

What is Linux?

(An operating System)

A set of programs that act as a link between the computer and the user.

In layman terms, an operating system is a special software that gives life to a machine.

(A Kernel)

The kernel is a computer program that is the core of a computer's operating system, with complete control over everything in the system.

It manages following resources of the Linux system:

File management, Process Management, I/O Management, Memory Management, Device Management etc.

Complete Linux System= Kernel + GNU system utilities and libraries + other management scripts + installation scripts.

It is often mistaken that linux torvalds has developed linux operating system, but actually he is only responsible for development of linux kernel.

FOS(Free/Open source software)

Linux is a free and open source software which means anyone can create/copy/modify the source code of the OS.

Linux Distributions

Linux kernel + Additional Software

Each distribution has its own focus.

Example of linux distributions are:

Fedora

Ubuntu

Debian

SuSE Linux Enterprise Server(SLES)

OpenSuSE

Linux/Unix?

Unix is copyrighted name only big companies are allowed to use the UNIX copyright and name, so IBM AIX and Sun Solaris and HP-UX all are UNIX OS.

Linux is a Unix clone written from scratch by linux torvalds.

Common things between Linux & Unix?

GUI, File and window managers(KDE,GNOME)

Shells(ksh, csh, bash)

Various office applications such as OpenOffice.org
Development tools(perl, php, python,GNU c/c++, compilers)
Posix(The portable operating system interface)interface.

Why Linux?

Free and Open Source
Stability and Reliability
Security
Flexibility
Hardware Support

What is a Shell?

A shell is a special user program which provide an interface to user to use OS services.
Shell is broadly classified into two categories:
Command Line Shell -Shell can be accessed by user using a command line interface.
A special program called Terminal in Linux/Mac OS or Command prompt in Windows.The result is then displayed on the terminal to the user.

The command line is more powerful.
There will always be a command line.
Server distributions do not include GUIs
Desktop distributions have GUIs and CLIs.

Command Prompt or prompt
A short text message at the start of command line on a command line interface.

The default prompt on the bash shell, which is the default shell on linux, contains the name of the user, the name of the computer and the name of current directory.

For example:
The user named sandra on a computer named localhost and who is working in a directory called work, the prompt would look like

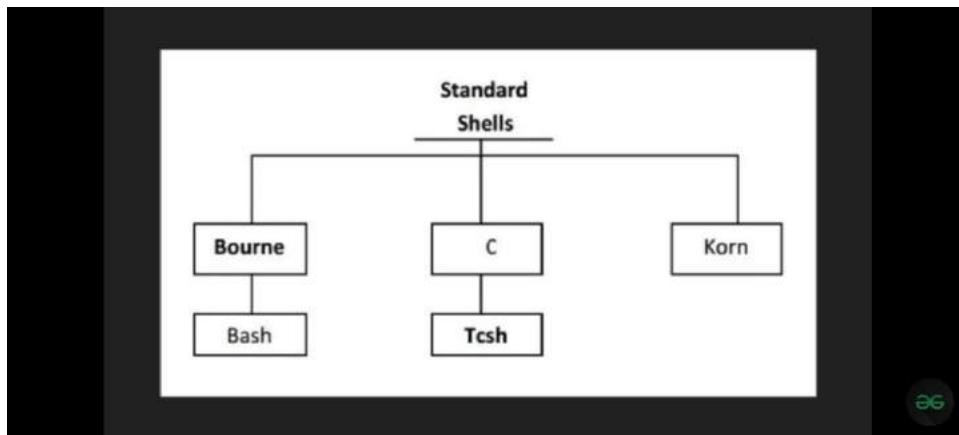
[sandra @ localhost work]\$

Graphical Shell- It provides means for manipulating programs based on GUI, by allowing for operations such as opening, closing, moving and resizing windows, as well as switching focus between windows.

There are several shells available for linux system:

BASH(Bourne Again SHell)
CSH(C SHell)
KSH(Korn SHell)

Each shell does the same job but understands different commands and provide different built in functions.



Shell Relationship:

When you move from one shell to another, UNIX remembers the path you followed by creating a parent-child relationship. Your login shell is always the most senior shell in the relationship- the parent or grandparent depending on how many shells you have used.

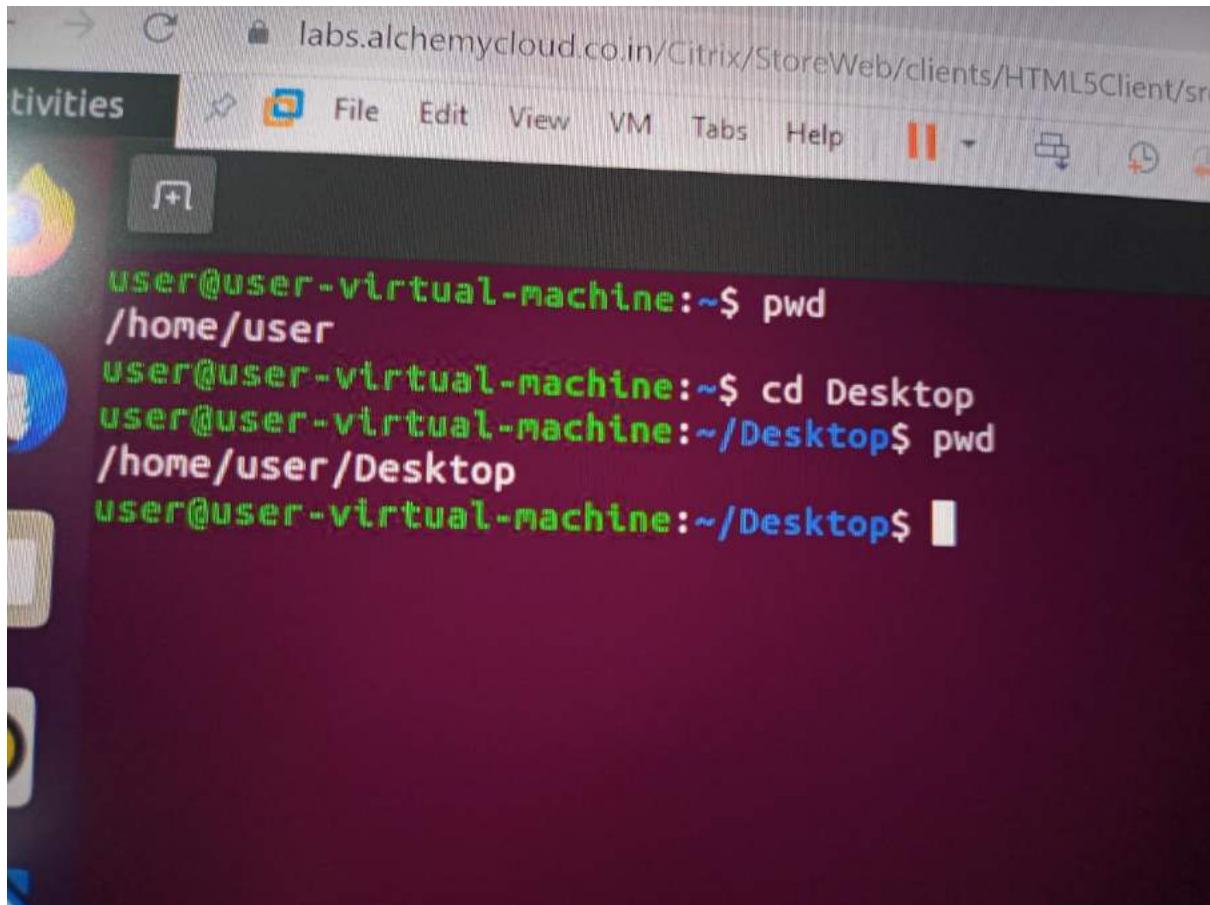
Basic commands

Pwd - It stands for print working directory.

