# Linux Command Line

During this lab, you will utilize a bash terminal to enhance your login session by using key components of the command line, identifying system time and date, and working with your previous commands to enable more efficient workflow.

## Objectives

In this lab, you will:

* Run familiarity commands to gain knowledge of your current system and current session
* Search and run previous bash commands

## Exercise 1: Run Familiarity Commands

In this exercise, you must be logged into a Linux machine and have a terminal open. You will run a few commands to gain a base knowledge of the system and session you are using.

### Steps

1. From the terminal, type **whoa** and press the TAB button, taking notice the auto complete feature to display the full command **whoami**.
2. Press ENTER to display your current username.
3. Type **hostname -s** and press ENTER to display shortened version of computer’s host name.
4. Type **uptime -p** and press ENTER to display up time of the system in easily readable format.
5. Sometimes, you really want to know the total count of lines, words, and characters in a file. Type **wc ~/.bash\_history** and press ENTER to see information about current bash history file.

Note

if you don’t see a **.bash\_history** file or receive an error, type **exit** and press ENTER to close the session to the machine. Log in to the machine by pressing centos on the top left and repeat step 5.

[labsuser@centos ~]$ whoami

labsuser

[labsuser@centos ~]$ hostname -s

centos

[labsuser@centos ~]$ uptime -p

up 1 week, 16 hours, 14 minutes

[labsuser@centos ~]$ wc ~/.bash\_history

4 7 48 /home/labsuser/.bash\_history

[labsuser@centos

1. From the terminal, type **who -H -a** and hit ENTER to display information about users that are logged in with some additional information.

[labsuser@centos ~]$ who -H -a

NAME LINE TIME IDLE PID COMMENT EXIT

labsuser + pts/0 2019-08-21 16:21 . 67 (ip-172-17-0-1.us-west-2.compute.internal)

[labsuser@centos ~]$

1. Type **TZ=America/New\_York date** and press ENTER. Then type **TZ=America/Los\_Angeles date**. This allows you to identify the date and time of alternate locations in the world.

[labsuser@centos ~]$ TZ=America/New\_York date

Wed Aug 21 12:28:39 EDT 2019

[labsuser@centos ~]$ TZ=America/Los\_Angeles date

Wed Aug 21 09:28:52 PDT 2019

[labsuser@centos ~]$

Note

If your system time is not set properly, you will receive a time that is incorrect.

1. In some professions, the Julian date is used to conduct business. You can check this by typing **cal -j** in your terminal to see your current month’s version of the Julian date.

[labsuser@centos ~]$ cal -j

August 2019

Sun Mon Tue Wed Thu Fri Sat

213 214 215

216 217 218 219 220 221 222

223 224 225 226 227 228 229

230 231 232 233 234 235 236

237 238 239 240 241 242 243

[labsuser@centos ~]$

1. Typing the **cal -s** or **cal -m** command will display alternate views of the calendar.

[labsuser@centos ~]$ cal -s

August 2019

Su Mo Tu We Th Fr Sa

1 2 3

4 5 6 7 8 9 10

11 12 13 14 15 16 17

18 19 20 21 22 23 24

25 26 27 28 29 30 31

[labsuser@centos ~]$ cal -m

August 2019

Mo Tu We Th Fr Sa Su

1 2 3 4

5 6 7 8 9 10 11

12 13 14 15 16 17 18

19 20 21 22 23 24 25

26 27 28 29 30 31

Note

There are many options to display calendars. Check out the cal man page for details.

1. For your last command, type **id labs*user*** in the terminal and press ENTER to see your unique ID and group information about your specific user.

[labsuser@centos ~]$ id labsuser

uid=102931(labsuser) gid=160616(labsuser) groups=160616(labsuser),10(wheel)

[labsuser@centos ~]$

## Exercise 2: Improve Workflow through bash History and Search

Following on with the previous exercise, you will attempt to ease your overall workload by reusing commands through search techniques, manual visualization of the bash history log, and reuse of the last command.

### Steps

1. Ensure your terminal is open.
2. Start by viewing the current bash history. Type **history** and press ENTER. Using the output, try and match the commands you see to the ones you used in exercise 1.
3. Alternatively, type **cat ~/.bash\_history** into the terminal and press ENTER to view the file that holds the same information.

[labsuser@centos ~]$ history

1 whoami

2 hostname -s

3 uptime -p

4 wc ~/.bash\_history

5 who -H -a

6 TZ=America/New\_York date

7 TZ=America/Los\_Angeles

8 cal -j

9 cal -s

10 cal -m

11 id labsuser

12 history

[labsuser@centos ~]$

1. To search your previous history, you can use the CTRL+R key combination to bring up a reverse history search. In the reverse history search feature of the terminal, type **TZ** and press **TAB**. This will bring up an old use of the **date** command and allow you to edit. Using your arrow buttons, you can now edit the command inline.

Note

This is a history searching feature that allows you to edit the command you search for. You must use **TAB** autocomplete to edit and run the commands.

(reverse-i-search)`TZ': TZ=America/Los\_Angeles date

1. Type **date** into the terminal and press ENTER. Now type !! and press ENTER. This will allow you to rerun the most recent command.

[labsuser@centos ~]$ TZ=America/Los\_Angeles date

Wed Aug 21 10:00:26 PDT 2019

[labsuser@centos ~]$ !!

TZ=America/Los\_Angeles date

Wed Aug 21 10:00:50 PDT 2019

[labsuser@centos ~]$

1. Type **history** and press ENTER. Using the output from the **history** command, type **!119** and press ENTER to rerun the **wc ~/.bash\_history** command on your bash history file.

Note

Replace **119** with the appropriate number in history. Ex: !48 to execute line 48.

[labsuser@centos ~]$ history

1 whoami

2 hostname -s

3 uptime -p

4 wc ~/.bash\_history

5 whoami

6 hostname -s

7 uptime -p

8 wc ~/.bash\_history

9 whoami

10 hostname -s

11 uptime -p

12 wc ~/.bash\_history

13 who -H -a

14 TZ=America/New\_York date

15 TZ=America/Los\_Angeles date

16 cal -j

17 cal -s

18 cal -m

19 id labsuser

20 history

[labsuser@centos ~]$ !12

wc

12 23 157 /home/labsuser/.bash\_history

[labsuser@centos ~]$