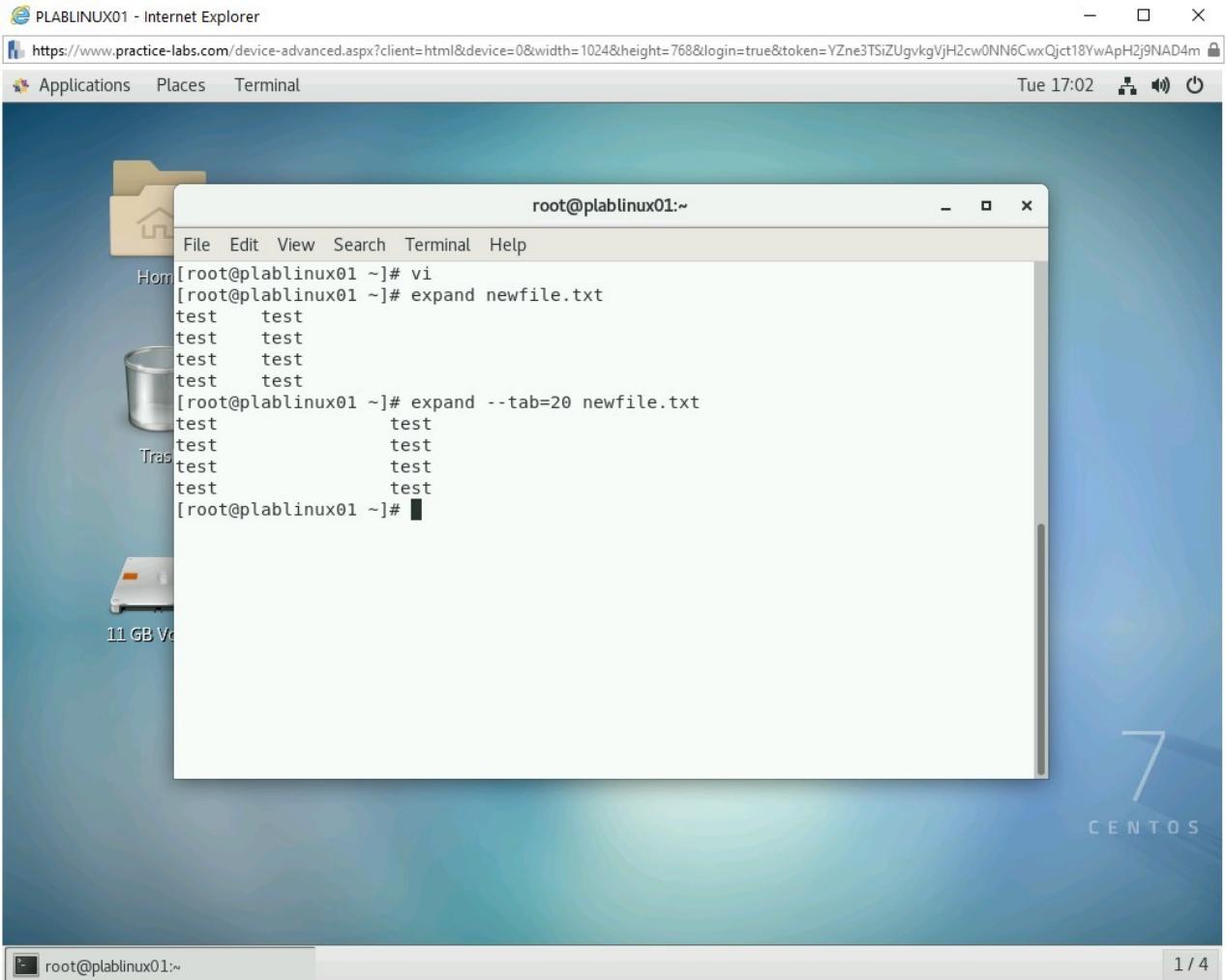


Figure 1.11 Screenshot of PLABLINUX01: Altering the tab spaces using the expand command.

Step 12

You can also redirect the output with the increased number of spaces to another file. Type the following command:

```
expand --tab=20 newfile.txt > newfile1.txt
```

Press **Enter**.

Note the use of the redirector operator (>)in the command.

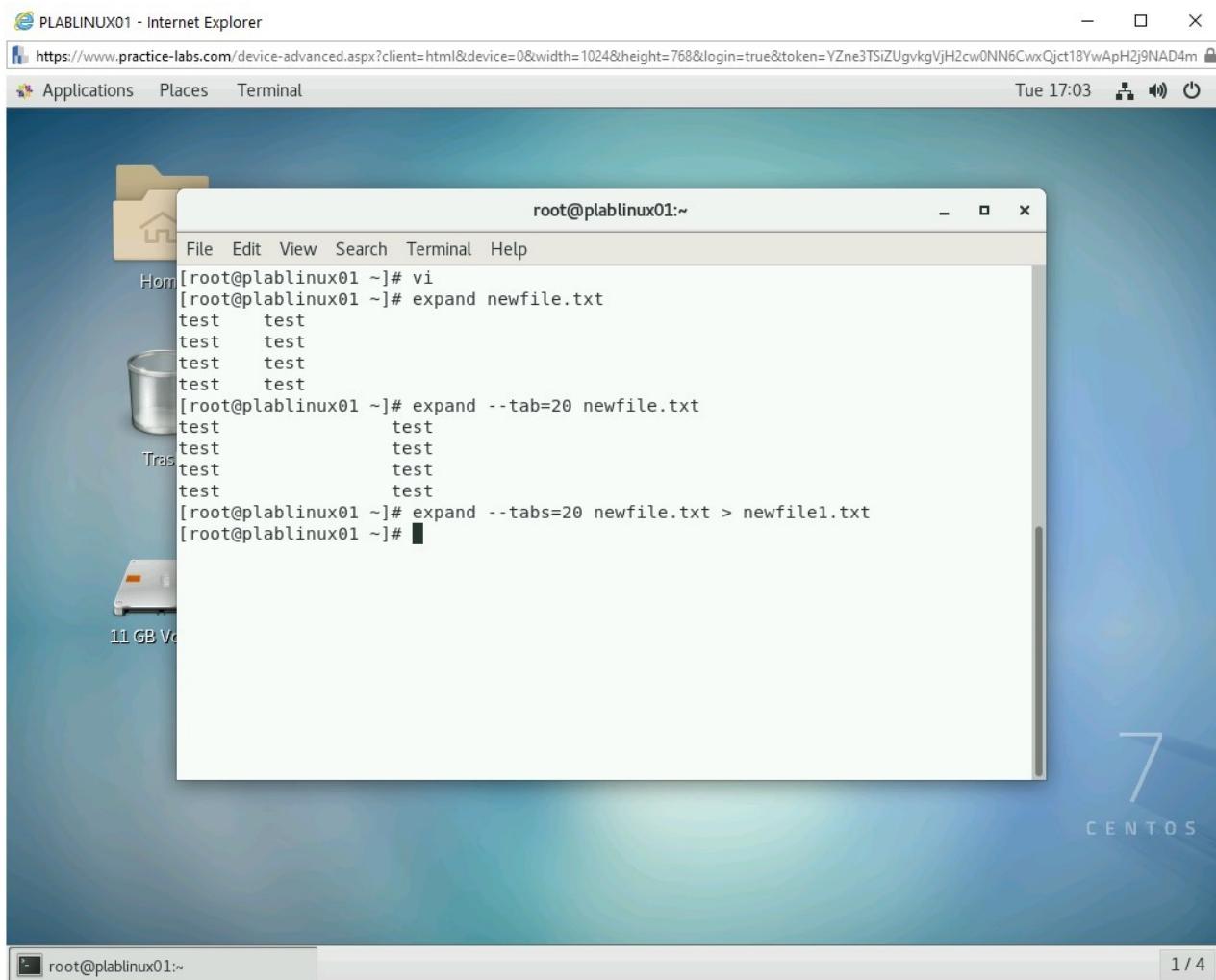


Figure 1.12 Screenshot of PLABLINUX01: Redirecting the output with the increased number of spaces to another file.

Step 13

To view the contents of the **newfile1.txt**, type the following command:

```
cat newfile1.txt
```

Press **Enter**.

Note: Alternately, you can also use the command, expand newfile1.txt, to view the contents of **newfile1.txt**.

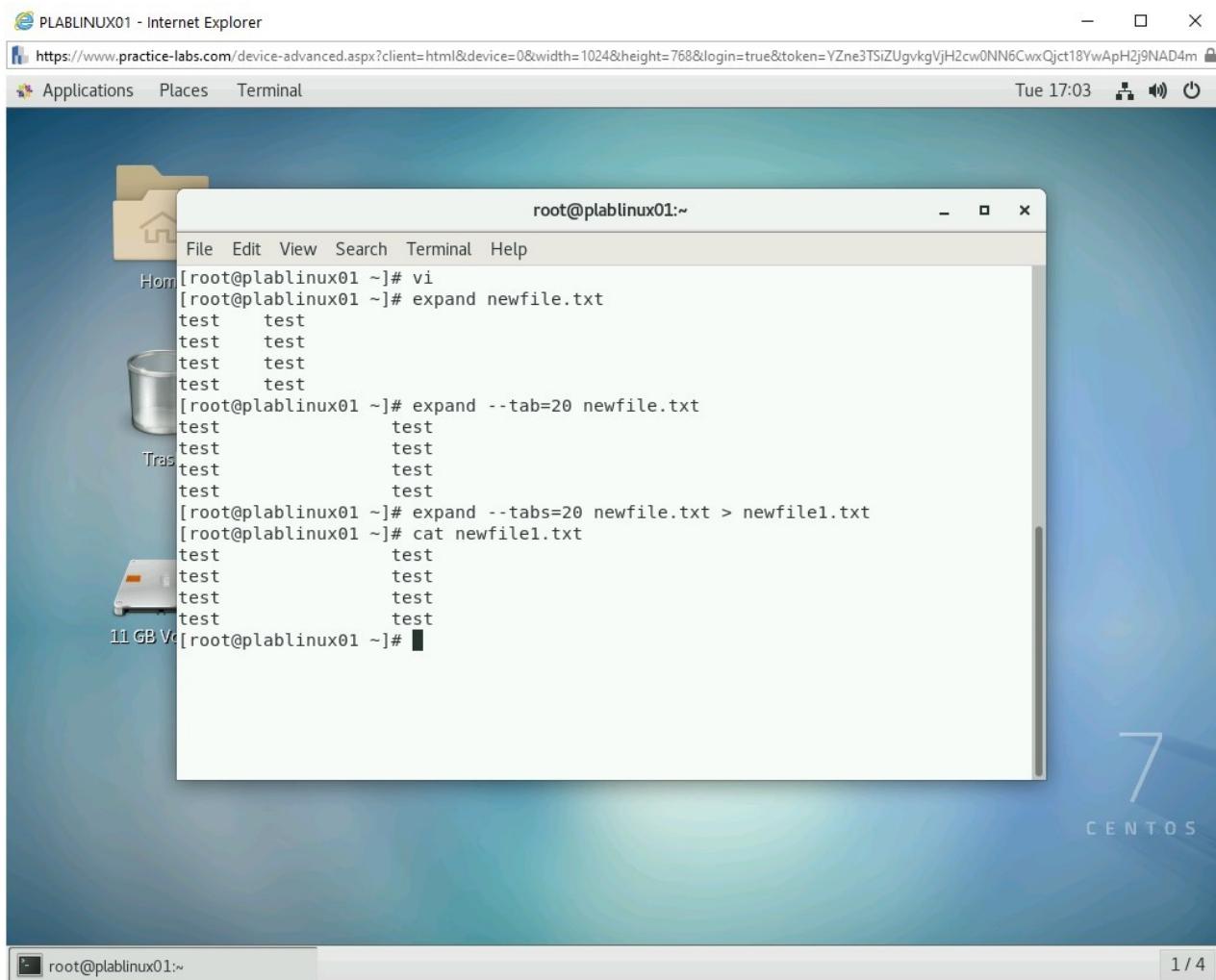


Figure 1.13 Screenshot of PLABLINUX01: Viewing the contents of the newfile1.txt.

Step 14

Clear the screen by entering the following command:

```
clear
```

Similar to expand, you can also use unexpand to convert the spaces back to default tabs in a file. Type the following command:

```
unexpand newfile1.txt
```

Press **Enter**.

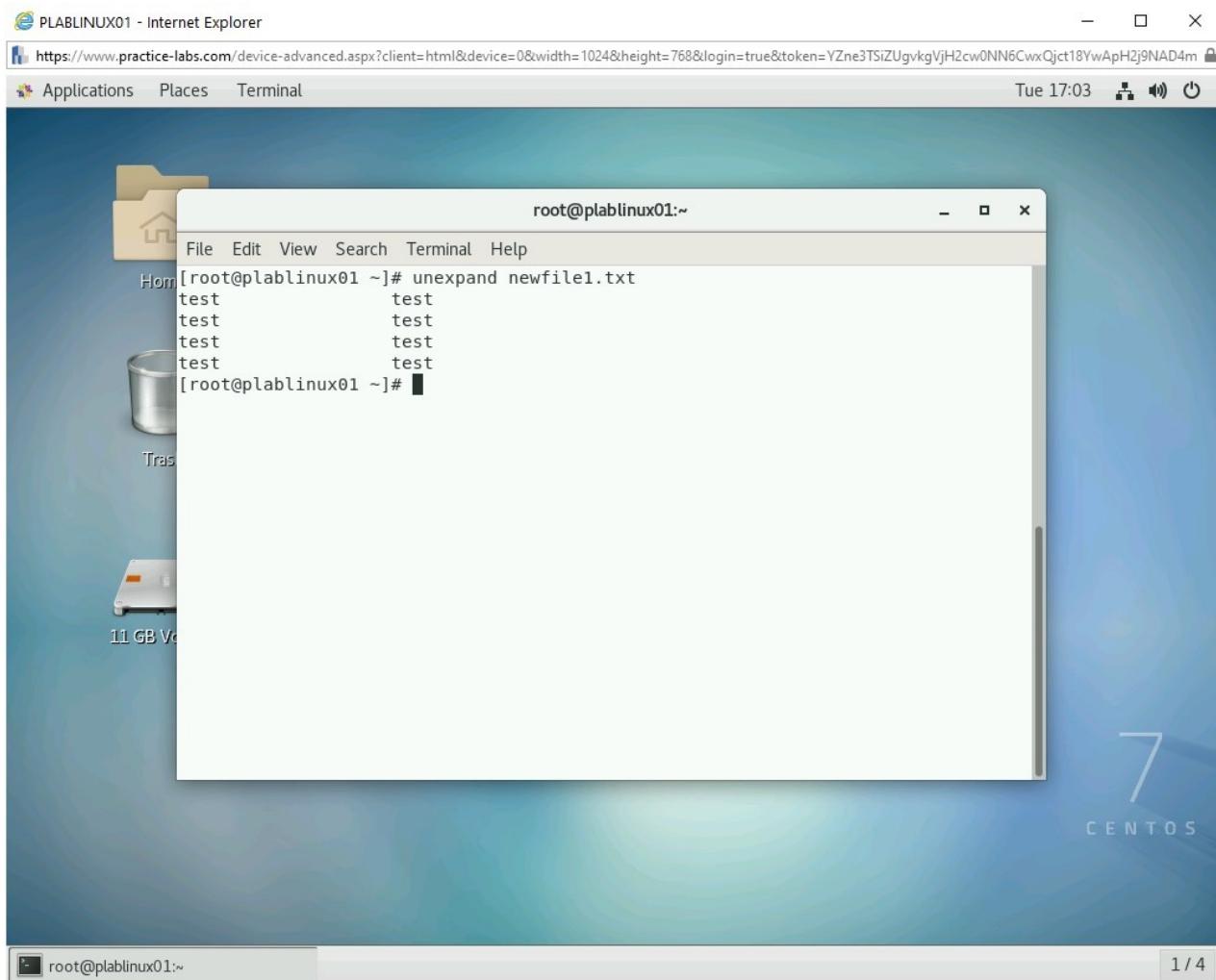


Figure 1.14 Screenshot of PLABLINUX01: Unexpanding the newfile1.txt file.

Step 15

Clear the screen by entering the following command:

```
clear
```

The fmt command is used to define the uniform spacing between words and sentences. When used with the -u parameter, it formats the text with one space between words and two spaces between sentences.