It displays the directory in which we are currently working.

cd - The cd 'dirname' command stands for change directory.

It changes current working directory in the terminal to the new directory name that we have specified after cd.



A screenshot of a Linux terminal window titled "Activities". The window has a dark background and displays the following command history:

```
user@user-virtual-machine:~$ pwd  
/home/user  
user@user-virtual-machine:~$ cd Desktop  
user@user-virtual-machine:~/Desktop$ pwd  
/home/user/Desktop  
user@user-virtual-machine:~/Desktop$
```

man - It stands for manual and it can display the inbuilt manual for most of the commands that we ever need.

Example:**man pwd**

clear - It is used to clean up the terminal so that we can type with more accuracy.

Command usage

Case sensitive

Lower case and upper case letters are not the same

-All linux commands are single words like ls, cat, who etc..

-All standard commands tend to be lowercase.

-Using the type command

type ls (It will give the location of ls)

Internal and External Commands

Internal commands are already loaded in the system. They can be executed anytime and are independent.

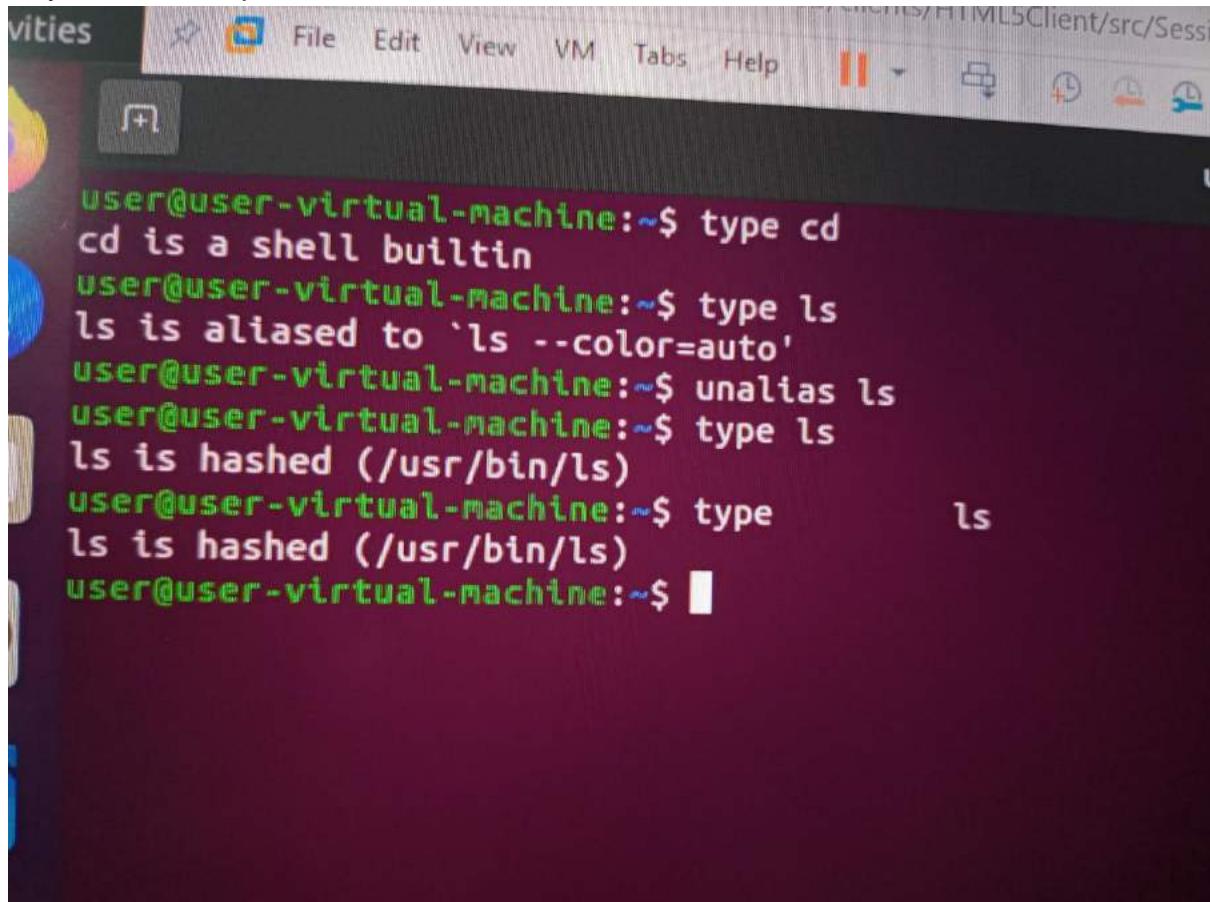
External commands are loaded when the user requests for them.

We can use "type" to know whether a command is internal or external command.

Command Structure

Commands and arguments have to be separated by spaces or tabs to enable the system to interpret them as words.

Any number of spaces can be used.



A screenshot of a terminal window titled "Activities" with a menu bar including File, Edit, View, VM, Tabs, Help, and several icons. The terminal window displays the following text:

```
user@user-virtual-machine:~$ type cd
cd is a shell builtin
user@user-virtual-machine:~$ type ls
ls is aliased to `ls --color=auto'
user@user-virtual-machine:~$ unalias ls
user@user-virtual-machine:~$ type ls
ls is hashed (/usr/bin/ls)
user@user-virtual-machine:~$ type ls
ls is hashed (/usr/bin/ls)
user@user-virtual-machine:~$
```

cd is an internal command and ls is an external command.

Options

There is a special type of argument that is mostly used with a - sign.

Example: ls -l abc

-l is an argument to ls by definition, but more importantly it is a special argument known as an option.

An option is normally preceded by a minus sign to distinguish it from filenames.

There must not be any whitespace between - and l, there must be a space between the command and argument i.e after ls

