# Instructions

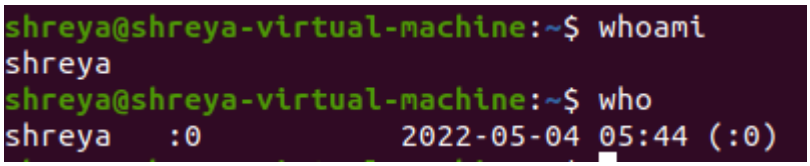
**1. Display the name of the current user using the following command**

**$ whoami**

It will display the user name as which you are logged into the virtual machine.

You can get a list of currently logged in users using the command

**$ who**




**2. Get basic information about the operating system**

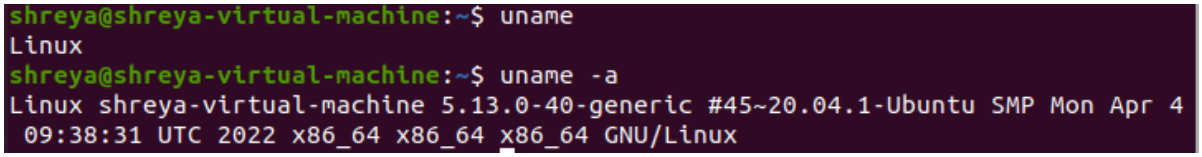
**$uname**

By default, the command prints the kernel name.

You will see “Linux” printed in the output.

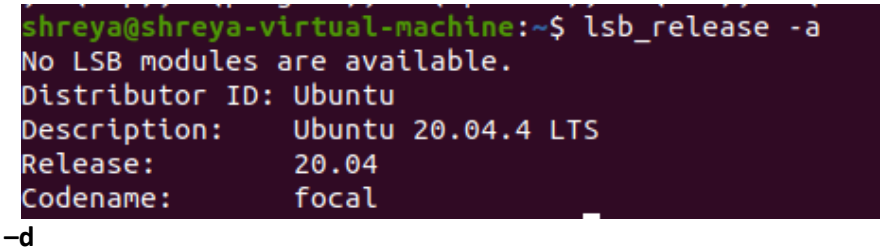
Using the -a option, prints all the system information in the following order: Kernel name, network node hostname, kernel release date, kernel version, machine hardware name, hardware platform, operating system.

**$uname -a**

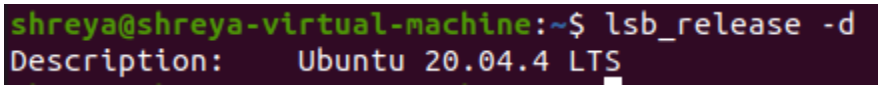



**3. Others way to view Ubuntu operating system versions:**

**$lsb_release -a**




**$lsb_release –d**



**4. To see which shell you are using**

**$echo $0**

```
shreya@shreya-virtual-machine:~$ echo $0
bash
```

To know about the default shell program path

**$printenv SHELL**

```
shreya@shreya-virtual-machine:~$ printenv SHELL
/bin/bash
```

If your default shell is not Bash, you can always switch to it, simply by entering 'bash' on the command line.

```
shreya@shreya-virtual-machine:~$ bash
shreya@shreya-virtual-machine:~$
```

**5. Obtain the user and group identity information**

**$id**

This command displays the user id and group id information of the current user.

It will display the uid (user id) and gid (group id) for the current user.

```
shreya@shreya-virtual-machine:~$ id
uid=1000(shreya) gid=1000(shreya) groups=1000(shreya),4(adm),24(cdrom),27(sudo)
,30(dip),46(plugdev),120(lpadmin),132(lxd),133(sambashare)
```

If you only write **$id –u**

```
shreya@shreya-virtual-machine:~$ id -u
1000
```

-u only return user id.

**$id –u –n** : This command returns corresponding user name related to an user id

```
shreya@shreya-virtual-machine:~$ id -u -n
shreya
```