To re-format the **newfile1.txt** file, type the following command:

```
fmt -u newfile1.txt
```

Press **Enter**.

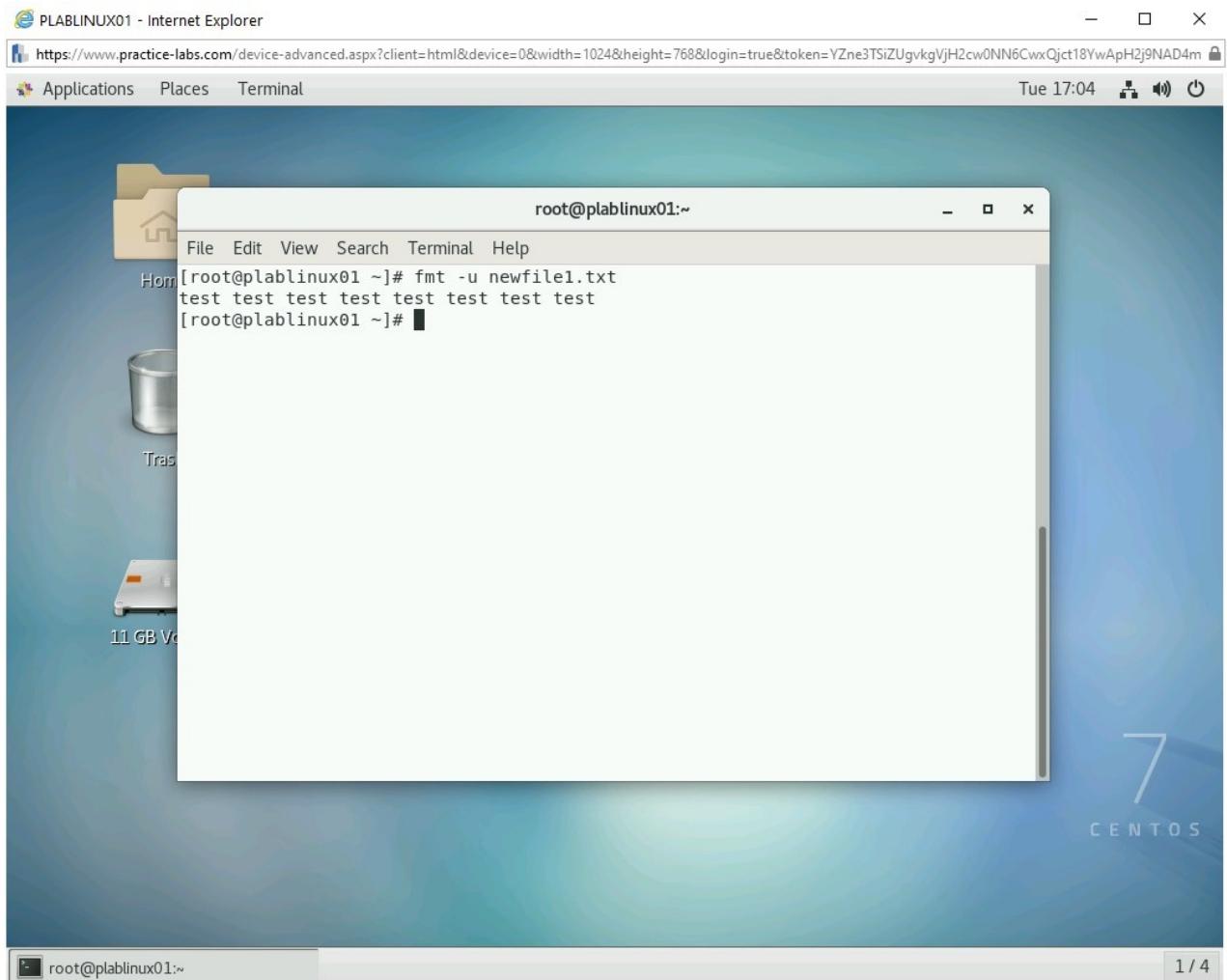


Figure 1.15 Screenshot of PLABLINUX01: Re-formatting the newfile1.txt file.

Step 16

Clear the screen by entering the following command:

```
clear
```

The head command is used to display a specific number of lines of a file. In addition, you can use this command to display a specific number of initial lines from multiple files. The output of these files is displayed in the sequential order in which they are defined.

By default, the head command displays the first 10 lines of a file. You can use the `-n` parameter to specify any other number of lines to display. Type the following command:

```
head /etc/yum.conf
```

Press **Enter**.

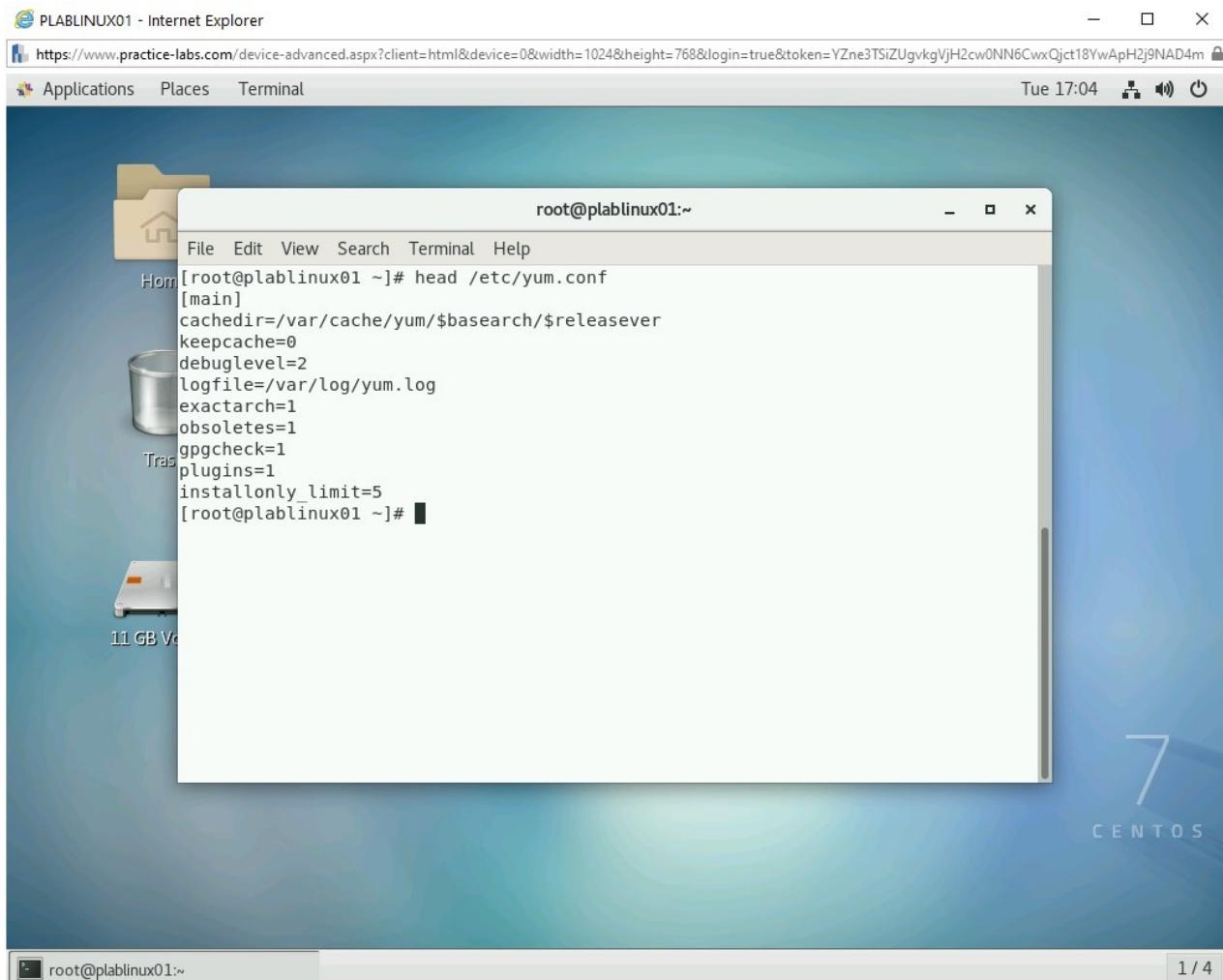


Figure 1.16 Screenshot of PLABLINUX01: Displaying the first 10 lines of the /etc/yum.conf file.

Step 17

Let's use the -n parameter and display only the first five lines. Type the following command:

```
head -n5 /etc/yum.conf
```

Press **Enter**.

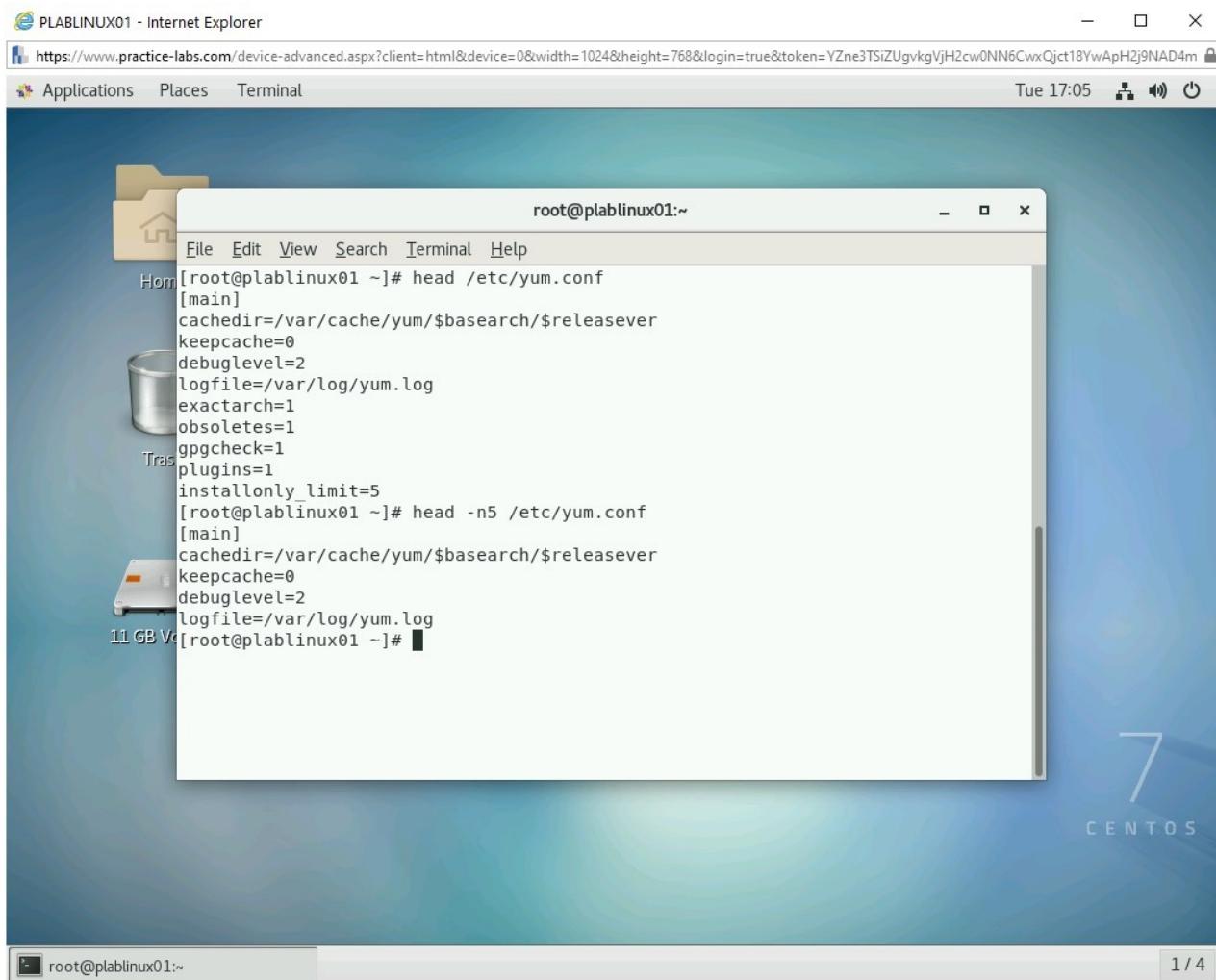


Figure 1.17 Screenshot of PLABLINUX01: Displaying the first five lines of the /etc/yum.conf file.

Step 18

Clear the screen by entering the following command:

```
clear
```

Let's now display the output from multiple files in sequential order. Type the following command:

```
head -n5 /etc/yum.conf /etc/passwd
```

Press **Enter**.

Note: The display from multiple files marks the output from each file with a header naming the file.

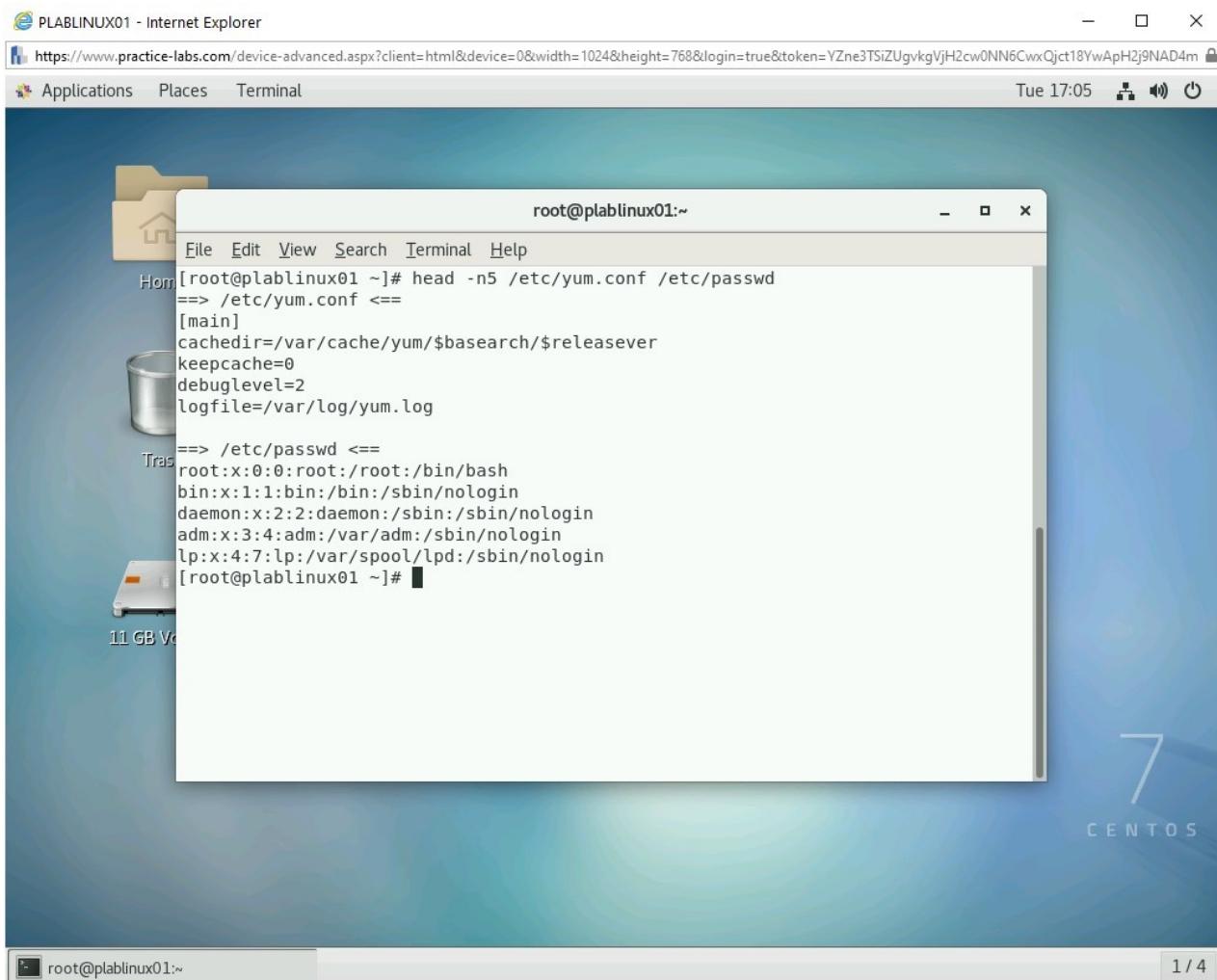


Figure 1.18 Screenshot of PLABLINUX01: Displaying the first five lines of multiple files in a sequence.