Options can be combined with only one - sign

ls -lat abc

is same as

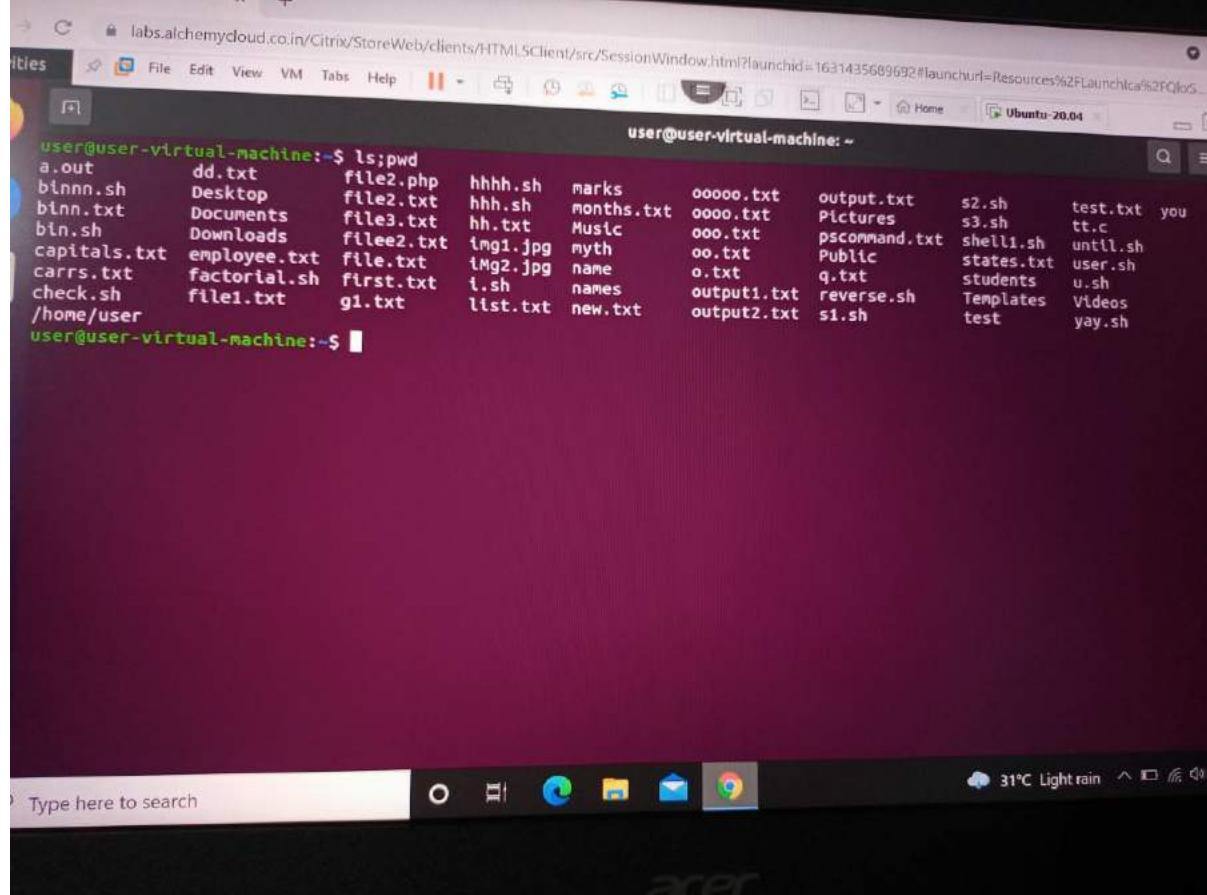
ls -l -a -t abc

Combining commands

Linux allows us to specify more than one command in the command line.
Each command has to be separated from the other by ;
When a command line contains a semicolon, the shell understands that the command on each side of it needs to be processed separately.

The ; is known as metacharacter.

ls;pwd



A screenshot of a Linux desktop environment showing a terminal window. The terminal window title is "Ubuntu-20.04". The command entered is "ls;pwd". The output shows a list of files and directories in the current directory, followed by the current working directory path. The desktop background is red, and the system tray at the bottom shows the date and time as "31°C Light rain".

```
user@user-virtual-machine:~$ ls;pwd
a.out      dd.txt    file2.php   hhhh.sh  marks    ooooo.txt  output.txt  $2.sh    test.txt  you
binnn.sh   Desktop   file2.txt   hhh.sh   months.txt  oooo.txt  Pictures   $3.sh    tt.c
binn.txt   Documents  file3.txt   hh.txt   Music     ooo.txt   pscommand.txt shell1.sh until.sh
bin.sh     Downloads  filee2.txt  img1.jpg myth     oo.txt    Public    states.txt user.sh
capitals.txt employee.txt file.txt   img2.jpg name     o.txt     q.txt    students  u.sh
carrs.txt   factorial.sh first.txt t.sh     names    output1.txt reverse.sh Templates Videos
check.sh    file1.txt  g1.txt    list.txt new.txt  output2.txt s1.sh    test     yay.sh
/home/user

user@user-virtual-machine:~$
```

Multi line prompts

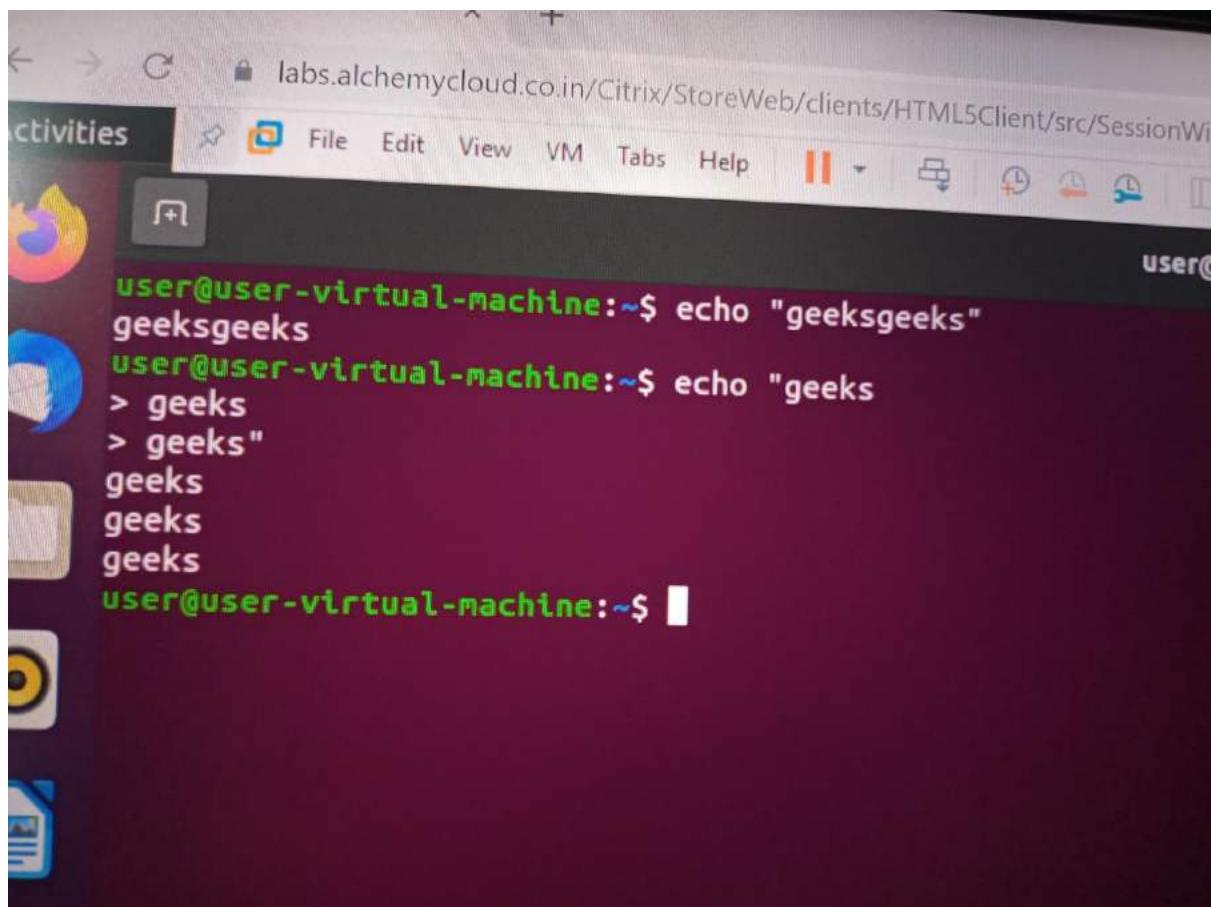
Though the terminal which is restricted to 80 characters, that does not prevent you from entering a command, in one line even though the total width may exceed 80 characters.

The command simply overflows to the next line.