**6. Get available disk space using $df**

The df command is used to display available disk space. Entering **$df**

```
shreya@shreya-virtual-machine:~$ df
Filesystem     1K-blocks    Used Available Use% Mounted on
udev              958640       0    958640   0% /dev
tmpfs             198772    1528    197244   1% /run
/dev/sda5       19992176 8250852  10702732  44% /
tmpfs             993856       0    993856   0% /dev/shm
tmpfs               5120       4      5116   1% /run/lock
tmpfs             993856       0    993856   0% /sys/fs/cgroup
/dev/loop0           128     128         0 100% /snap/bare/5
/dev/loop4         66816   66816         0 100% /snap/gtk-common-themes/1519
/dev/loop2         63488   63488         0 100% /snap/core20/1405
/dev/loop5         55552   55552         0 100% /snap/snap-store/558
/dev/loop6         44672   44672         0 100% /snap/snapd/14978
/dev/loop3        254848  254848         0 100% /snap/gnome-3-38-2004/99
/dev/loop7         45824   45824         0 100% /snap/snapd/15534
/dev/sda1         523248       4    523244   1% /boot/efi
tmpfs             198768      20    198748   1% /run/user/1000
/dev/sr0         3299872 3299872         0 100% /media/shreya/Ubuntu 20.04.4 LT
S amd64
/dev/loop8         63488   63488         0 100% /snap/core20/1434
```

**$df ~**

This command return disk space usages in the home directory

```
shreya@shreya-virtual-machine:~$ df ~
Filesystem     1K-blocks    Used Available Use% Mounted on
/dev/sda5       19992176 8250852  10702732  44% /
```

The output includes the size, the capacity used and available space for file systems.

**7. View currently running processes:**

**$ps**

The ps command lists each processes that is currently running and its PID (process id).

```
shreya@shreya-virtual-machine:~$ ps
    PID TTY          TIME CMD
   2050 pts/0    00:00:00 bash
   2646 pts/0    00:00:00 bash
   2766 pts/0    00:00:00 ps
```

However, the output only contains the processes that are owned by you.

By using the -e option, you can display all of the processes running on the system. The includes processes owned by other users.

**$ps –e**

```
shreya@shreya-virtual-machine:~$ ps -e
    PID TTY          TIME CMD
      1 ?        00:00:05 systemd
      2 ?        00:00:00 kthreadd
      3 ?        00:00:00 rcu_gp
      4 ?        00:00:00 rcu_par_gp
      6 ?        00:00:00 kworker/0:0H-events_highpri
      9 ?        00:00:00 mm_percpu_wq
     10 ?        00:00:00 rcu_tasks_rude_
     11 ?        00:00:00 rcu_tasks_trace
     12 ?        00:00:00 ksoftirqd/0
     13 ?        00:00:00 rcu_sched
     14 ?        00:00:00 migration/0
     15 ?        00:00:00 idle_inject/0
     16 ?        00:00:00 cpuhp/0
     17 ?        00:00:00 cpuhp/1
     18 ?        00:00:00 idle_inject/1
     19 ?        00:00:00 migration/1
     20 ?        00:00:00 ksoftirqd/1
     22 ?        00:00:00 kworker/1:0H-events_highpri
     23 ?        00:00:00 kdevtmpfs
```