Step 19

Clear the screen by entering the following command:

```
clear
```

You can use the **join** command to display the combined contents of the specified files. The join command combines the contents of each line of the source files and displays the output as a combined-single line for each line in the source files. The join command is more like a **SQL join** where the data is drawn from multiple tables and displayed as a combined result.

For this task, you will create two files called **fruits.txt** and **rates.txt**. To create these, you will step similar to **Step 8**.

To begin, let's create the **fruits.txt** file. Type the following command:

```
vi
```

A new blank text file is created. On the command line and put it into the insert mode by pressing:

```
i
```

Type the following text:

```
Apples
```

Press **Enter**.

Type the following text:

```
Oranges
```

Note: please refer to the screenshot to see how it needs to be formatted.

Now, save the file by pressing **ESC** and then typing the following command:

```
:wq fruits.txt
```

Press **Enter**.

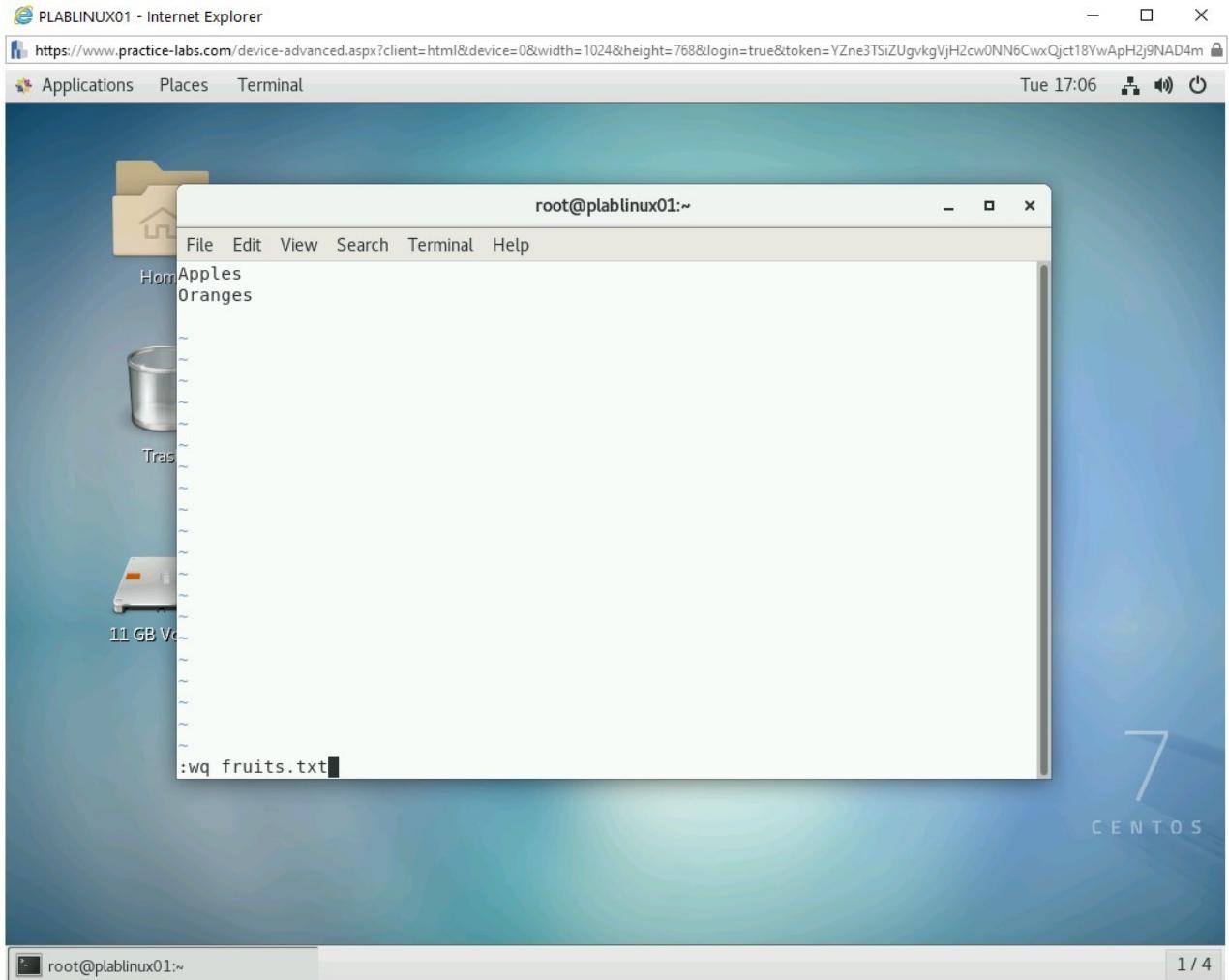


Figure 1.19 Screenshot of PLABLINUX01: Creating and saving a new file named fruits.txt.

Step 20

Follow the same as step 19 but enter:

```
$10  
$5
```

And name the file **rates.txt**.

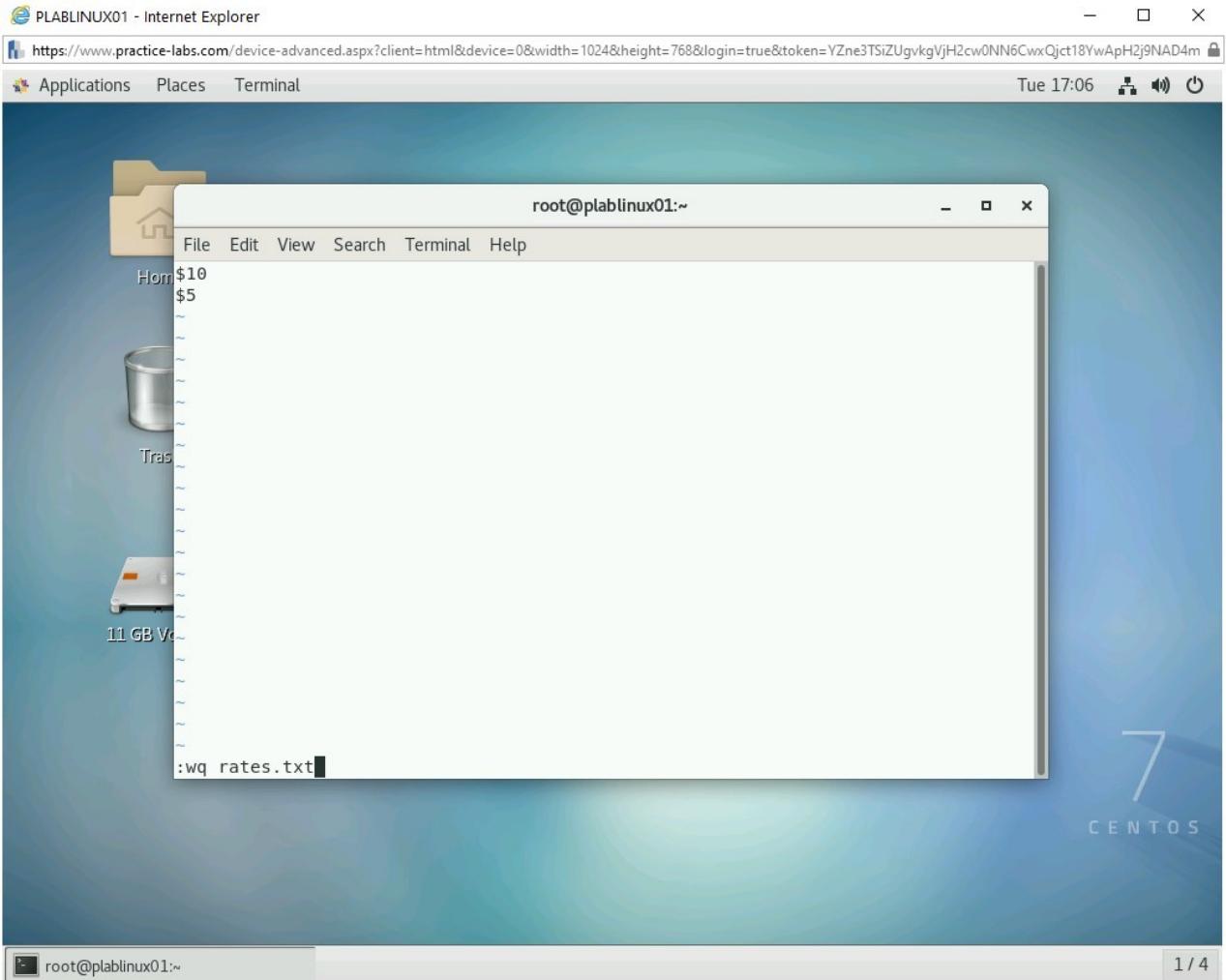


Figure 1.20 Screenshot of PLABLINUX01: Creating and saving a new file named **rates.txt**.

Step 21

To view the contents of the **fruits.txt** file, type the following commands:

```
cat fruits.txt
```

Press **Enter**.

To view the contents of the **rates.txt** file, type the following commands:

```
cat rates.txt
```

Press **Enter**.

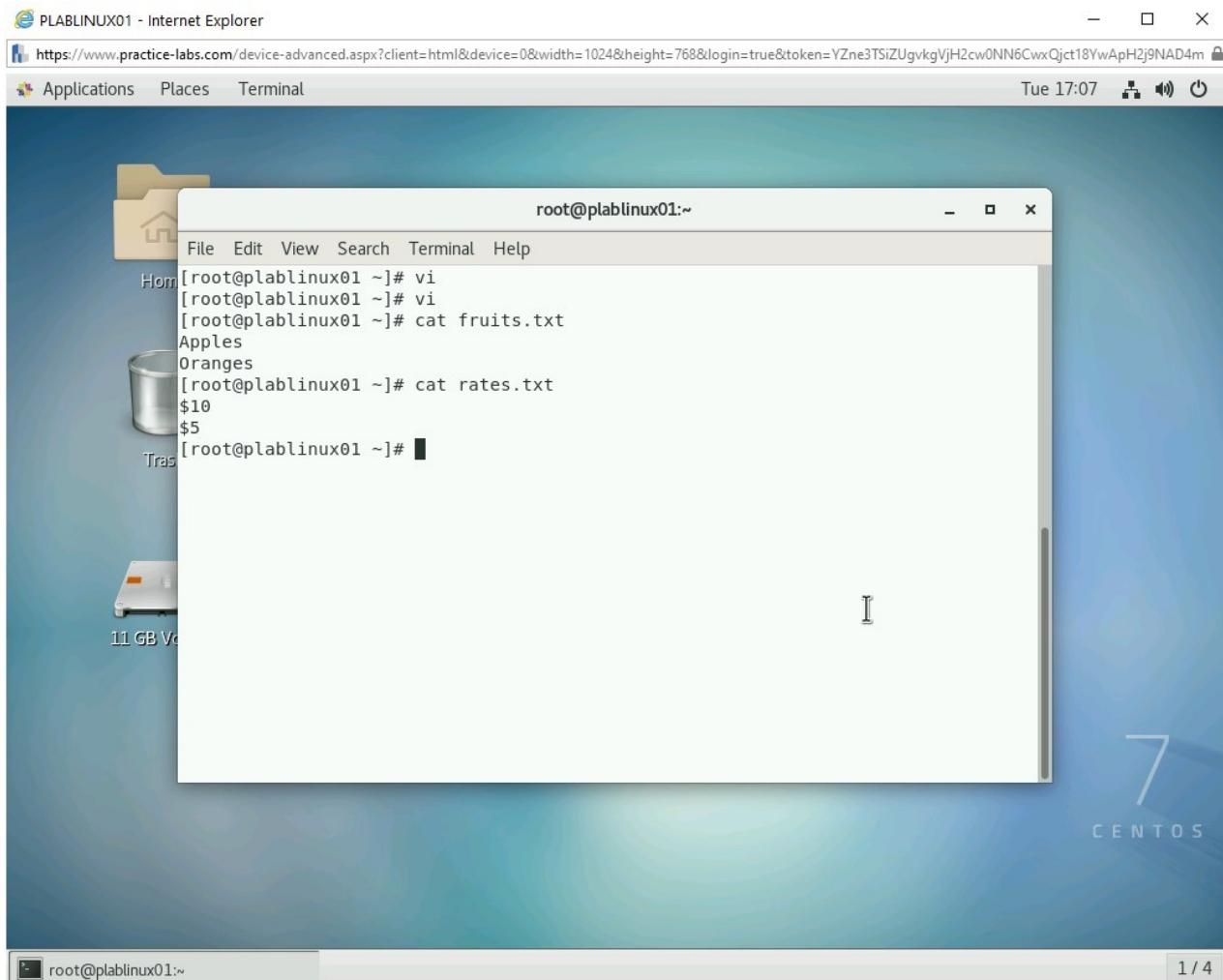


Figure 1.21 Screenshot of PLABLINUX01: Viewing the rates.txt file.

Step 22

You will use the join command on both the files to display the joined output.

To view the joined output, type the following command:

```
join -j2 rates.txt fruits.txt
```

Press **Enter**.

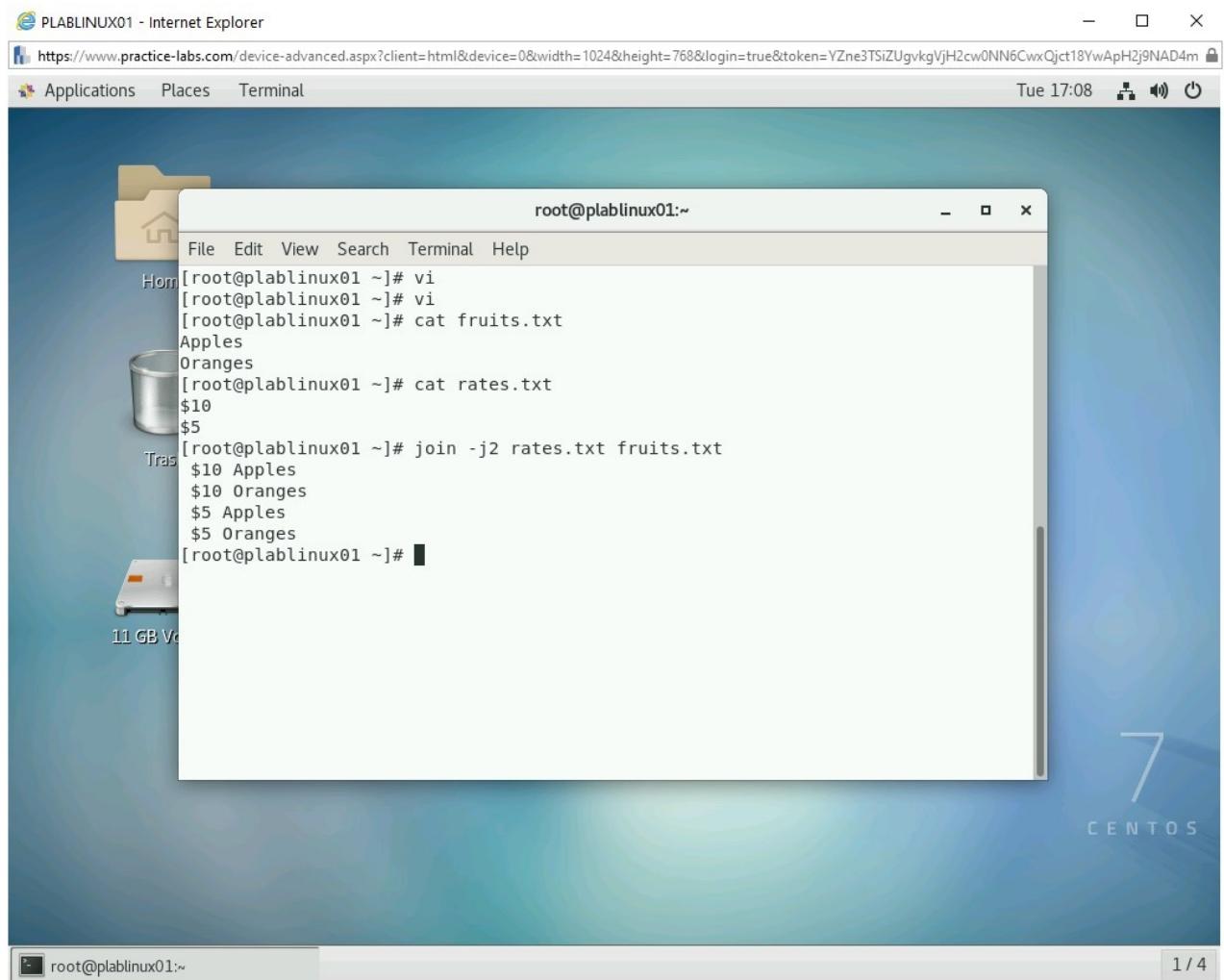


Figure 1.22 Screenshot of PLABLINUX01: Viewing the joined output of two different files.