Sometimes it is desirable to split a long command line into multiple lines. In that case the shell issues a secondary prompt, usually > to indicate that the command line isn't complete.

```
$echo "Line 1
>Line 2
```

>Line 3"



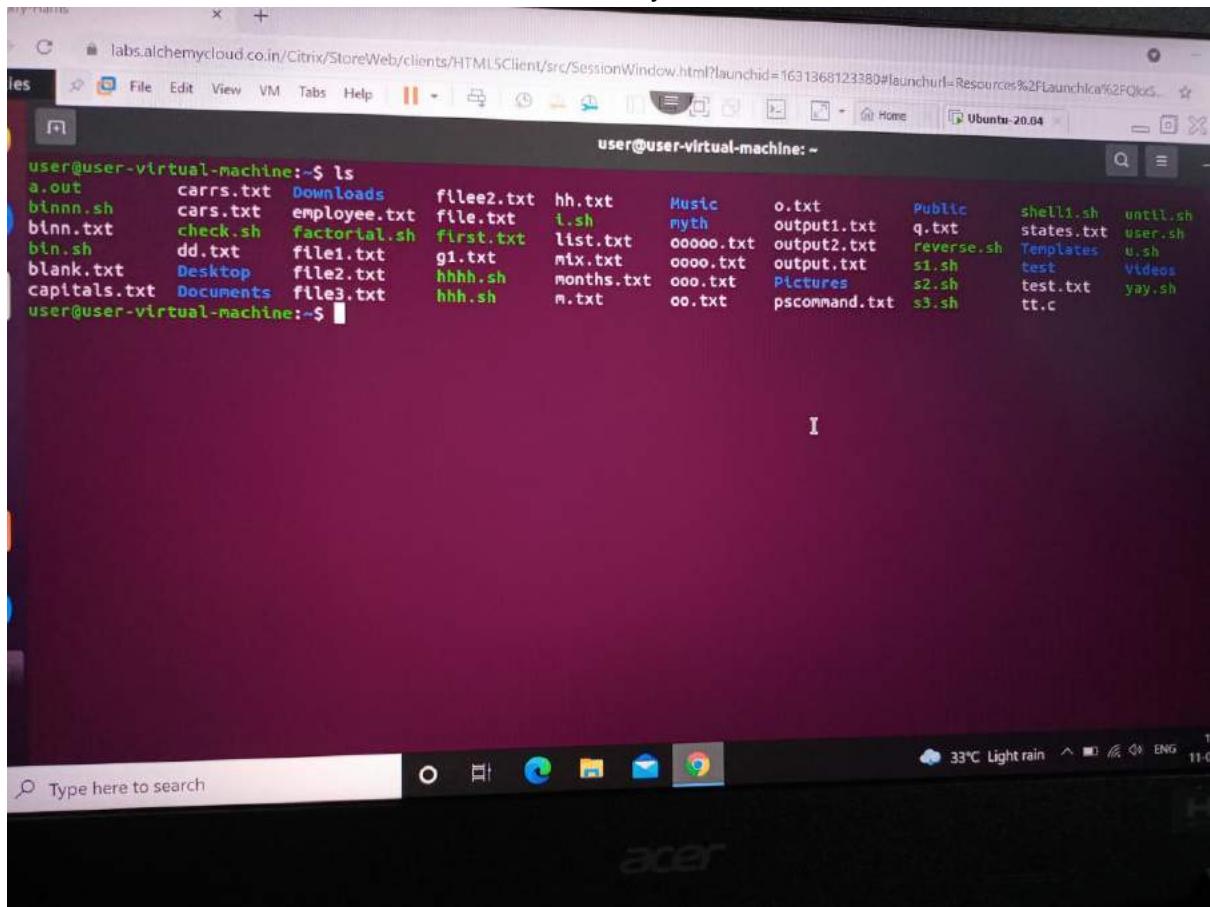
A screenshot of a Linux desktop environment showing a terminal window. The terminal window has a dark red background and displays the following text:

```
user@user-virtual-machine:~$ echo "geeksgeeks"
geeksgeeks
user@user-virtual-machine:~$ echo "geeks
> geeks
> geeks"
geeks
geeks
geeks
user@user-virtual-machine:~$
```

The terminal window is part of a desktop interface with a dock on the left containing icons for various applications like a file manager, terminal, and browser.

Ls command

Ls is a linux shell command that is used to obtain a list of all file names in the current directory.



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "user@user-virtual-machine:~". Inside the terminal, the command "ls" is run, displaying a long list of files and directories. The desktop background is red, and the taskbar at the bottom shows various application icons and system status indicators like battery level and temperature.

```
user@user-virtual-machine:~$ ls
a.out      carrs.txt   Downloads    filee2.txt  hh.txt      Music       o.txt      Public     shell1.sh  until.sh
binn.sh    cars.txt    employee.txt file.txt    i.sh        myth       output1.txt  q.txt      states.txt user.sh
binn.txt   check.sh   factorial.sh first.txt   list.txt   ooooo.txt  output2.txt reverse.sh Templates  u.sh
bin.sh     dd.txt     file1.txt    g1.txt     mix.txt   oooo.txt   output.txt  s1.sh     test      Videos
blank.txt  Desktop   file2.txt   hhhh.sh   months.txt ooo.txt    Pictures   s2.sh     test.txt  yay.sh
capitals.txt Documents file3.txt   hhh.sh    m.txt     oo.txt    pscommand.txt s3.sh     tt.c
```

ls -a (all)

Lists all the files (including hidden files.)

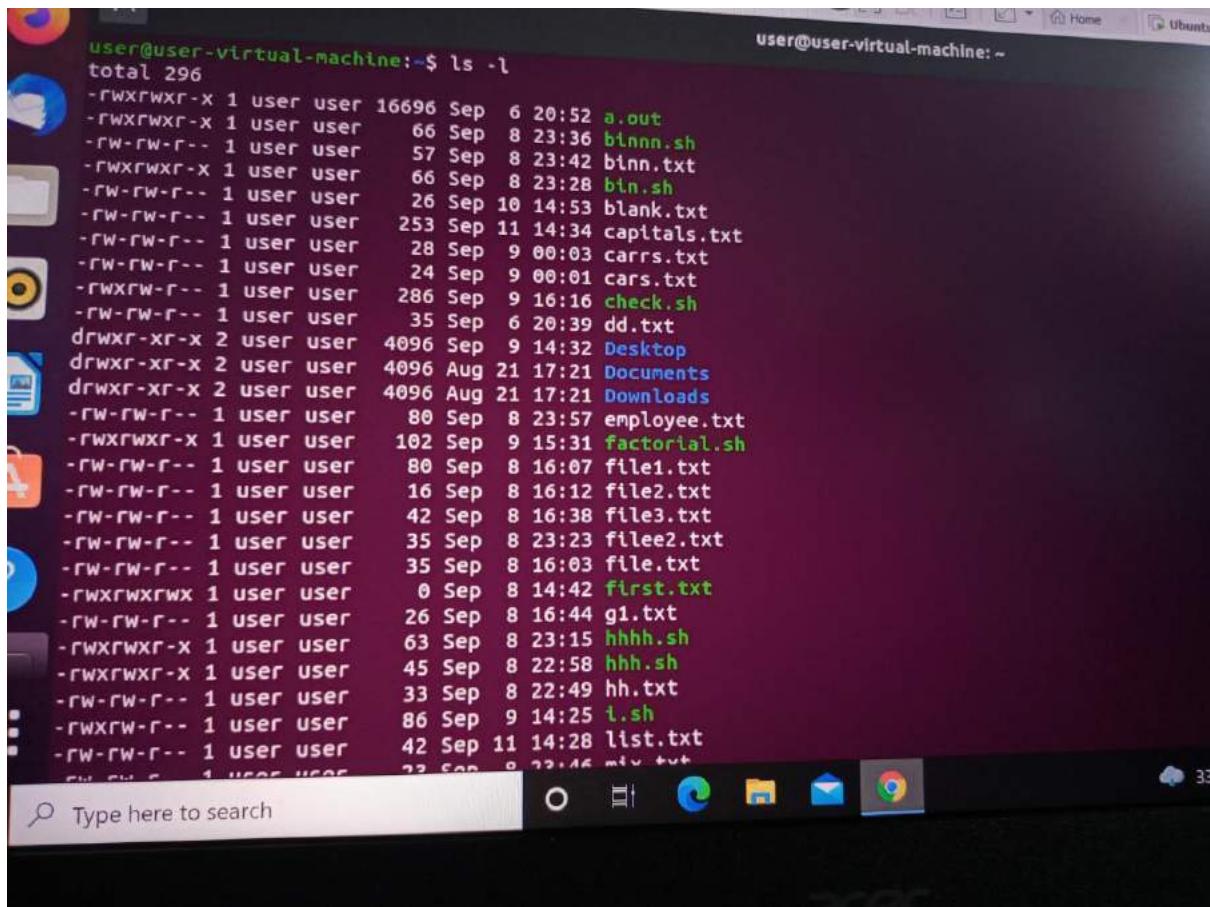
Hidden files are not displayed by default.