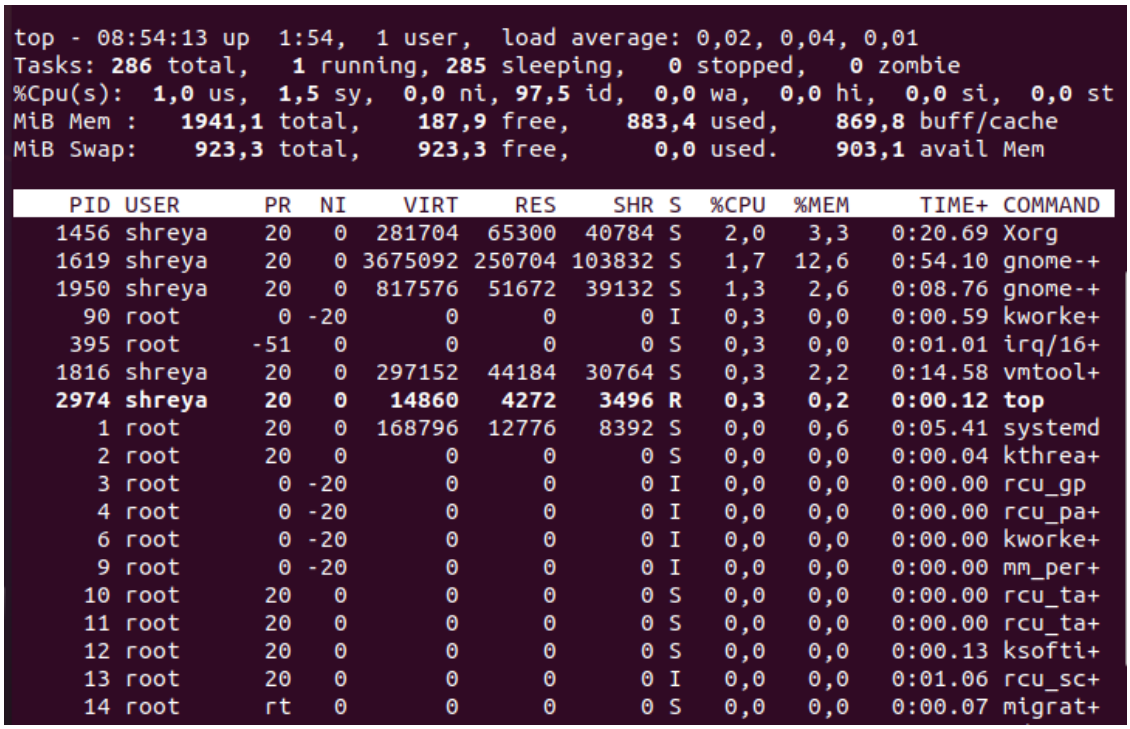
**$ps –u root**

This option displays the processes with root user privilege.

**8. Get information on the running processes and system resources**

The “top” command acts as a task manager and will show running processes and their resource usage. It shows summary information of the system and a table of more detailed information related to the processes or threads, which are currently running and managed by the kernel. This includes information related to cpu and memory usage per process.

**$top**




When you start “top”, you'll be presented with the following elements on the main top screen.

**Summary area** - shows information like system uptime, number of users, load average, and overall memory usage

**Column header** - attribute names

**Task area** - displays the data for each process, or PID

The output keeps refreshing until you press **q** or **Ctrl+c**.

If you want to exit automatically after a specified number of repetitions, use the -n option as follows:

**$top -n 10**

You can press the following keys while top is running to sort the table:

| **Key** | **Sorts by** |
|---|---|
| M | Memory Usage |
| P | CPU Usage |
| N | Process ID (PID) |
| T | Running Time. |

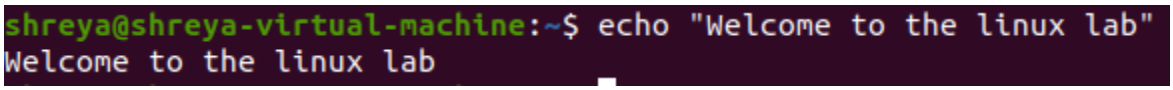
For example, you can find out which process is consuming the most memory by entering shift + m.

**9. Display Messages**

**$ echo**

The echo command displays the given text on the screen. For example, entering:

echo "Welcome to the linux lab"

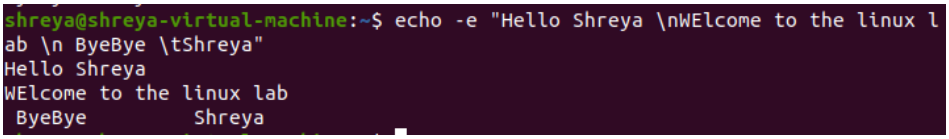



These special characters help you better format your output.

| Special Character | Effect |
|---|---|
| \n | start a new line |
| \t | insert a tab |

Use the -e option of the echo command when working with special characters. For example:

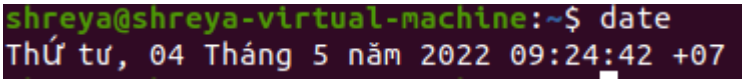
echo -e "This will be printed \nin two lines"




**10. Display date and time**

**$date**

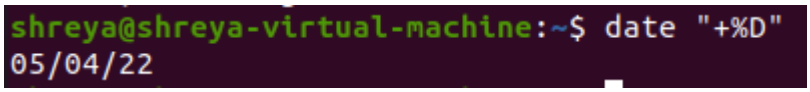
The date command displays the current date and time.




It has several options which help you display the current date and time in your favourite format.

For example, the following command displays the current date in mm/dd/yy format.

**$date "+%D"**




Here are some of the popular format specifiers that you can try out:

| Specifier | Explanation |
|---|---|
| %d | Display the day of the month (01 to 31) |
| %h | Displays the abbreviated month name (Jan to Dec) |
| %m | Displays the month of year (01 to 12) |

%Y Displays the four-digit year

%T Displays the time in 24 hour format as HH:MM:SS

%H Displays the hour

**$date "+%"**

```
shreya@shreya-virtual-machine:~$ date "+%d"
04
shreya@shreya-virtual-machine:~$ date "+%h"
Thg 5
shreya@shreya-virtual-machine:~$ date "+%m"
05
shreya@shreya-virtual-machine:~$ date "+%y"
22
shreya@shreya-virtual-machine:~$ date "+%Y"
2022
shreya@shreya-virtual-machine:~$ date "+%T"
09:29:44
shreya@shreya-virtual-machine:~$ date "+%H"
09
```

**11. List the files and directories**

To list the files and directories in the current directory, enter:

**$ls**

If your directory happens to be empty, ls will not return anything.