Step 23

Clear the screen by entering the following command:

```
clear
```

You can number the lines of a file using the `nl` command. The `nl` command displays the number of lines with text.

Using the `-ba` parameter, you can number all lines in a file, even if they don't include text.

```
nl -ba /etc/yum.conf
```

Press **Enter**.

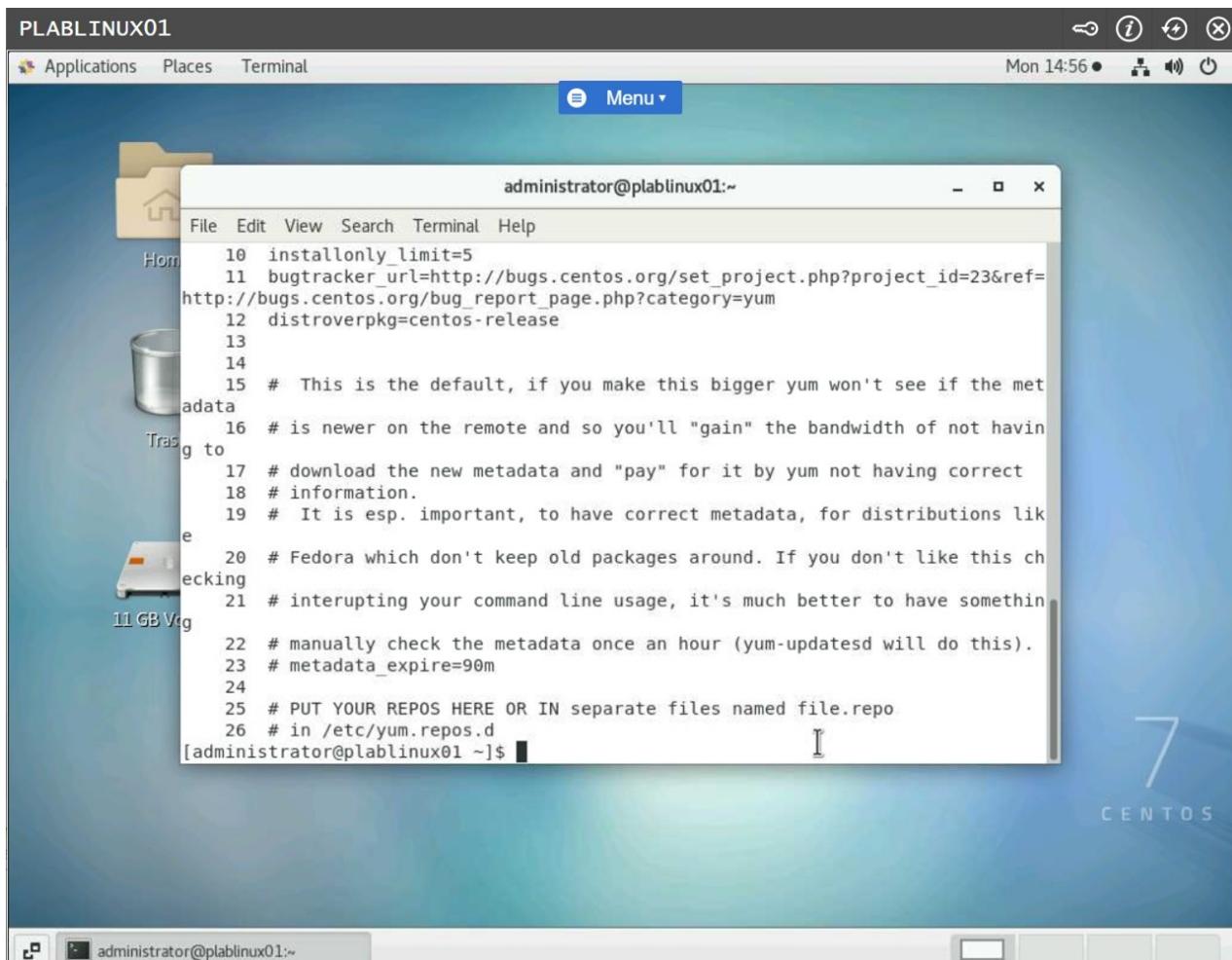


Figure 1.23 Screenshot of PLABLINUX01: Numbering the lines using the `nl` command.

Step 24

Clear the screen by entering the following command:

```
clear
```

Using the `od` command, you can view the file in different number formats, namely octal, hexadecimal, or any other. The `-t` parameter is used to specify the type of output. The `c` parameter displays the ASCII characters. type the following command:

```
od -t c fruits.txt
```

Press Enter.

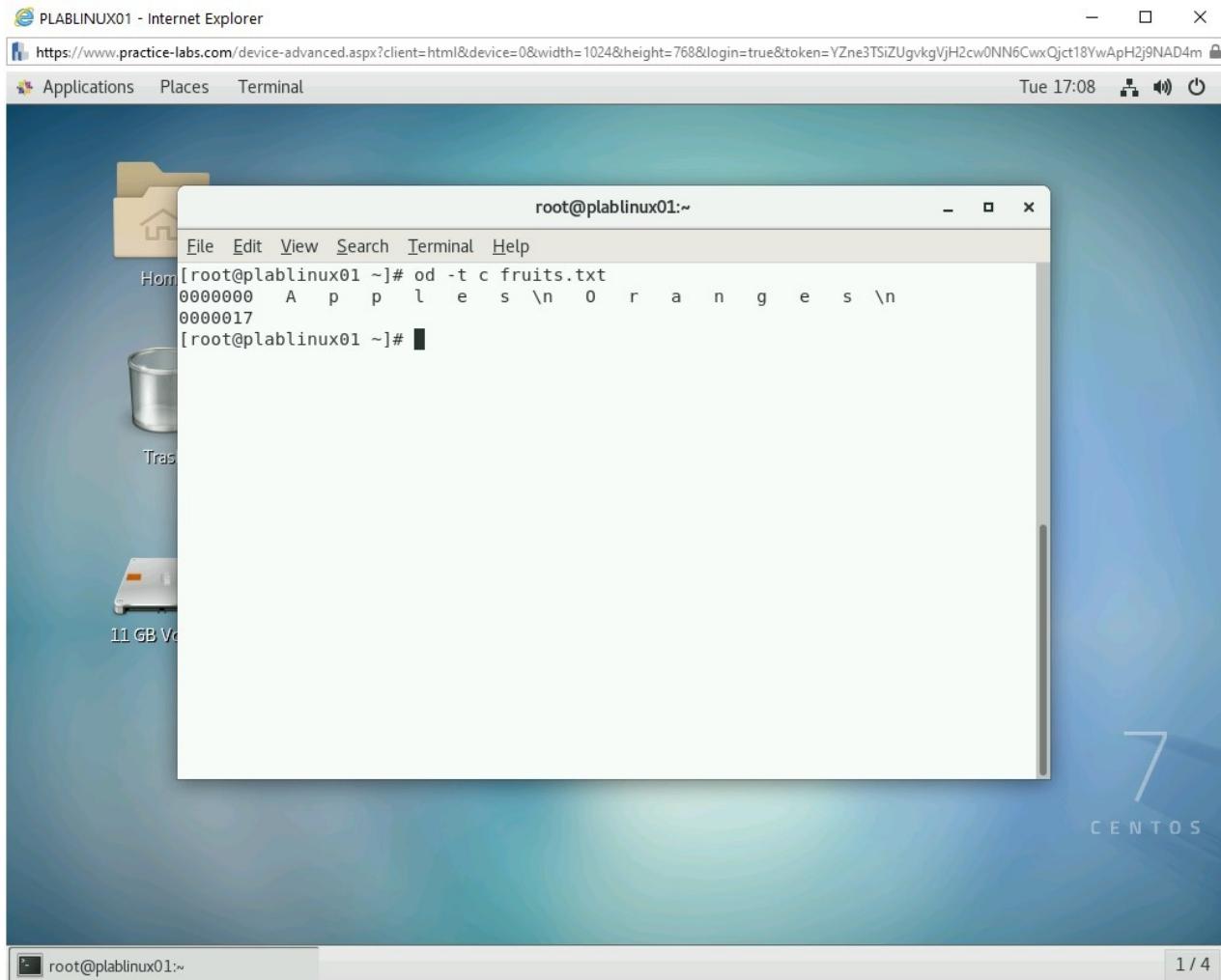


Figure 1.24 Screenshot of PLABLINUX01: Viewing the file in different number formats.

Step 25

Clear the screen by entering the following command:

```
clear
```

Using the paste command, you can paste the output from multiple files into a vertical format. Type the following command:

```
paste rates.txt fruits.txt
```

Press **Enter**.

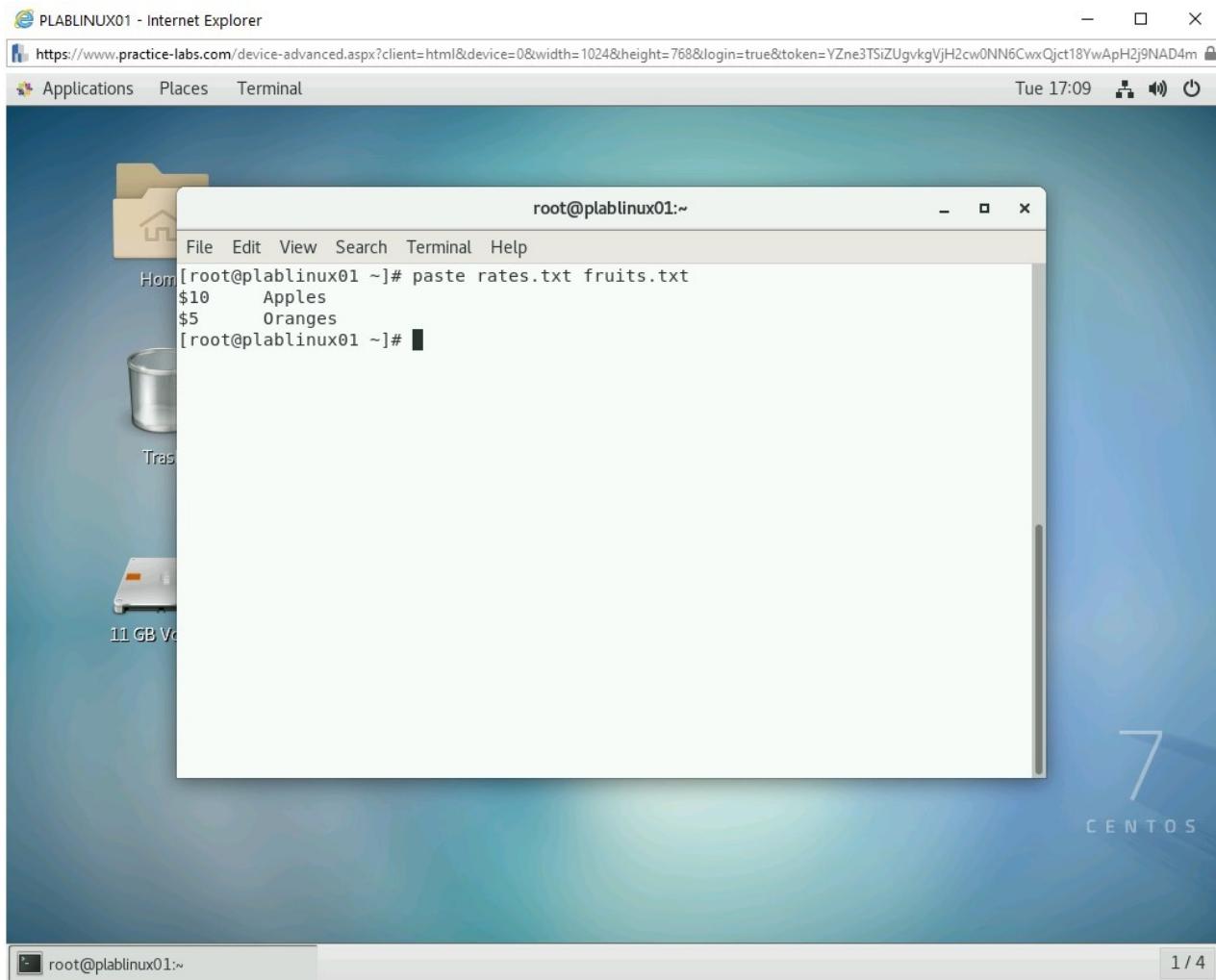


Figure 1.25 Screenshot of PLABLINUX01: Pasting the output from multiple files into a vertical format.

Step 26

The **-s** parameter in the `paste` command enables you to re-orient the output of the command into a horizontal format rather than a vertical one. The **-s** parameter displays the output of multiple files into a separate-line-for-each file format.

To view such an output, type the following command:

```
paste -s fruits.txt rates.txt
```

Press **Enter**.

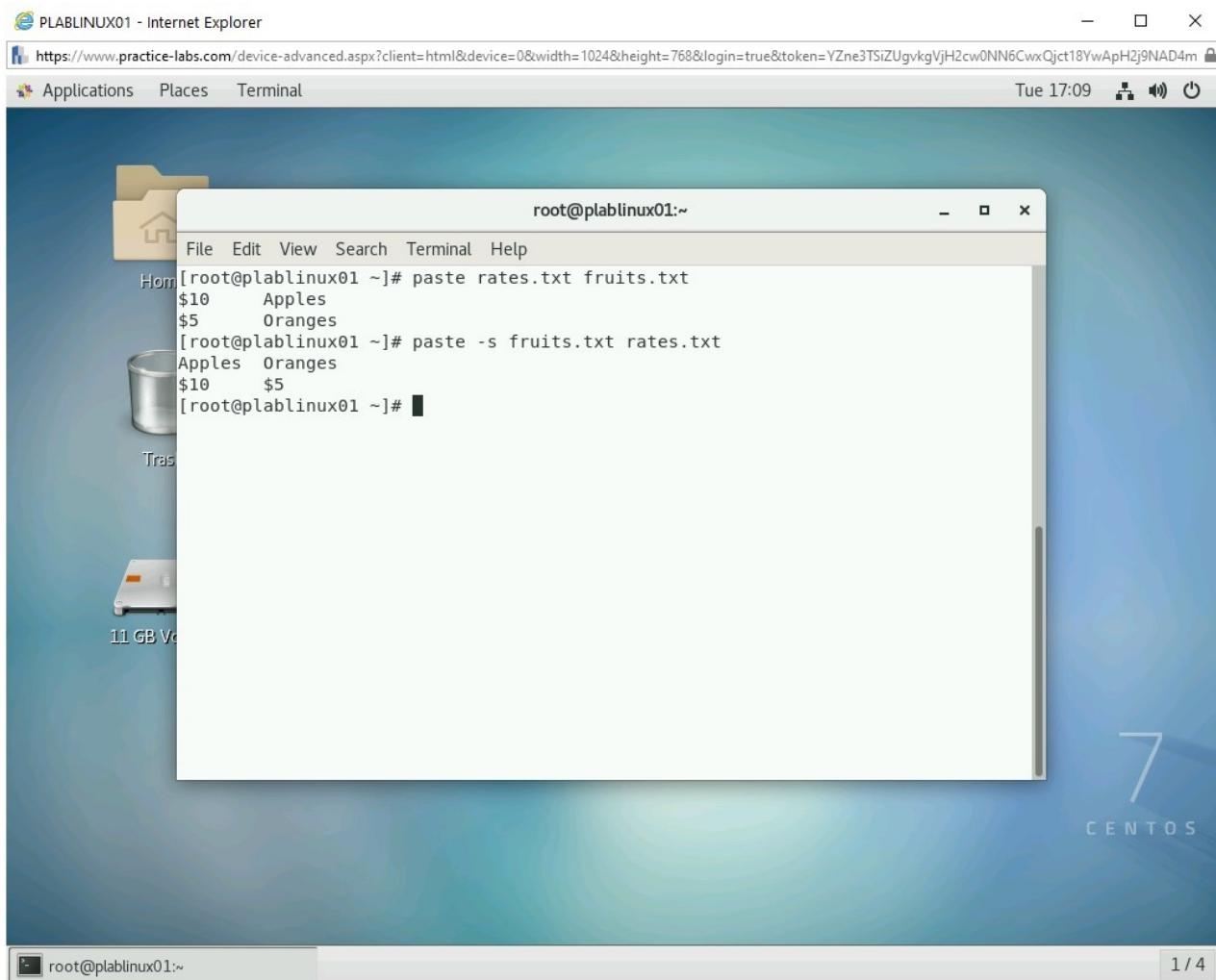


Figure 1.26 Screenshot of PLABLINUX01: Displaying the output of multiple files into a separate-line-for -each file format.