Hidden files names begin with a dot.

```
user@user-virtual-machine:~$ ls -a
.           bin.sh      dd.txt      file3.txt  hh.txt      Music      output2.txt  s1.sh      test
..          blank.txt   Desktop    filee2.txt  i.sh       myth       output.txt  s2.sh      test.txt
.a.out      .cache     Documents  file.txt   .lessht   00000.txt  Pictures   s3.sh      tt.c
.bash_history capitals.txt Downloads  first.txt  list.txt  00000.txt  .profile   shell1.sh
.bash_logout  cars.txt   factorial.sh g1.txt    .local    00000.txt  pscommand.txt .ssh
.bashrc      cars.txt   factorial.sh gnupg    mlx.txt   000.txt    Public    states.txt
binnn.sh     check.sh   file1.txt  hhhh.sh   months.txt o.txt     q.txt     .sudo_as_admin_successful Templates
binnn.txt    .config    file2.txt  hhh.sh    m.txt    output1.txt reverse.sh  user.sh
user@user-virtual-machine:~$
```

ls -l (long)

ls Lists all the attributes of all files in the current directory.



The image shows a screenshot of an Ubuntu desktop environment. In the center is a terminal window titled "user@user-virtual-machine:~". The terminal displays the command "ls -l" followed by a detailed list of files and their metadata. The desktop background is dark, and various icons are visible in the dock at the bottom, including a Home icon, a Dash icon, a LibreOffice icon, a file folder icon, a mail icon, and a Google Chrome icon. The system tray in the bottom right corner shows battery status and a network icon.

```
user@user-virtual-machine:~$ ls -l
total 296
-rwxrwxr-x 1 user user 16696 Sep  6 20:52 a.out
-rwxrwxr-x 1 user user    66 Sep  8 23:36 binnn.sh
-rw-rw-r-- 1 user user    57 Sep  8 23:42 binn.txt
-rwxrwxr-x 1 user user   66 Sep  8 23:28 bin.sh
-rw-rw-r-- 1 user user   26 Sep 10 14:53 blank.txt
-rw-rw-r-- 1 user user  253 Sep 11 14:34 capitals.txt
-rw-rw-r-- 1 user user   28 Sep  9 00:03 carrs.txt
-rwxrwxr-- 1 user user   24 Sep  9 00:01 cars.txt
-rw-rw-r-- 1 user user  286 Sep  9 16:16 check.sh
-rw-rw-r-- 1 user user   35 Sep  6 20:39 dd.txt
drwxr-xr-x 2 user user 4096 Sep  9 14:32 Desktop
drwxr-xr-x 2 user user 4096 Aug 21 17:21 Documents
drwxr-xr-x 2 user user 4096 Aug 21 17:21 Downloads
-rw-rw-r-- 1 user user   80 Sep  8 23:57 employee.txt
-rwxrwxr-x 1 user user  102 Sep  9 15:31 factorial.sh
-rw-rw-r-- 1 user user   80 Sep  8 16:07 file1.txt
-rw-rw-r-- 1 user user   16 Sep  8 16:12 file2.txt
-rw-rw-r-- 1 user user   42 Sep  8 16:38 file3.txt
-rw-rw-r-- 1 user user   35 Sep  8 23:23 filee2.txt
-rw-rw-r-- 1 user user   35 Sep  8 16:03 file.txt
-rwxrwxrwx 1 user user    0 Sep  8 14:42 first.txt
-rw-rw-r-- 1 user user   26 Sep  8 16:44 g1.txt
-rwxrwxr-x 1 user user   63 Sep  8 23:15 hhhh.sh
-rwxrwxr-x 1 user user   45 Sep  8 22:58 hhh.sh
-rw-rw-r-- 1 user user   33 Sep  8 22:49 hh.txt
-rwxrwxr-- 1 user user   86 Sep  9 14:25 i.sh
-rw-rw-r-- 1 user user  42 Sep 11 14:28 list.txt
-rw-rw-r-- 1 user user   23 Sep  8 22:46 mix tut
```

ls -t (time of last modification)

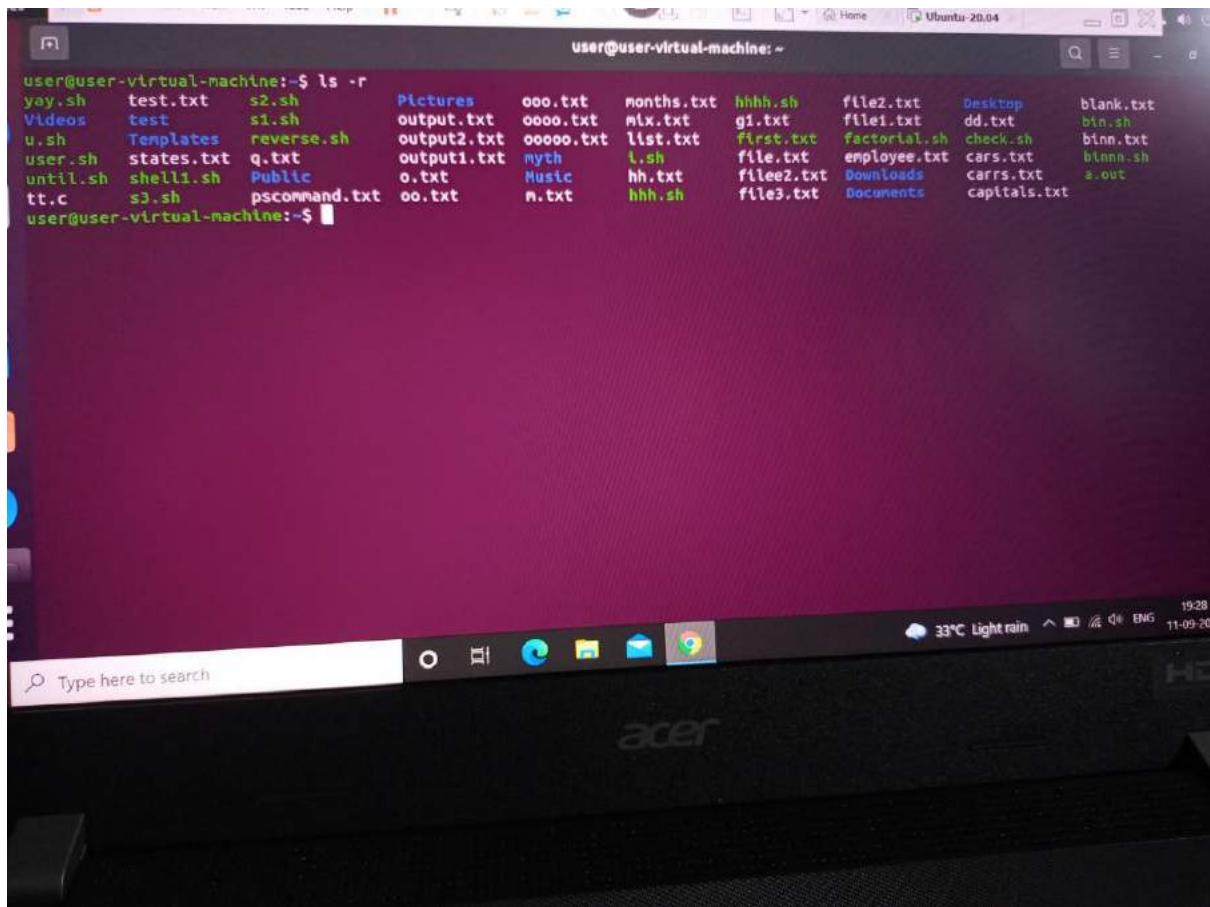
The -t option presents the files in the order of their modification time, the last modified placed first.