The following command will list the many binary and executable files which are present in your /bin directory.

**ls /bin**

The /bin directory happens to be where Linux commands such as ls and pwd are stored. For example, you can see that ls is present by entering:

**ls /bin/ls**

To list all files starting with b in the /bin directory, try entering:

**ls /bin/b***

To list all files ending in r in the /bin directory, enter:

**ls /bin/*r**

To print a long list of files that has additional information compared to the simple ls command, such as the last modified date, enter:

**ls -l**

Here are some popular options that you can try with the **ls** command.

| Option | Description |
|---|---|
| -a | list all files, including hidden files |
| -d | list directories only, do not include files |
| -h | with -l and -s, print sizes like 1K, 234M, 2G |
| -l | include attributes like permissions, owner, size, and last-modified date |
| -S | sort by file size, largest first |
| -t | sort by last-modified date, newest first |
| -r | reverse the sort order |

To get a long listing of all files in /etc, including any hidden files, enter:

ls -la /etc

Here we combined the options -l and -a using the shorter notation, -la.

**12. Search and locate files**

**$find**

The find command is used to search for files in a directory. You can search for files based on different attributes, such as the file's name, type, owner, size, or timestamp.

The find command conducts a search of the entire directory tree starting from the given directory name.

For example, the following command finds all .txt files in the subfolders of the /etc directory:

**find /etc -name '*.txt'**

Along with the listing of all .txt files, you may get some Permission denied errors.

**13. Access control commands**

Required files:

Run the following code to acquire the required files for this exercise:

**$wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-LX0117EN-SkillsNetwork/labs/module%201/usdoi.txt**

Each file and each directory has permissions set for three permission categories: the 'owner', the 'group' and 'all users'.

The following permissions are set for each file and directory:

| **Permission** | **symbol** |
|---|---|
| read | r |

write w

execute x

To see the permissions currently set for a file, run the command **$ls -l**.

For example, to see the permissions for the file named usdoi.txt in your current directory, run:

**$ls -l usdoi.txt**

A sample output looks like:

-rw-r--r-- 1 theia theia 8121 May 31 16:45 usdoi.txt

The permissions set here are rw-r--r--. The - preceding these permissions indicates that usdoi.txt is a file. If it were a directory, you would see a d instead of the -.

The first three entries correspond to **the owner**, the next three correspond to **the group**, and the last three are for all others. You can see the **owner** of the file has **read and write permissions**, while the user **group** only has **read permissions**, and all other users have read permission. No users have execute permissions, as indicated by the - instead of an x in the third position for each user category.

**13.1. Change permissions**

**$chmod**

The chmod (change mode) command lets you change the permissions set for a file.

The change of permissions is specified with the help of a combination of the following characters:

Option Description

r, w and x permissions: read, write and execute, respectively

u, g and o user categories: owner, group and all others, respectively

+, - operations: grant and revoke, respectively

The command below removes read permission for all users (user, group and other) on the file usdoi.txt:

**$chmod -r usdoi.txt**

You can verify the changed permissions by entering:

**$ls -l usdoi.txt**

To add read access to all users on usdoi.txt, enter:

**$chmod +r usdoi.txt**

Now verify the changed permissions:

**$ls -l usdoi.txt**

To remove the read permission for 'all other users' category, enter: **$chmod o-r usdoi.txt**

Verify the changed permissions: **$ls -l usdoi.txt**

**14.1 Display all file contents**

**$cat**

The cat command displays contents of files.

The following command prints the content of the file usdoi.txt which you downloaded earlier.

**$cat usdoi.txt**

**14.2. Display file contents page-wise**

**$more**

The more command displays the file contents page by page.

Press spacebar to display the next page.

**$more usdoi.txt**

**14.3. Display first few lines of a file**

**$head**

Print the first 10 lines of the file usdoi.txt.

**$head usdoi.txt**

You can specify the number of lines to be printed.

Print the first 3 lines of the file usdoi.txt.

**$head -3 usdoi.txt**

**14.4. Display last lines of a file**

**$tail**

Print the last 10 lines of the file usdoi.txt.

**$tail usdoi.txt**

You can specify the number of lines to be printed.

Print the last 2 lines of the file usdoi.txt.

**$tail -2 usdoi.txt**

**14.5. Count lines, words or characters**

**$wc**

If you want to find the number of lines, words and characters in a file, for example usdoi.txt, enter the command:

**wc usdoi.txt**

The output contains the number of lines followed by number of words followed by number of characters in the file.