Step 27

Clear the screen by entering the following command:

```
clear
```

Using the pr command, you can convert a text file into a more appropriately formatted file with the headers, pagination, and column fills. With the -d parameter, you can add double space between lines. Type the following command:

```
pr -d /etc/yum.conf
```

Press **Enter**.

Notice the change in the line spacing of the output file. In addition, you can scroll down to see the date and page number added at the beginning of the display.

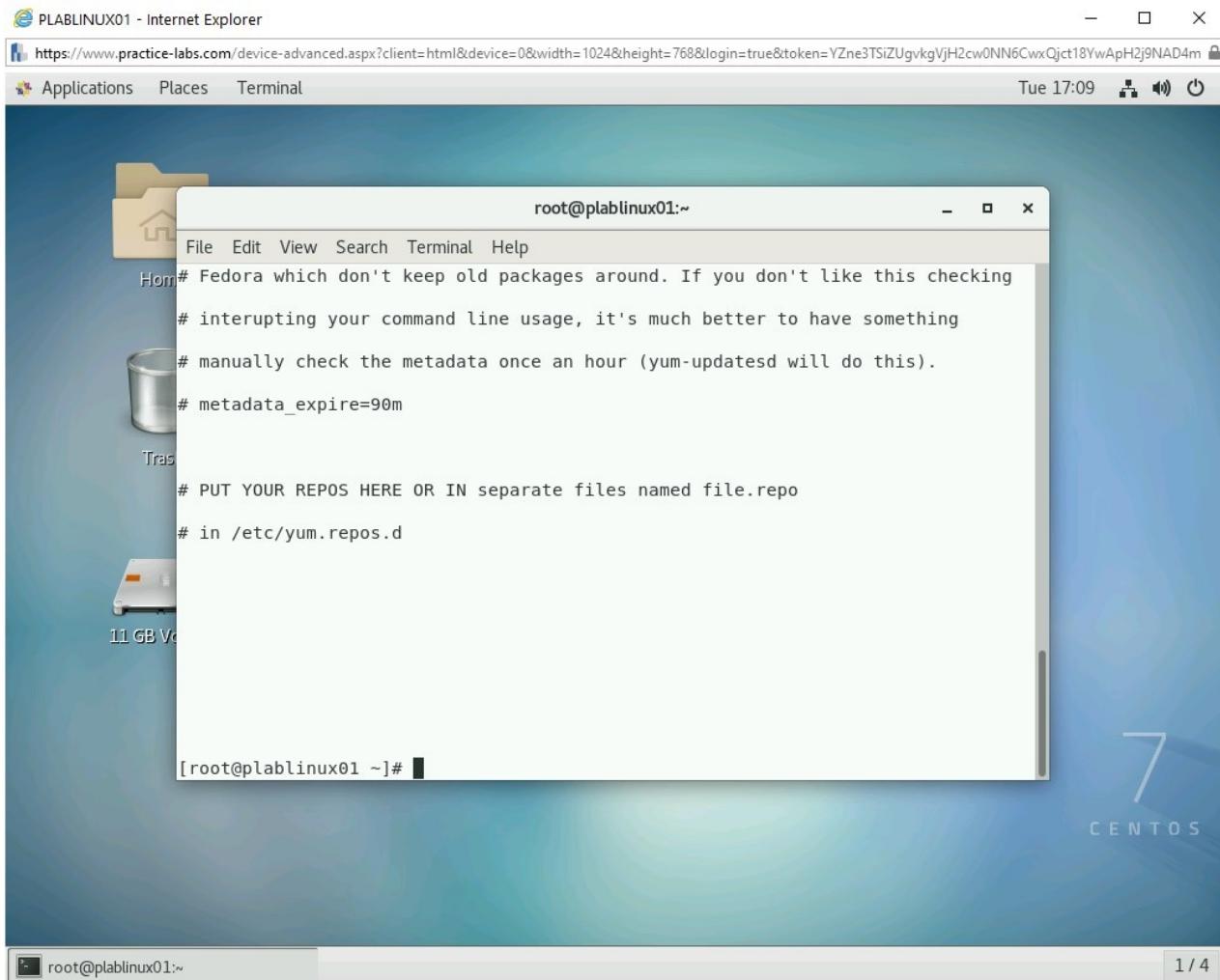


Figure 1.27 Screenshot of PLABLINUX01: Converting a text file into a more appropriately formatted file with the headers, pagination, and column fills.

Step 28

Clear the screen by entering the following command:

```
clear
```

Using the sort command, you can sort out standard input to standard output. This command helps you rearrange the lines in a text file so that they are sorted in different ways, such as numerically and alphabetically. Let's use this command in conjunction with the ps command. The -n parameter sorts the data numerically in ascending

manner. The **-k** parameter sorts on the key starting POS1. Type the following command:

```
ps aux | sort -k 6 -n | more
```

Press **Enter**.

Note: To finish the query please press space until the input bar is visible again or press **crtl+c**.

Note that the values in the second column are sorted out in ascending manner.

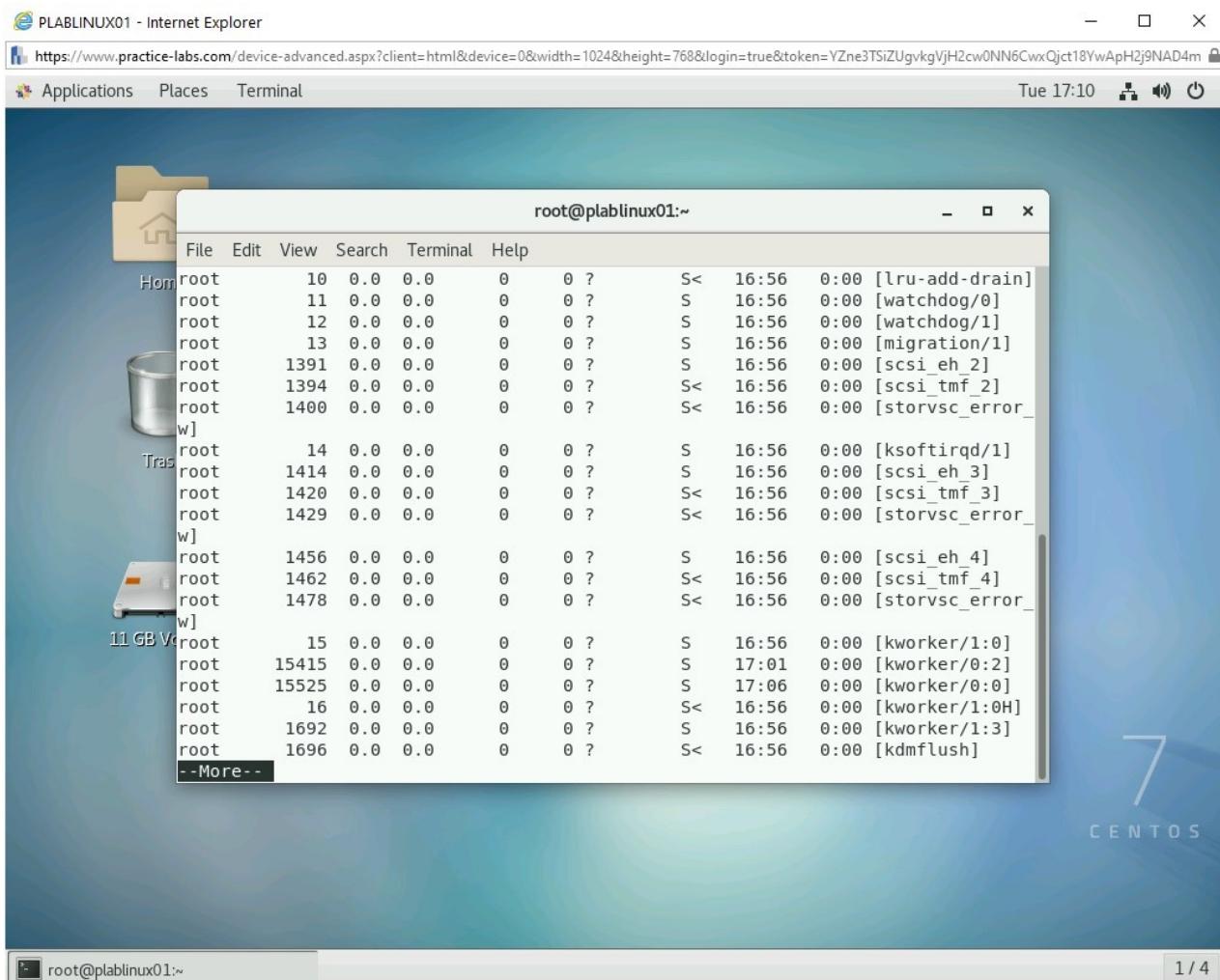


Figure 1.28 Screenshot of PLABLINUX01: Sorting out standard input to standard output.

Step 29

Clear the screen by entering the following command:

```
clear
```

Using the split command, you can divide the contents of a file into multiple files. You need to specify the number of lines that you want to divide into multiple files. You also provide the name of the file as the standard output. If the output has to be more than one file, then the output file is suffixed with aa, ab and so on.

Let's make a copy of **/etc/yum.conf** file and split the copied file. Type the following command:

```
cp /etc/yum.conf yum.conf
```

Press **Enter**.

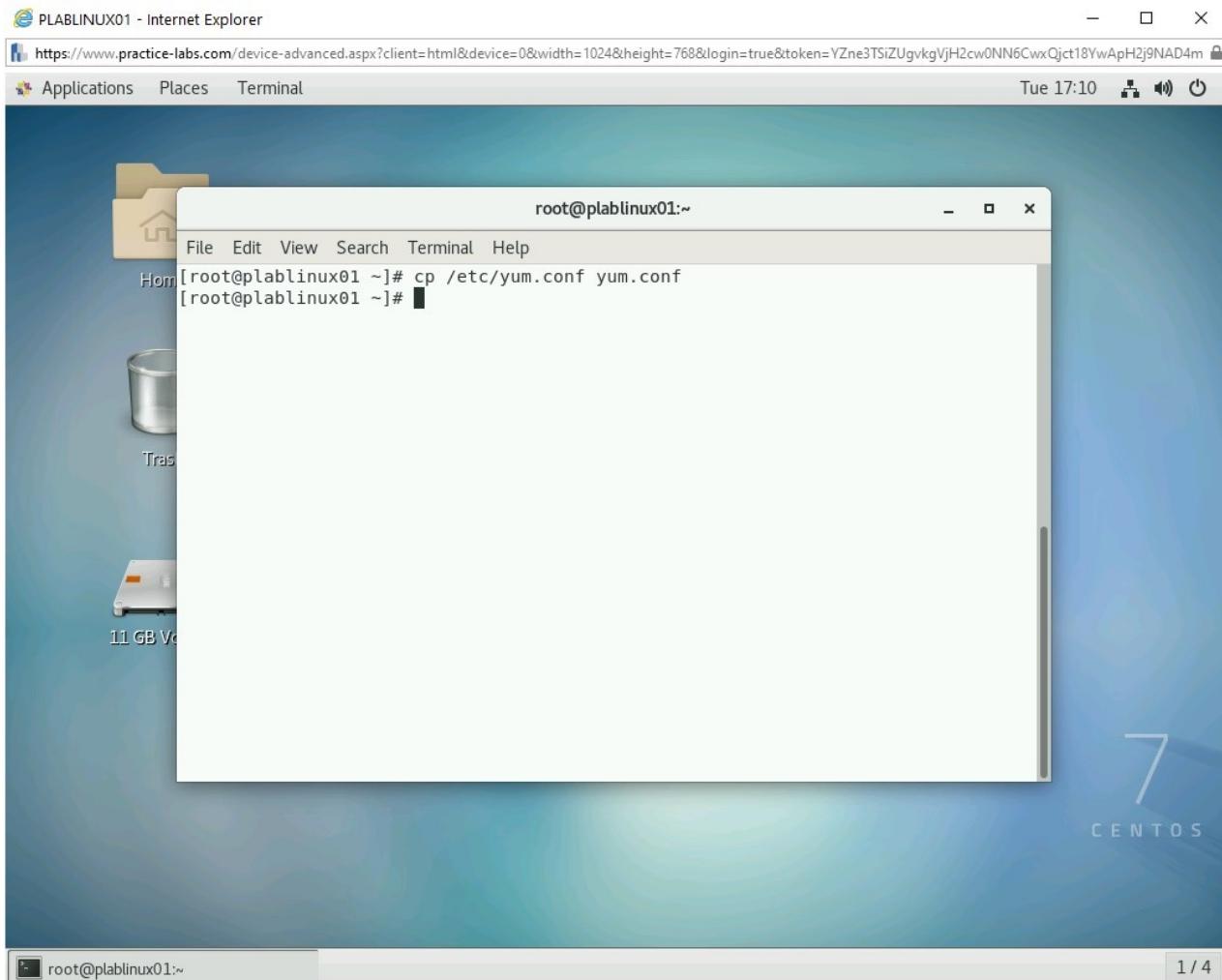


Figure 1.29 Screenshot of PLABLINUX01: Copying the /etc/yum.conf file as yum.conf.

Step 30

Type the following command:

```
split -2 yum.conf splityum
```

Press **Enter**.

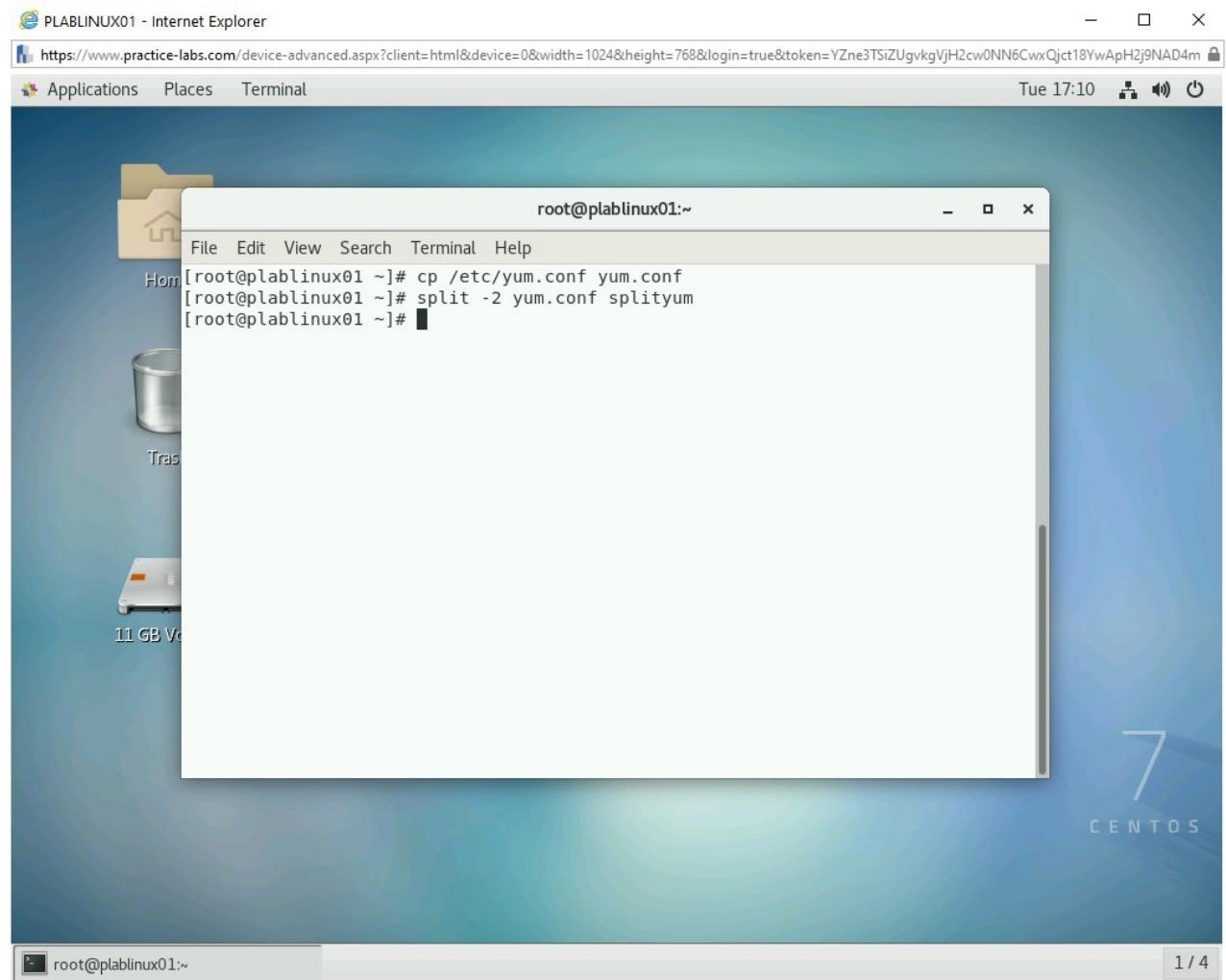


Figure 1.30 Screenshot of PLABLINUX01: Splitting the yum.conf file into splityum files.