ls -tl

```
user@user-virtual-machine:~$ ls -t
capitals.txt  check.sh    u.sh      months.txt  mix.txt   bin.sh    hh.txt    file2.txt shell1.sh Music
list.txt      reverse.sh  s3.sh    carrs.txt   q.txt     filee2.txt g1.txt    file1.txt a.out    Pictures
states.txt    factorial.sh s2.sh    cars.txt   oooooo.txt  ooo.txt  output2.txt file.txt tt.c    Public
blank.txt     test        yay.sh   employee.txt blnn.txt  hhhh.sh  output1.txt myth    dd.txt    Templates
pscommand.txt Desktop    s1.sh    o.txt     binnn.sh   oo.txt   output.txt  test.txt Documents Videos
until.sh      i.sh       user.sh  m.txt    oooo.txt  hhh.sh   file3.txt  first.txt Downloads
user@user-virtual-machine:~$ ls -tl
total 296
-rw-rw-r-- 1 user user  253 Sep 11 14:34 capitals.txt
-rw-rw-r-- 1 user user   42 Sep 11 14:28 list.txt
-rw-rw-r-- 1 user user  298 Sep 11 14:21 states.txt
-rw-rw-r-- 1 user user   26 Sep 10 14:53 blank.txt
-rw-rw-r-- 1 user user 50275 Sep 10 14:05 pscommand.txt
-rwxrwxr-X 1 user user  270 Sep  9 17:24 until.sh
-rwxrwxr-- 1 user user  286 Sep  9 16:16 check.sh
-rwxrwxr-X 1 user user  137 Sep  9 16:06 reverse.sh
-rwxrwxr-X 1 user user  102 Sep  9 15:31 factorial.sh
-rwxrwxr-X 2 user user 4096 Sep  9 14:39 test
drwxrwxr-X 2 user user 4096 Sep  9 14:32 Desktop
drwxr-xr-X 2 user user   86 Sep  9 14:25 i.sh
-rwxrwxr-- 1 user user  259 Sep  9 13:04 u.sh
-rwxrwxr-X 1 user user  186 Sep  9 12:09 s3.sh
-rwxrwxr-X 1 user user  136 Sep  9 11:53 s2.sh
-rwxrwxr-X 1 user user  183 Sep  9 11:50 yay.sh
-rwxrwxr-X 1 user user  187 Sep  9 10:55 s1.sh
-rwxrwxrwx 1 user user  98 Sep  9 10:26 user.sh
-rw-rw-r-- 1 user user   40 Sep  9 00:05 months.txt
-rw-rw-r-- 1 user user  28 Sep  9 00:03 carrs.txt
-rw-rw-r-- 1 user user  24 Sep  9 00:01 cars.txt
user@user-virtual-machine:~$
```

ls -r (reverse)

The -r option reverses the order of presentation. The files will then be listed, with the names sorted in reverse ASCII order.



The screenshot shows a terminal window on an Acer laptop screen. The terminal displays the output of the command `ls -r`, listing numerous files and directories in reverse alphabetical order. The desktop environment includes a search bar, a dock with icons for Home, Dash, and other applications, and a system tray showing weather (33°C Light rain), battery level (100%), and date (11-09-20).

```
user@user-virtual-machine:~$ ls -r
yay.sh    test.txt    s2.sh      Pictures   ooo.txt   months.txt  hhhh.sh   file2.txt   Desktop   blank.txt
Videos    test       s1.sh      output.txt  oooo.txt  mix.txt     g1.txt    file1.txt   dd.txt    bin.sh
u.sh      Templates  reverse.sh output2.txt  ooooo.txt list.txt   first.txt factorial.sh check.sh binn.txt
user.sh   states.txt q.txt      output1.txt myth      l.sh      file.txt   employee.txt cars.txt binnn.sh
until.sh  shell1.sh Public    o.txt      Music     hh.txt   filee2.txt Downloads  carrs.txt a.out
tt.c     s3.sh      pscommand.txt oo.txt   m.txt     hhh.sh   file3.txt Documents capitals.txt
user@user-virtual-machine:~$
```

ls -S (size)

The `-S` option gives the biggest file in size first in the output.

A screenshot of a Linux desktop environment, likely Ubuntu 20.04, showing a terminal window and a file manager. The terminal window at the top displays the command `ls -S` followed by a long list of files and directories. The desktop below shows a dark-themed interface with an Acer logo, a search bar, and various application icons in the dock.

```
user@user-virtual-machine:~$ ls -S
pscommand.txt myth states.txt yay.sh t.sh hhhh.sh file3.txt dd.txt ooooo.txt mix.txt
a.out Pictures check.sh reverse.sh s2.sh employee.txt binn.txt llist.txt file2.txt carrs.txt m.txt
Desktop Public until.sh s3.sh filer1.txt o.txt output1.txt file.txt test.txt file2.txt
Documents Templates u.sh capitals.txt factorial.sh bin.sh shelli.sh output2.txt hh.txt blank.txt oooo.txt
Downloads test s1.sh user.sh tt.c hhh.sh output.txt ooo.txt g1.txt first.txt
Music Videos s1.sh
user@user-virtual-machine:~$
```

ls -Sr

It will list the file which is smallest in size first as we have combined the -r option.

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "user@user-virtual-machine: ~". The command "ls -Sr" is run, displaying a sorted list of files and directories. The desktop background is dark, and the taskbar at the bottom shows various icons and a weather widget indicating "33°C Light rain".

```
user@user-virtual-machine:~$ ls -Sr
first.txt  g1.txt    ooo.txt    output.txt  shell1.sh  bin.sh      factorial.sh  capitals.txt  test       Downloads
oooo.txt   blank.txt hh.txt     output2.txt q.txt      binnn.sh   s3.sh       u.sh        Templates  Documents
file2.txt   test.txt  file.txt   output1.txt o.txt      file1.txt  sz.sh       until.sh    Public     Desktop
m.txt      carrs.txt f1lee2.txt list.txt    binn.txt   employee.txt reverse.sh  check.sh    Pictures   a.out
mix.txt    ooooo.txt dd.txt    file3.txt   hhhh.sh   i.sh       yay.sh    states.txt  myth      Music
cars.txt   oo.txt    months.txt hhh.sh    tt.c      user.sh   si.sh     Videos    
```