To print only the number of lines in usdoi.txt, use the -l option:

**wc -l usdoi.txt**

To print only the number of words in usdoi.txt, use the -w option:

**wc -w usdoi.txt**

To print only the number of characters in usdoi.txt, use te -c option:

**wc -c usdoi.txt**

**Customizing view of file content**

**15. View sorted file lines: $sort**

To view the sorted lines of usdoi.txt:
**$sort usdoi.txt**
To view the reverse-sorted lines of usdoi.txt:
**$sort -r usdoi.txt**

**16. View with repeated, consecutive lines merged into one :** uniq

First download the following file:

**$wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-LX0117EN-SkillsNetwork/labs/module%201/zoo.txt**

View the raw contents of zoo.txt:
**$cat zoo.txt**
View the contents of zoo.txt with equal, consecutive lines merged into one:
**$uniq zoo.txt**

**17. Extract lines matching specified criteria**

**grep:** The grep command allows you to specify a pattern and search for lines from the input text that contain a match to the pattern.

The following command prints all lines in the file usdoi.txt, which contain the word people.
**$grep people usdoi.txt**
Some of the frequently used options for grep are:

| **Option** | **Description** |
|---|---|
| -n | Along with the matching lines, also print the line numbers |

-c Get the count of matching lines
-i Ignore the case of the text while matching
-v Print all lines which do not contain the pattern
-w Match only if the pattern matches whole words

Prints all lines from the /etc/passwd file, which do not contain the pattern login.
**$grep -v login /etc/passwd**

**18. View lines of file with filter applied to each line:** cut

The cut command allows you to view the lines of a file after a filter is applied to each line. For example, you can use cut with the -c option to view the first two characters of each line:
**$cut -c -2 zoo.txt**

or each line starting from the second character:

**$cut -c 2- zoo.txt**

**19. View multiple files side by side :** paste

Download the following file:

**wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-LX0117EN-SkillsNetwork/labs/module%201/zoo_ages.txt**

The paste command allows you to view multiple files at once - with lines being aligned as columns. You can see what that looks like by entering:

**$paste zoo.txt zoo_ages.txt**

You can also customize the delimiter. Instead of the default tab, you could specify a comma as follows:
**$ paste -d "," zoo.txt zoo_ages.txt**

**File and folder archiving and compression**
**20. Create and manage file archives: tar**

The tar command allows you to pack multiple files and directories into a single archive file.

The following command creates an archive of the entire /bin directory into a file named bin.tar.

The options used are as follows:

Option Description
-c Create new archive file
-v Verbosely list files processed
-f Archive file name

**$tar -cvf bin.tar /bin**

To see the list of files in the archive, use the -t option:

**tar -tvf bin.tar**

To untar the archive or extract files from the archive, use the -x option:

**tar -xvf bin.tar**

Use the ls command to verify that the folder bin is extracted.

**ls -l**

**21. Package and compress archive files: zip**

The zip command allows you to compress files.

The following command creates a zip file named config.zip consisting of all the files with extension .conf in the /etc directory.

**zip config.zip /etc/*.conf**

The -r option can be used to zip an entire directory.

The following command creates an archive of the /bin directory.

**zip -r bin.zip /bin**

**22. Extract, list, or test compressed files in a ZIP archive**

**unzip**

The following command lists the files of the archive called config.zip

**unzip -l config.zip**

The following command extracts all the files in the archive bin.zip.

**unzip -o bin.zip**

We added the -o option to force overwrite, in case you run the command more than once.

You should see a folder named bin created in your directory.

**Networking commands**
**23. Show the system's host name :** hostname

To view the current host name, run the command below .

**$hostname**
You can use the -i option to view the IP adrress of the host:

**$hostname –i**

**24. Test if a host is reachable: $ping**

Check if www.google.com is reachable. The command keeps sending data packets to the www.google.com server and prints the response it gets back. (Press Ctrl+C to stop pinging)

**ping www.google.com**

If you want to ping only for a limited number of times, use -c option.

**ping -c 5 www.google.com**

**26. Transfer data from or to a server :** curl

You can use curl to access the file at the following URL and display the file's contents on your screen:

**$curl https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/usdoi.txt**
or to access the file at the given URL and save it in your current working directory:

curl -O https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/usdoi.txt

**27. Downloading file(s) from a URL: wget**

The wget command is similar to curl - however it's primary use is for file downloading. One unique feature of wget is that it can recursively download files at a URL.

To see how wget works, first remove usdoi.txt from your current directory:

**rm usdoi.txt**

and re-download it using wget as follows:

**wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Bash%20Scripting/usdoi.txt**