Step 31

Type the following command to view the output:

```
ls
```

Press **Enter**.

Note that a number of **splityum** files have been created. In addition, notice the suffixes **aa**, **ab**, etc. used to denote the various portions of the **splityum** file.

Note that it will continue to create files with two lines till the time all lines in the **/etc/yum.conf** are consumed. The output shows multiple files with suffixes, starting with **aa**, **ab**, **ac**, and so on.

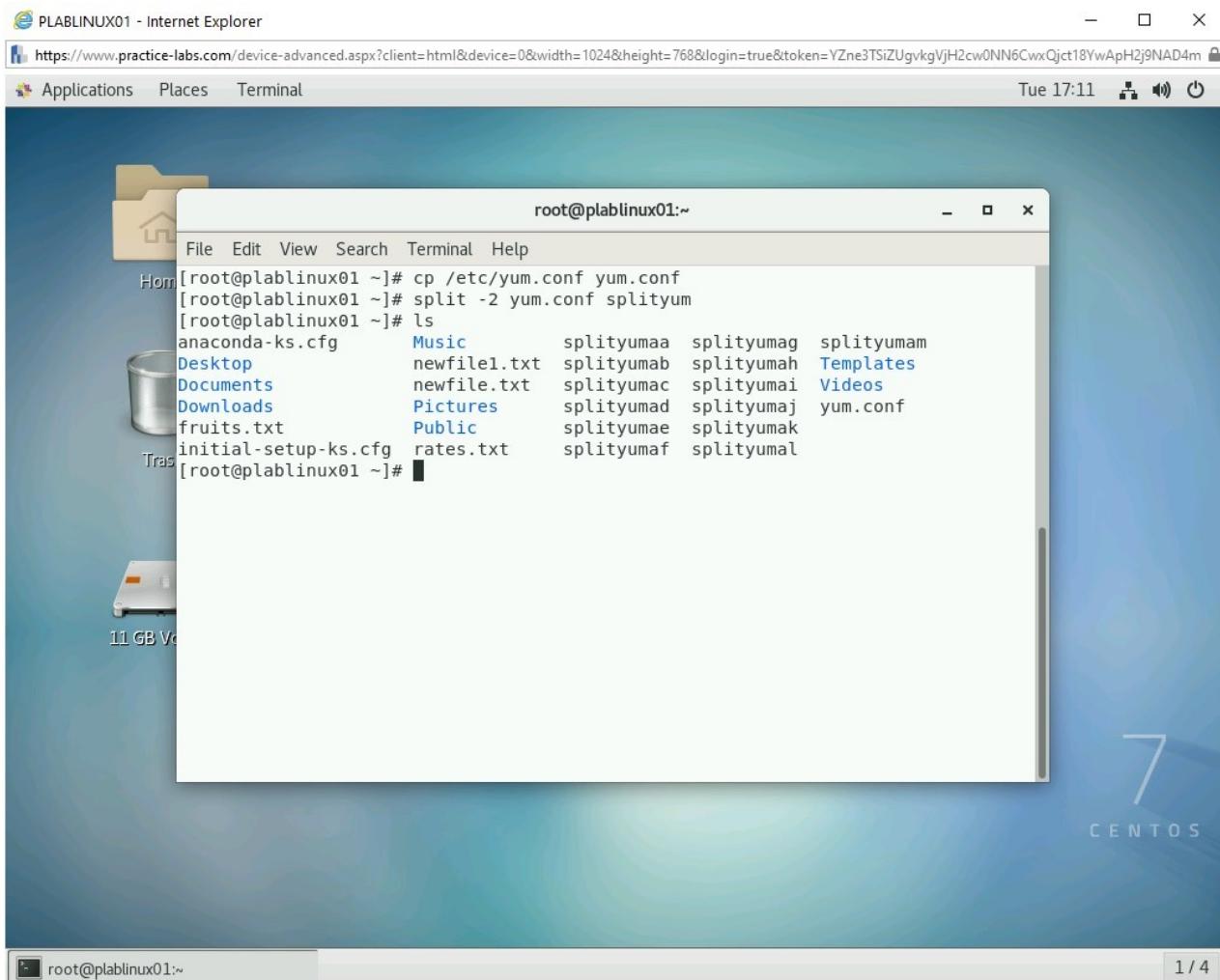


Figure 1.31 Screenshot of PLABLINUX01: Listing the splityum files.

Step 32

Clear the screen by entering the following command:

clear

You can view the last few lines of a file using the tail command. The -n parameter defines the number of lines that you want to see from the end of the file. Type the following command:

```
tail -n10 yum.conf
```

Press **Enter**.

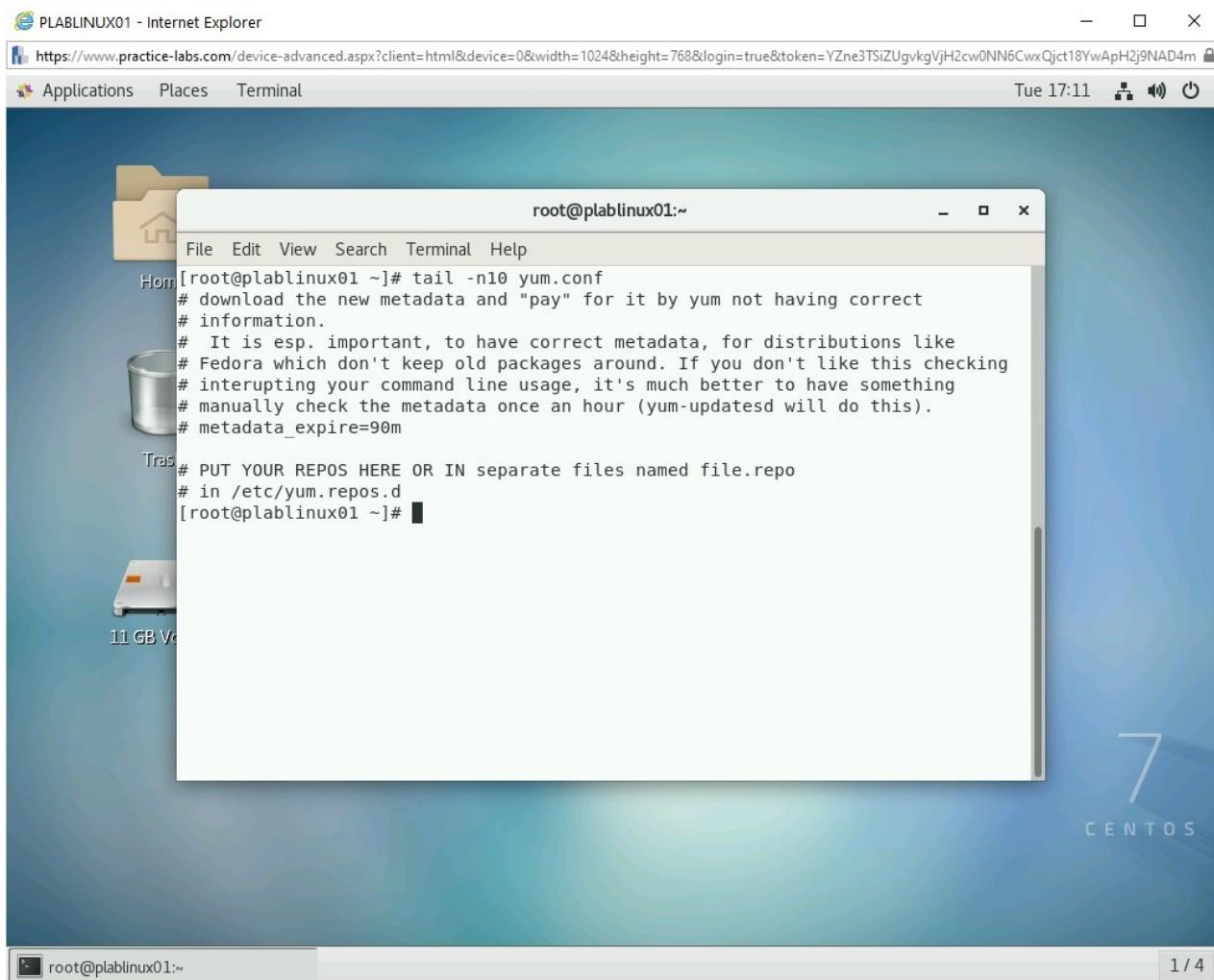


Figure 1.32 Screenshot of PLABLINUX01: Displaying the last few lines of a file using the tail command.

Step 33

Clear the screen by entering the following command:

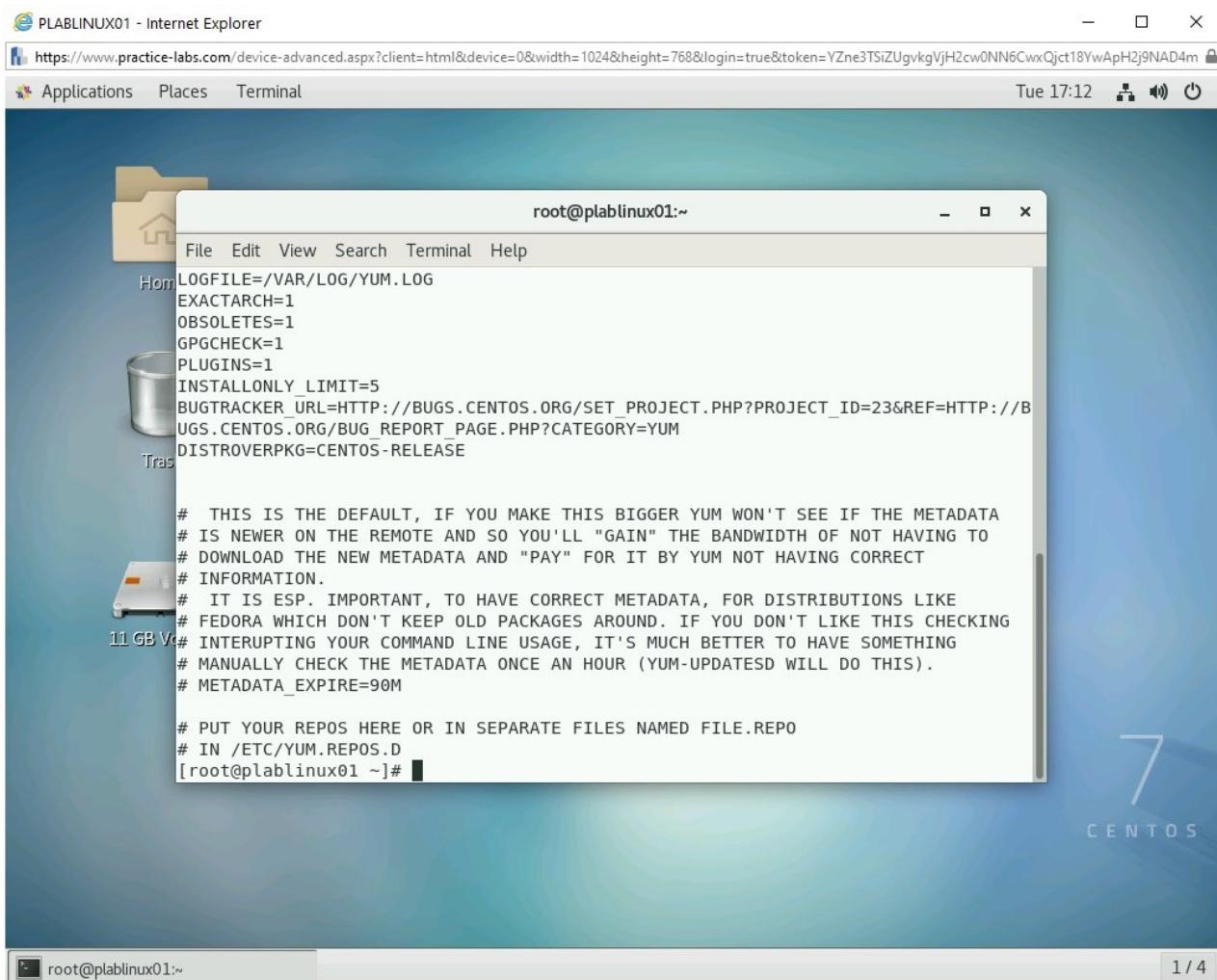
```
clear
```

The tr command allows you to translate the format of characters of the source file into a specified format of characters in the destination file. You can use the command in conjunction with the cat command to view the output. Type the following command:

```
cat yum.conf | tr a-z A-Z
```

Press **Enter**.

Note that, as specified in the command, all the lower-case **a** to **z** characters are translated to upper case.



The screenshot shows a terminal window titled "root@plablinux01:~" running on a CentOS 7 desktop. The window displays the output of the command "cat yum.conf | tr a-z A-Z". The output shows various configuration parameters for the yum package manager, such as LOGFILE, EXACTARCH, OBSOLETES, GPGCHECK, PLUGINS, INSTALLONLY_LIMIT, BUGTRACKER_URL, and DISTROVERPKG. It also includes a note about metadata expiration and repository configuration. The desktop background features the classic CentOS 7 wallpaper with the number 7 and the word CENTOS.

```
root@plablinux01:~  
LogFile=/var/log/yum.log  
exactarch=1  
obsoletes=1  
gpgcheck=1  
plugins=1  
installonly_limit=5  
bugtracker_url=http://bugs.centos.org/set_project.php?project_id=23&ref=http://bugs.centos.org/bug_report_page.php?category=yum  
distroverpkg=centos-release  
  
# THIS IS THE DEFAULT, IF YOU MAKE THIS BIGGER YUM WON'T SEE IF THE METADATA  
# IS NEWER ON THE REMOTE AND SO YOU'LL "GAIN" THE BANDWIDTH OF NOT HAVING TO  
# DOWNLOAD THE NEW METADATA AND "PAY" FOR IT BY YUM NOT HAVING CORRECT  
# INFORMATION.  
# IT IS ESP. IMPORTANT, TO HAVE CORRECT METADATA, FOR DISTRIBUTIONS LIKE  
# FEDORA WHICH DON'T KEEP OLD PACKAGES AROUND. IF YOU DON'T LIKE THIS CHECKING  
# INTERRUPTING YOUR COMMAND LINE USAGE, IT'S MUCH BETTER TO HAVE SOMETHING  
# MANUALLY CHECK THE METADATA ONCE AN HOUR (YUM-UPDATESD WILL DO THIS).  
# metadata_expire=90m  
  
# PUT YOUR REPOS HERE OR IN SEPARATE FILES NAMED FILE.REPO  
# IN /etc/yum.repos.d  
[root@plablinux01 ~]#
```

Figure 1.33 Screenshot of PLABLINUX01: Displaying the lowercase characters into the upper case characters.

Step 34

Clear the screen by entering the following command:

```
clear
```

The uniq command allows to eliminating the adjacent duplicate lines.

```
cat newfile.txt
```

Press **Enter**.