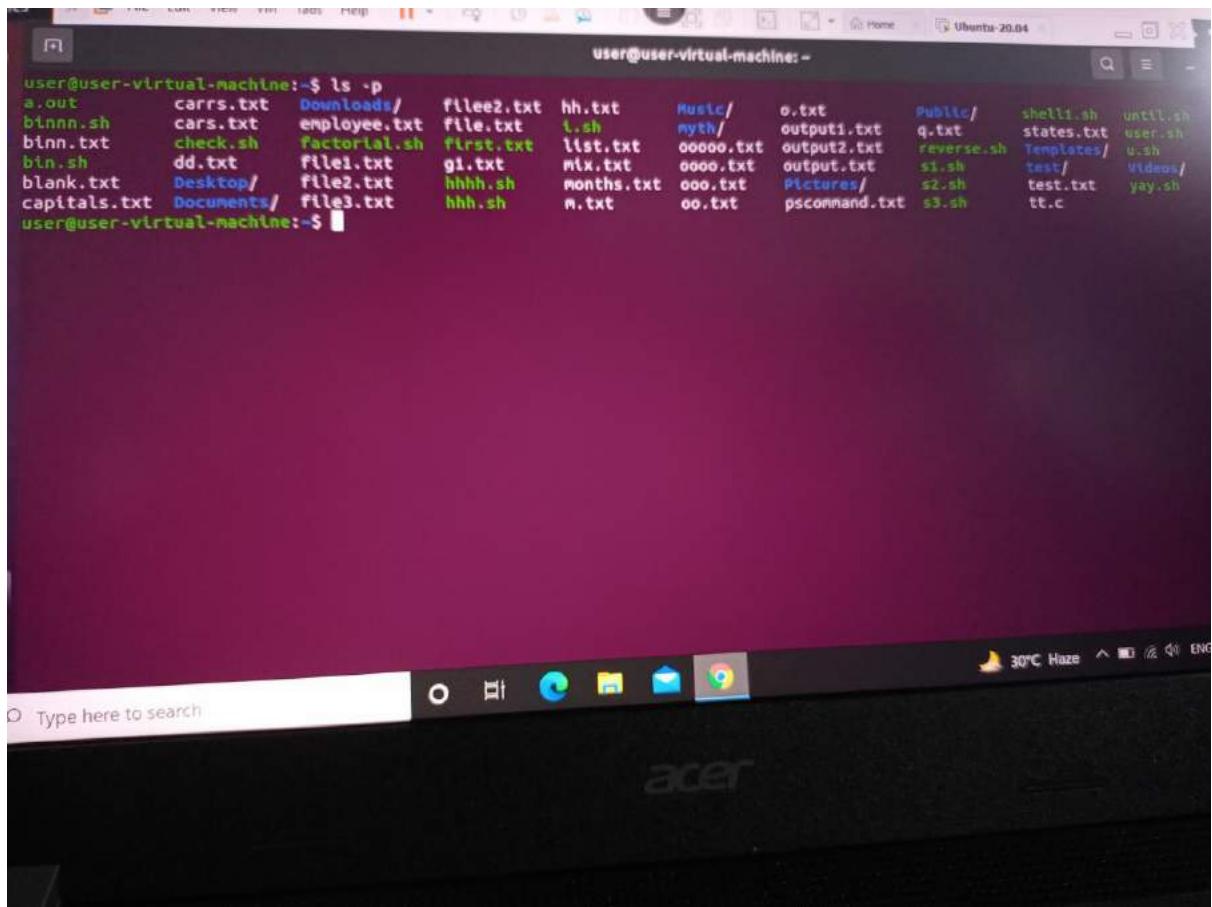
ls -m

Displays the list as a comma separated list.

A screenshot of a Linux desktop environment, specifically Ubuntu 20.04, showing a terminal window. The terminal window title is "user@user-virtual-machine: ~". The user has run the command "ls -m" which lists numerous files and directories in a long format. The files include various scripts (binnn.sh, factorial.sh, file1.txt, etc.), text files (blank.txt, capitals.txt, cars.txt, etc.), and other utilities (check.sh, dd.txt, first.txt, etc.). The desktop background is a dark red gradient. At the bottom, there's a dock with icons for the Dash, Home, and several application icons. A system tray shows the date (11-09-2021), time (19:50), battery level (33°C Light rain), and network status. The Acer logo is visible at the bottom center of the screen.

ls -p

Displays directories with /

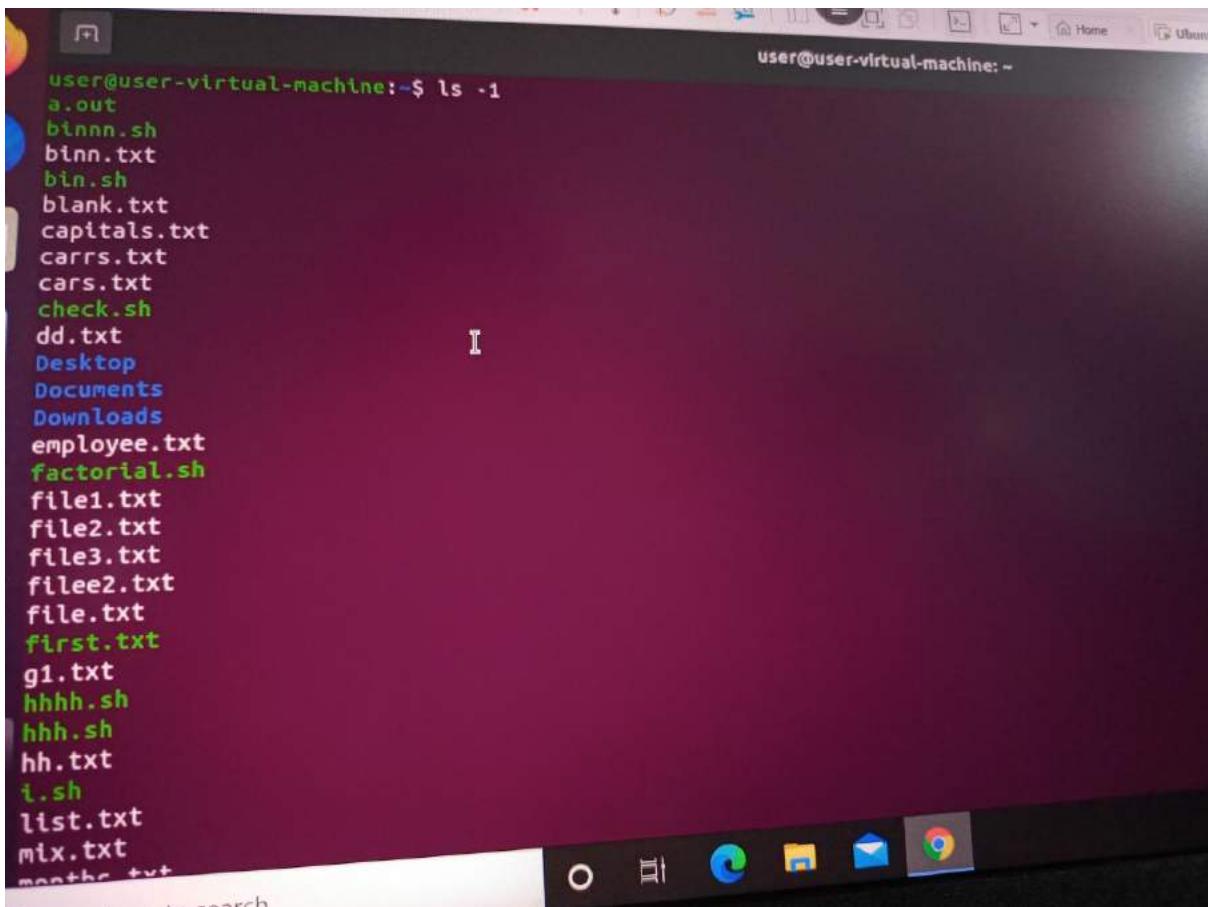


The screenshot shows a terminal window on an Acer laptop running Ubuntu 20.04. The terminal displays the output of the command `ls -p`, listing numerous files and directories in a long, single-line format. The desktop environment includes a search bar, a dock with icons for Home, Dash, and other applications, and a system tray showing the date and time.

```
user@user-virtual-machine:~$ ls -p
a.out      carrs.txt   Downloads/   filee2.txt  hh.txt    Music/      o.txt      Public/    shelli.sh  until.sh
btlnn.sh   cars.txt    employee.txt file.txt    i.sh     myth/      output1.txt  q.txt      states.txt user.sh
btln.txt   check.sh   factorial.sh first.txt  llst.txt  ooooo.txt  output2.txt reverse.sh Templates/ u.sh
bin.sh     dd.txt     file1.txt    gi.txt    mix.txt  oooo.txt  output.txt  s1.sh      test/      Videos/
blank.txt  Desktop/   file2.txt    hhhh.sh  months.txt  oo.txt    Pictures/  s2.sh      test.txt  yay.sh
capitals.txt Documents/ file3.txt    hhh.sh   m.txt    oo.txt    pscommand.txt s3.sh      tt.c
```

`ls -1`

Displays each entry on a line.



```
user@user-virtual-machine:~$ ls -l
a.out
binnn.sh
bin.sh
blank.txt
capitals.txt
carrs.txt
cars.txt
check.sh
dd.txt
Desktop
Documents
Downloads
employee.txt
factorial.sh
file1.txt
file2.txt
file3.txt
filee2.txt
file.txt
first.txt
g1.txt
hhhh.sh
hhh.sh
hh.txt
i.sh
list.txt
mix.txt
months.txt
```

ls -F

Flag filenames.