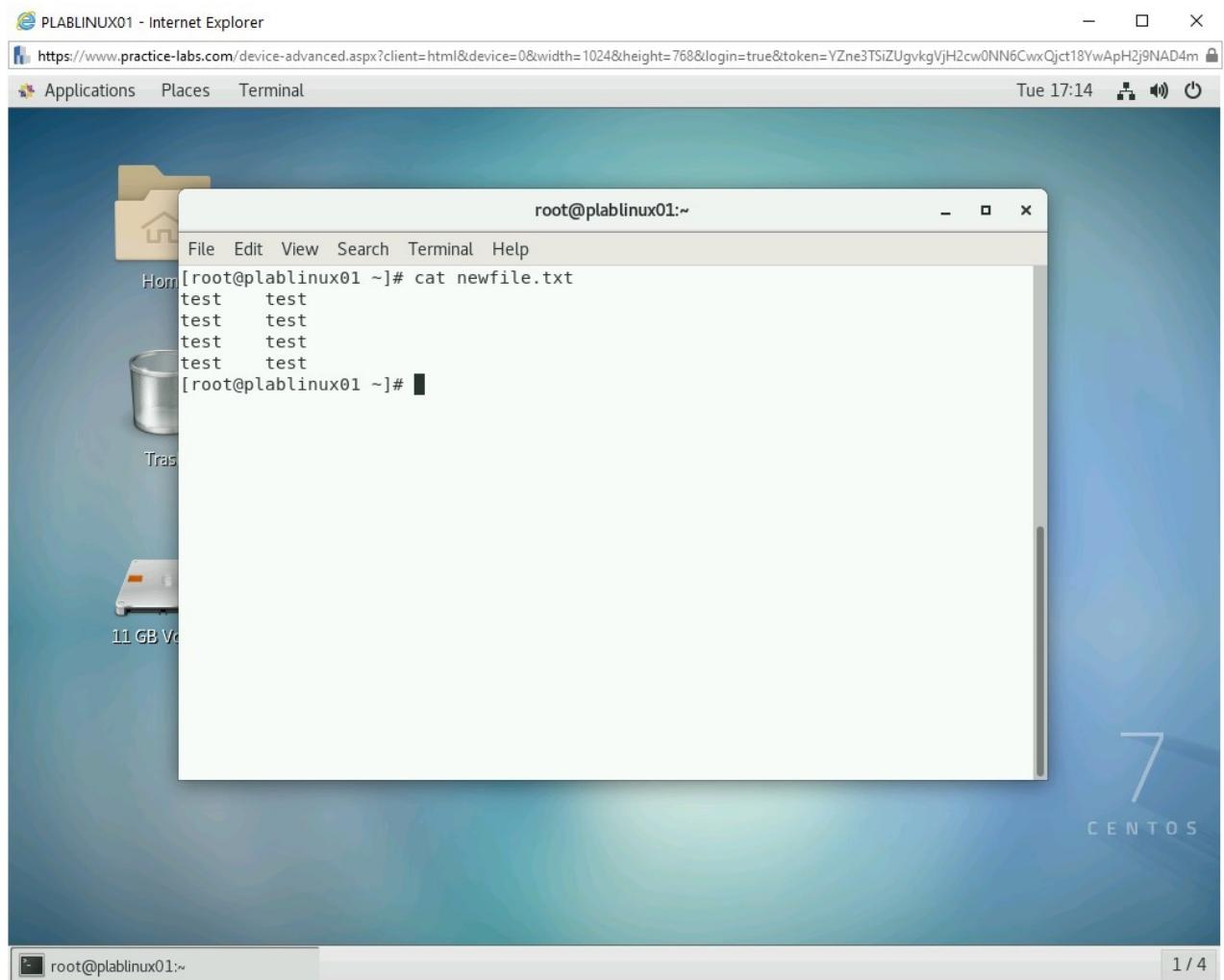


Figure 1.34 Screenshot of PLABLINUX01: Viewing the newfile.txt file.

Step 35

Now, type the following command:

```
uniq newfile.txt
```

Press **Enter**.

Note only unique lines are displayed in the output and duplicate adjacent lines are deleted.

Note: The source file is unchanged. You can verify the same by entering the command: cat newfile.txt

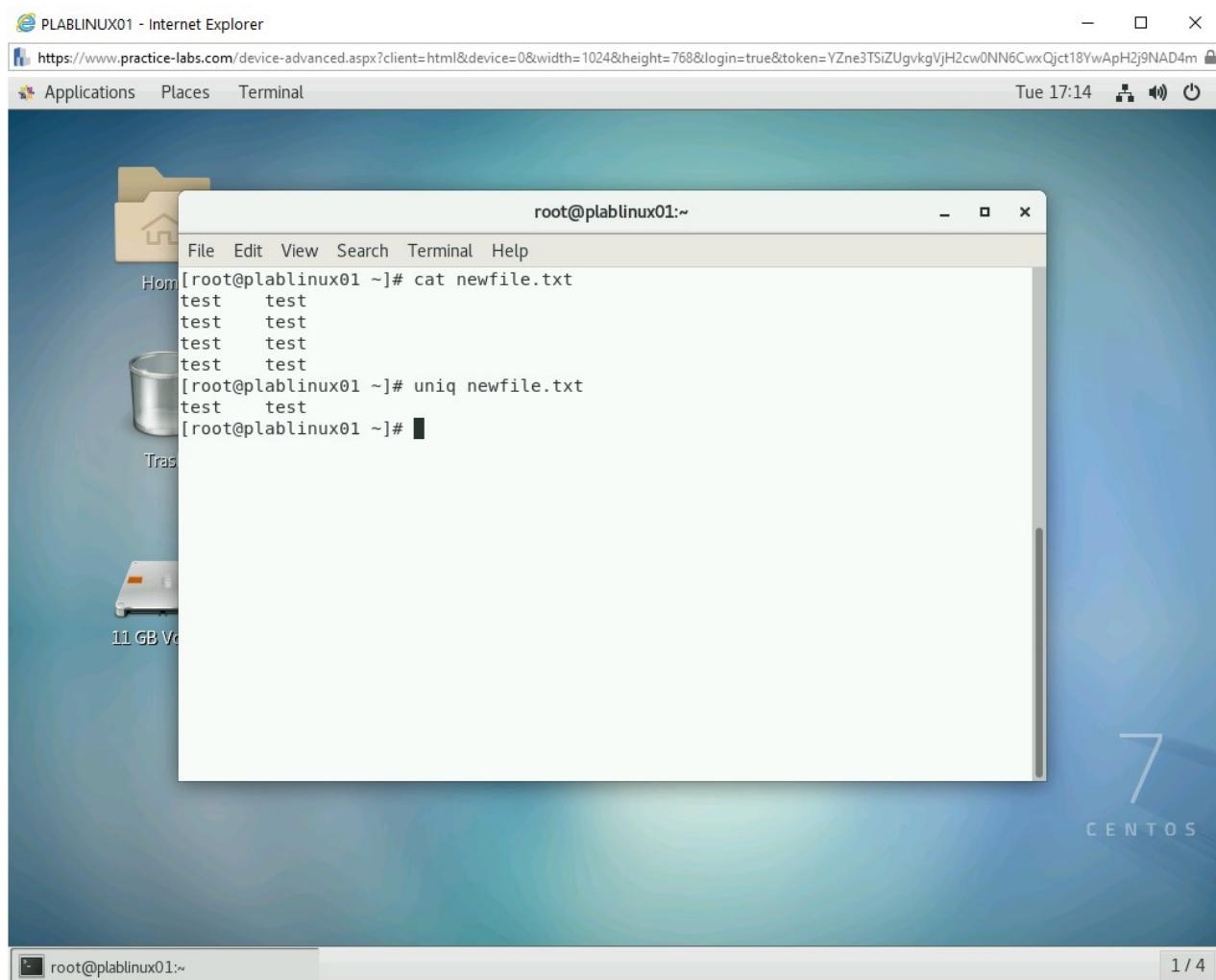


Figure 1.35 Screenshot of PLABLINUX01: Displaying the unique output of the file.

Step 36

Clear the screen by entering the following command:

```
clear
```

The wc command helps you to count the lines, words, and characters of a file.

Type the following command:

```
wc yum.conf
```

Press **Enter**.

Notice the output indicates that **yum.conf** file contains **26** lines made up of **126** words; these words are further composed of **970** characters.

Note: The number of words, lines, and characters may slightly differ in your lab environment.

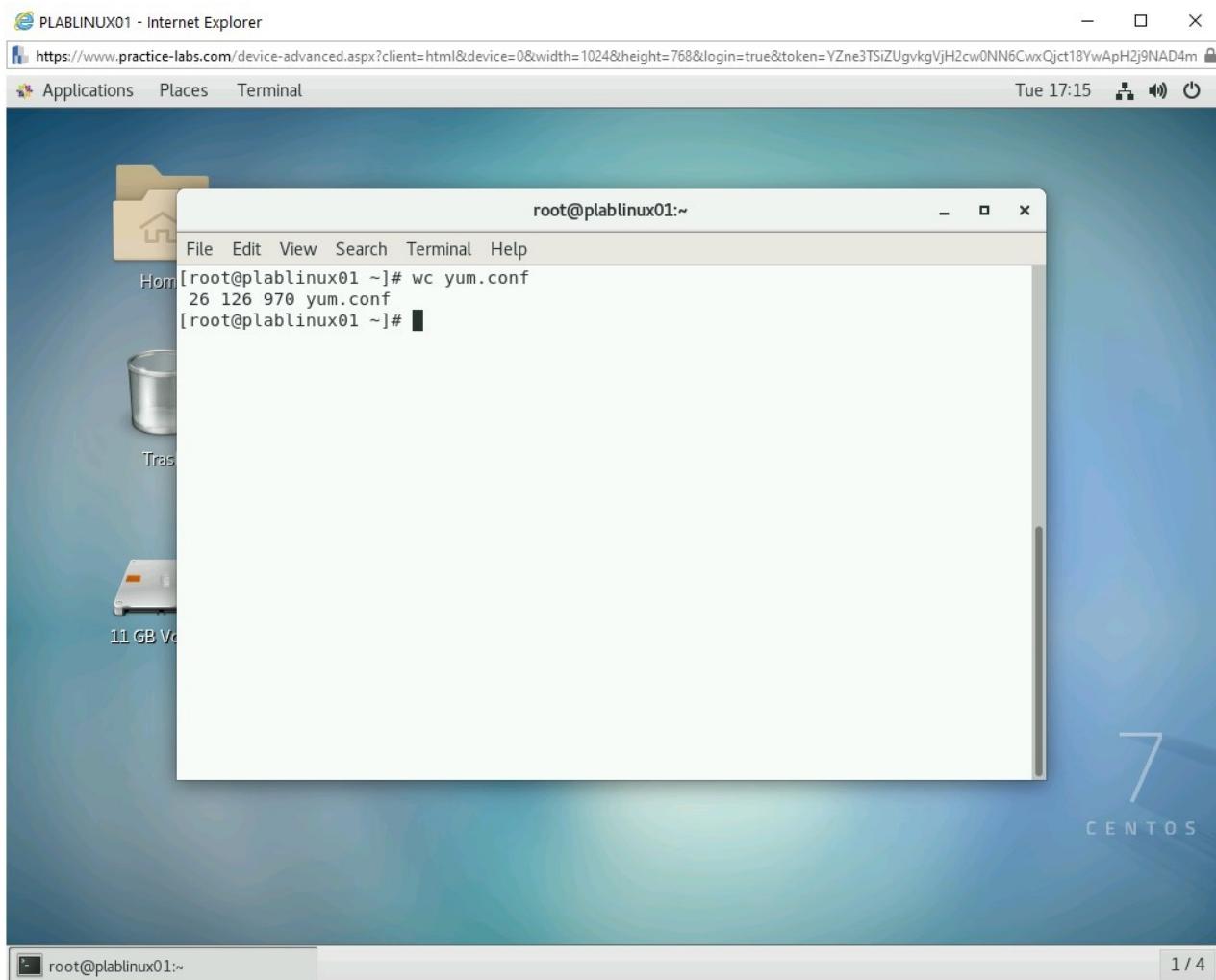


Figure 1.36 Screenshot of PLABLINUX01: Counting the lines, words, and characters of the yum.conf file.

Step 37

You can also view only the number of lines in a file. Type the following command:

```
wc -l yum.conf
```

Press **Enter**.

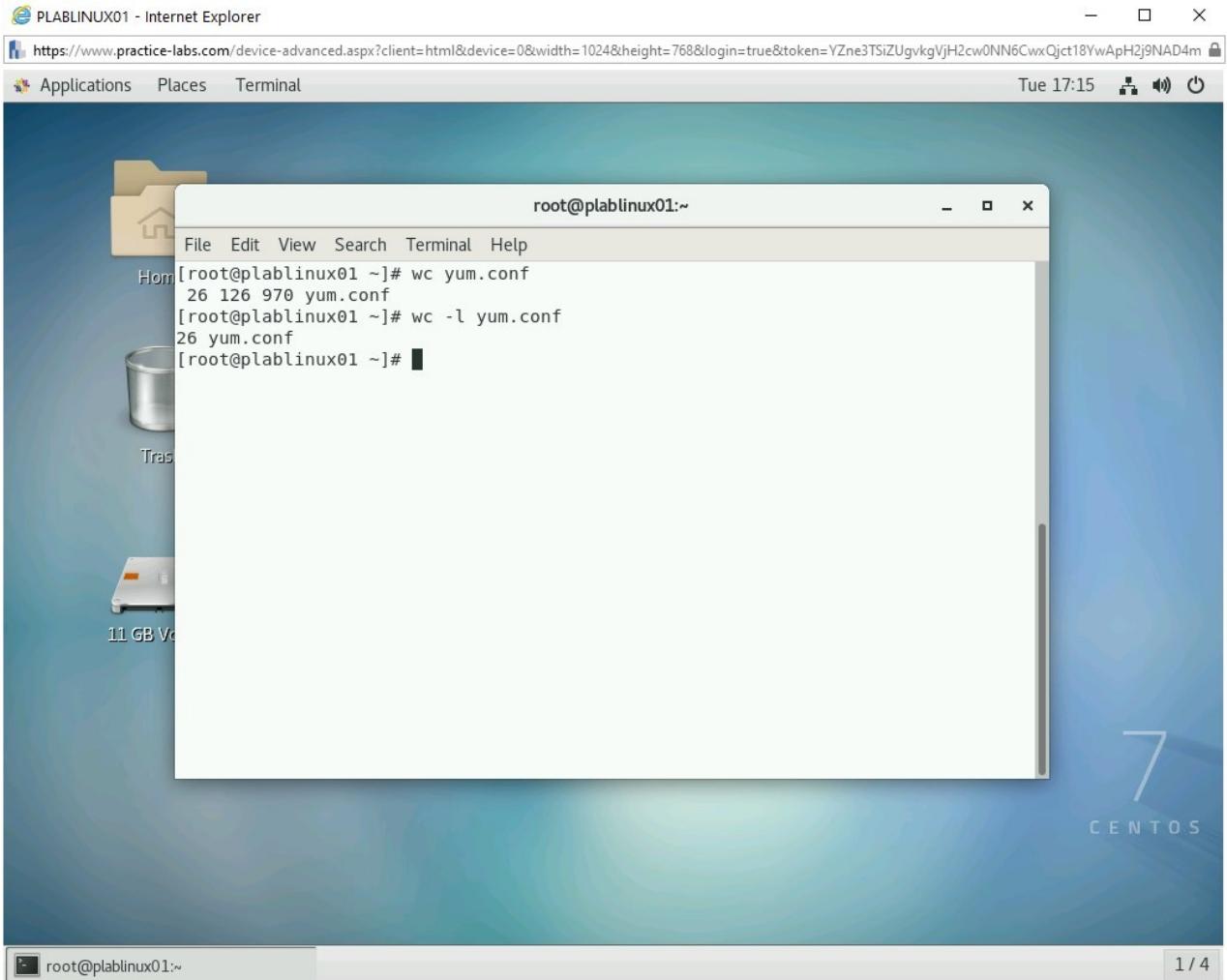


Figure 1.37 Screenshot of PLABLINUX01: Viewing only the number of lines of the yum.conf file.

Step 38

Let's count the number of words in a file. Type the following command:

```
WC -w yum.conf
```

Press **Enter**.

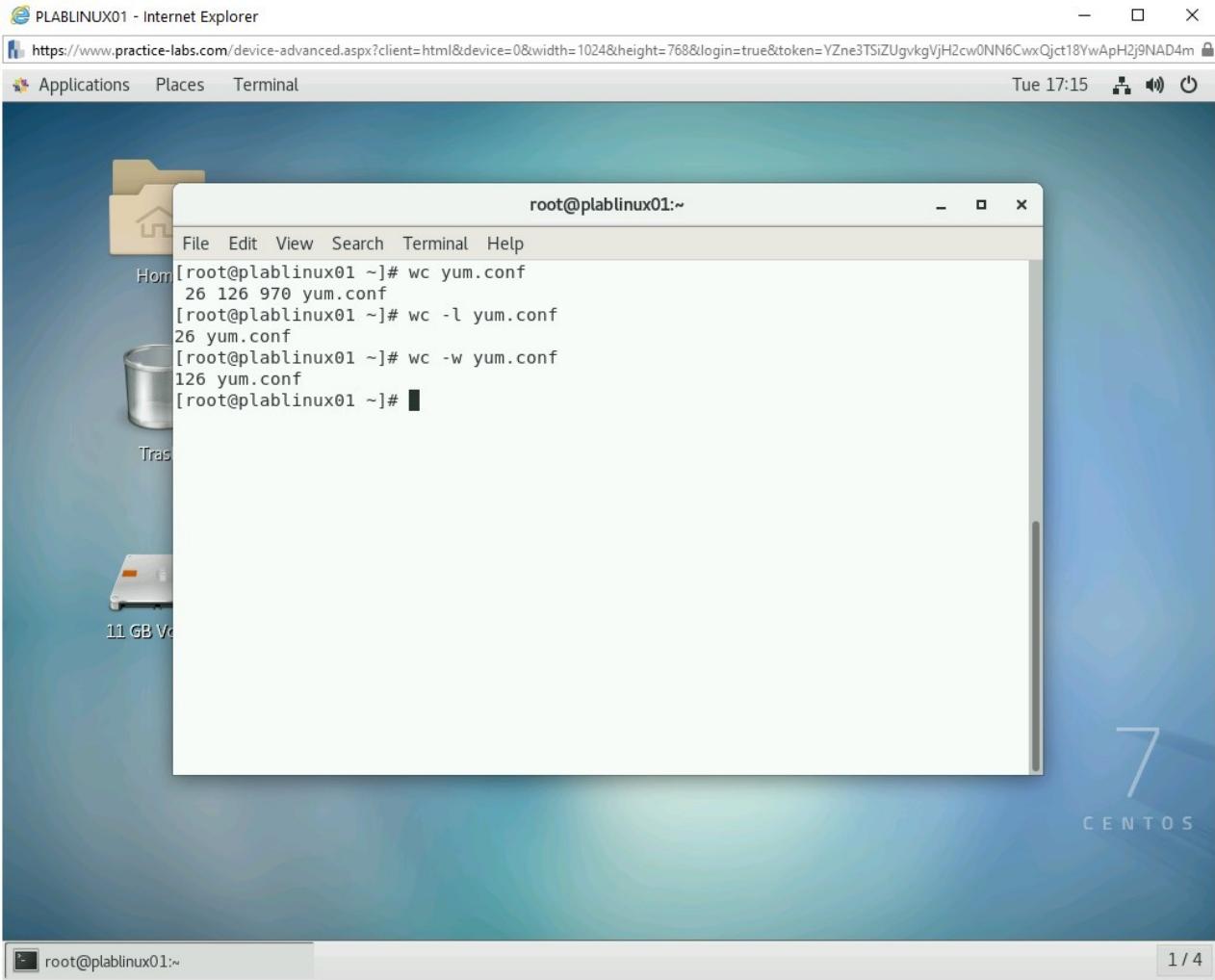


Figure 1.38 Screenshot of PLABLINUX01: Viewing only the number of words in the yum.conf file.

Step 39

Now, let's count the number of characters in a file. Type the following command:

```
WC -c yum.conf
```

Press **Enter**.

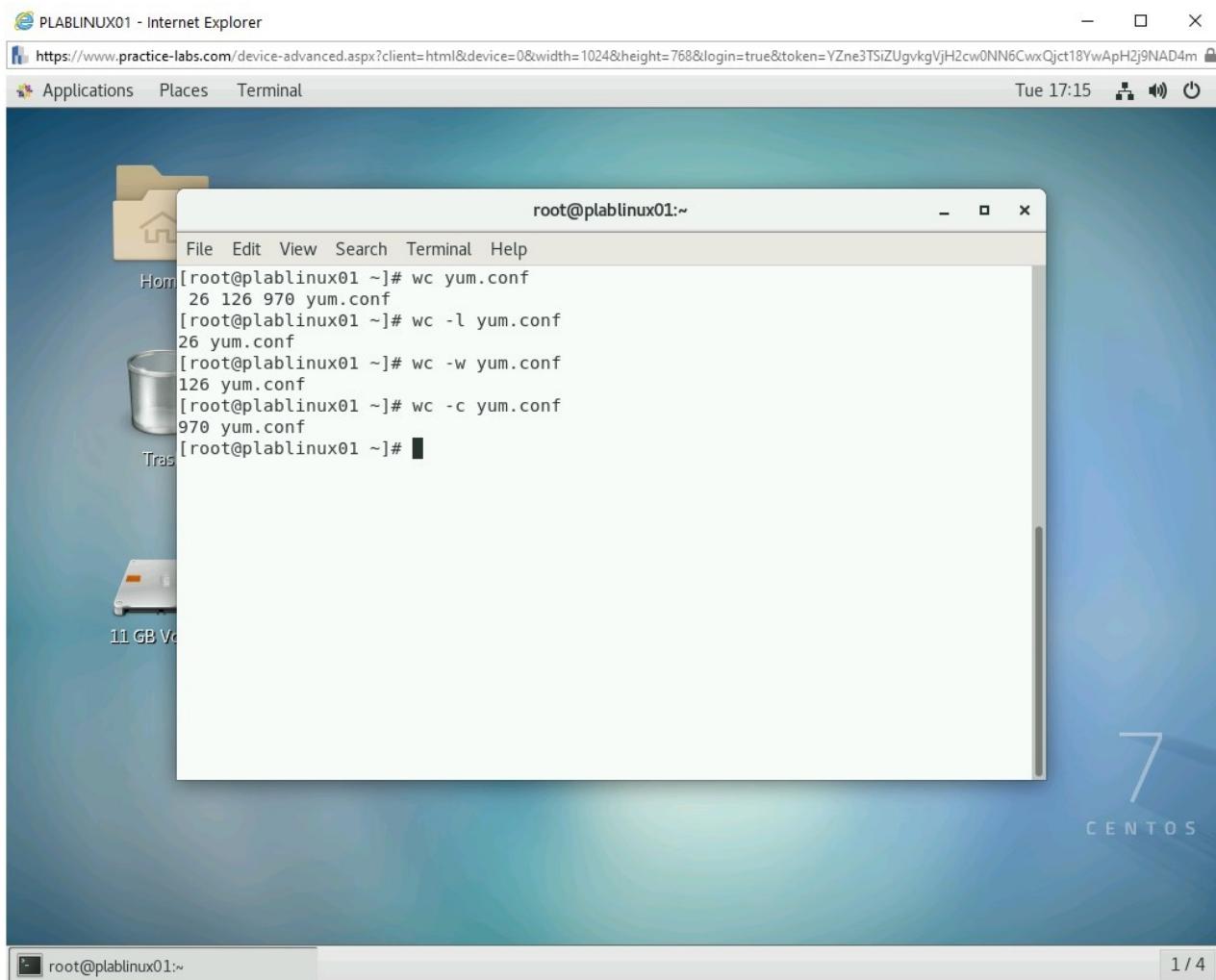


Figure 1.39 Screenshot of PLABLINUX01: Counting the number of characters in the yum.conf file.

Step 40

Clear the screen by entering the following command:

```
clear
```

The sed command allows you to change the text of file with the new text automatically. Let's first view the **newfile.txt**. Type the following command:

```
cat newfile.txt
```

Press **Enter**.

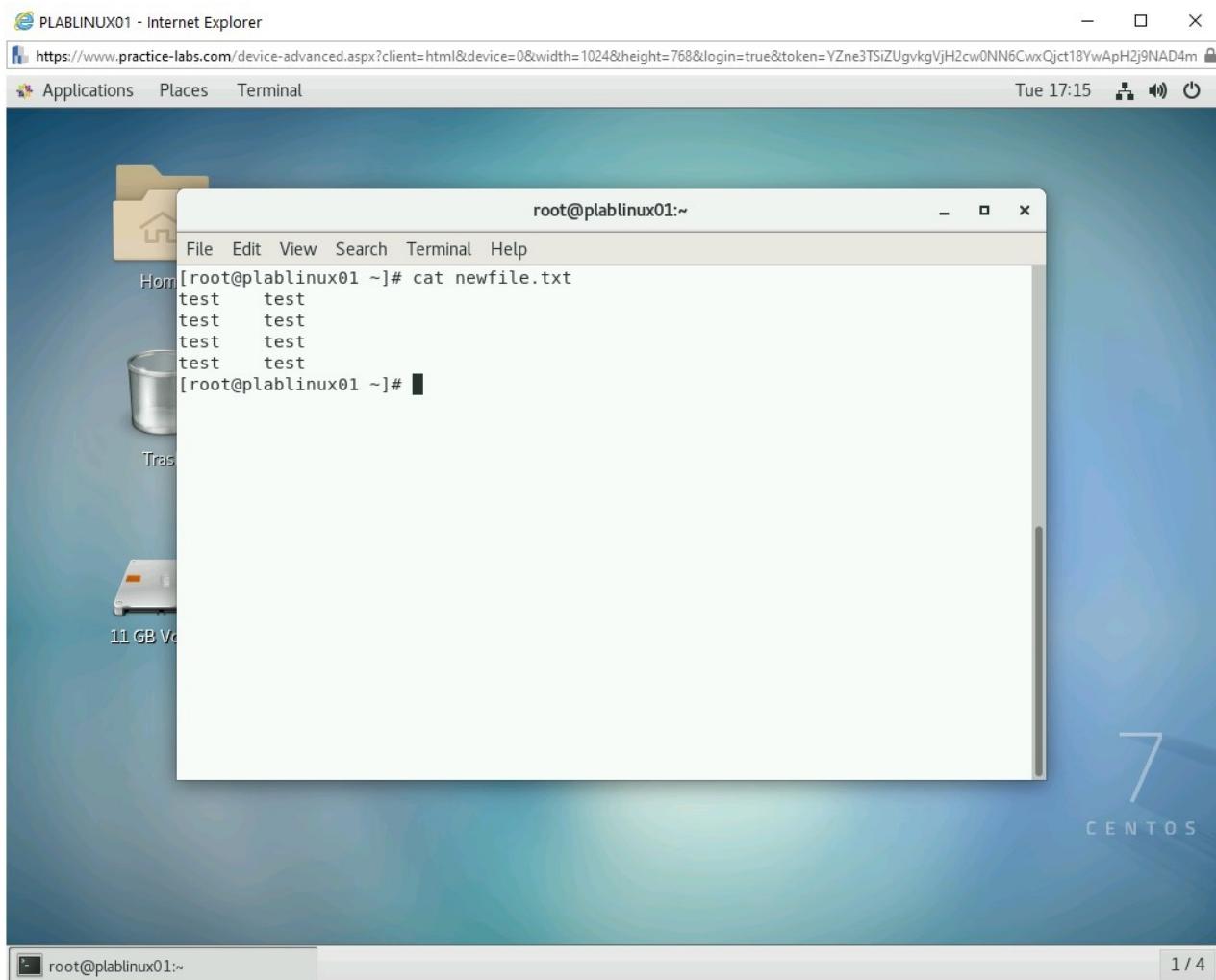


Figure 1.40 Screenshot of PLABLINUX01: Viewing the newfile.txt file.

Step 41

Now, let's replace the first occurrence of test with the word **apple**. Type the following command:

```
sed 's/test/apple/' newfile.txt
```

Press **Enter**.

Note that first occurrence of test in the source file has been replaced with the word apple in the output file.

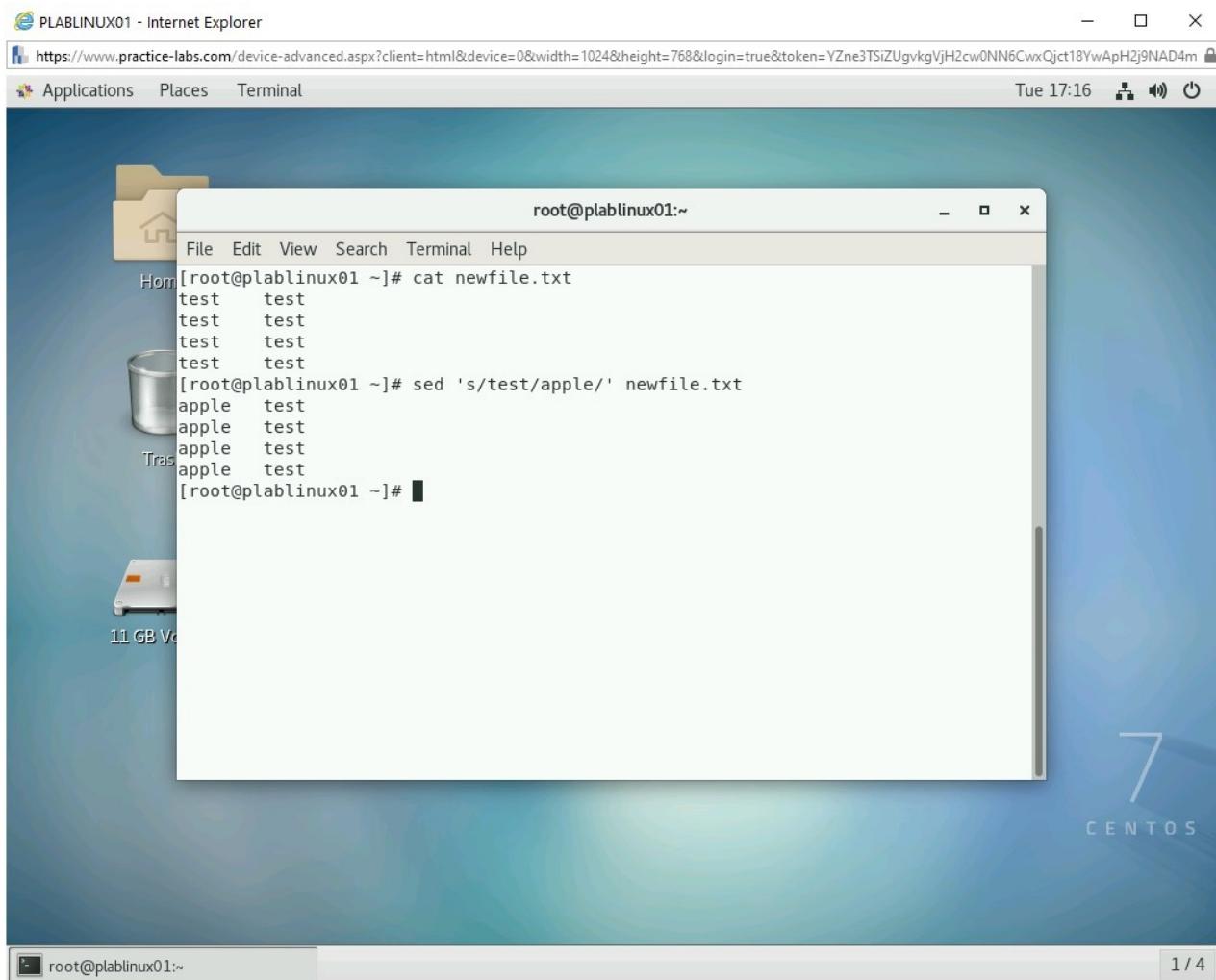


Figure 1.41 Screenshot of PLABLINUX01: Replacing specific characters in a file.

Keep all devices in their current state and proceed to the next exercise.

Review

Well done, you have completed the **Process Text Streams using Filters** Practice Lab.

Summary

You completed the following exercise:

- Exercise 1 - Process Text Streams using Filters

You should now be able to:

- Process Text Files Using Text Utility Filters

Feedback

Shutdown all virtual machines used in this lab. Alternatively, you can log out of the lab platform.