The screenshot shows a Linux desktop environment with a terminal window open in a browser window. The terminal window displays the output of the 'ls -F' command on a virtual machine. The file manager window shows a list of files and directories on the desktop.

```
user@user-virtual-machine:~$ ls -F
a.out*    cars.txt  Downloads/   filee2.txt  hh.txt    Music/      o.txt      Public/    shell1.sh*  untl1.sh*
binnn.sh*  cars.txt  employee.txt  file.txt   i.sh*     myth/      output1.txt  q.txt      states.txt  user.sh*
binn.txt   check.sh* factorial.sh* first.txt* list.txt  ooooo.txt  output2.txt  reverse.sh* Templates/  u.sh*
bin.sh*    dd.txt    file1.txt    g1.txt    mix.txt  oooo.txt   output.txt  s1.sh*    test/      Videos/
blank.txt  Desktop/  file2.txt    hhhh.sh* Months.txt  ooo.txt    Pictures/  s2.sh*    test.txt  yay.sh*
capitals.txt Documents/ file3.txt    hhh.sh*  M.txt    oo.txt    pscommand.txt s3.sh*    tt.c
user@user-virtual-machine:~
```

ls -b

Displays nonprinting characters in octal.

ls -c

Displays files by file timestamp.

ls -C

Displays files in a columnar format.(default)

ls -d

Displays only directories.

ls -f

Interprets each name as a directory, not a file.

ls -g

Displays the long format listing, but exclude the owner name.

ls -i

Displays the inode for each file.

ls -L

Displays the file or directory referenced by a symbolic link.

ls -n

Displays the long format listing, with GID and UID numbers.

ls -o

Displays the long format listing, but excludes group name.

`ls -q`
Displays all nonprinting characters as ?
`ls -R`
Displays subdirectories as well.
`ls -u`
Displays files by file access time.
`ls -x`
Displays files as rows across the screen.

Cut command

The cut command in linux is a command for cutting out the sections from each line of the files and writing the result to standard output.

It can be used to cut parts of a line by byte position, character and field.

The cut command slices the line and extract the text.

`cut -b`

To extract the specific bytes, we need to follow `-b` option with the list of byte numbers separated by comma.

Range of bytes can also be specified using the hyphen(-). It is necessary to specify list of byte numbers otherwise it gives errors.

Tabs and backspaces are treated like a character of 1 byte.

The screenshot shows a terminal window with a dark background and light-colored text. The terminal session starts with the command `cat students`, which outputs a list of names: Ankur, Anubhav, Chirul, Himanshu, Ishaan, Shubham, and Vishal. This is followed by the command `cut -b 1,2,3 students`, which outputs the first three bytes of each name: Ank, Anu, Cha, etc. Finally, the command `cut -b 1-3,5-7 students` is run, outputting the first three bytes of each name followed by the fifth and seventh bytes: Ankr, Anuhav, Chaul, Himnsh, Ishan, Shuham, Visal.

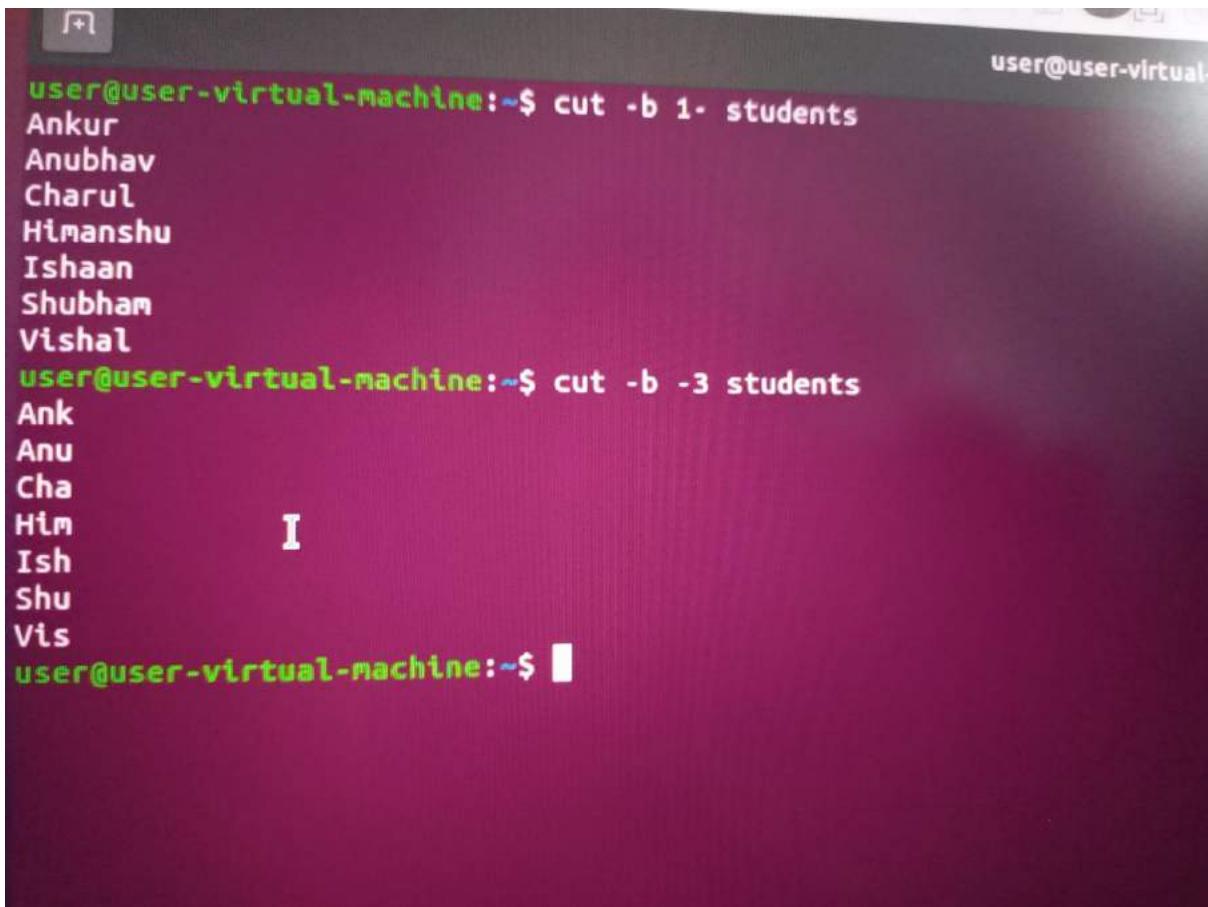
```
user@user-virtual-machine:~$ cat students
Ankur
Anubhav
Chirul
Himanshu
Ishaan
Shubham
Vishal
user@user-virtual-machine:~$ cut -b 1,2,3 students
Ank
Anu
Cha
Him
Ish
Shu
Vis
user@user-virtual-machine:~$ cut -b 1-3,5-7 students
Ankr
Anuhav
Chaul
Himnsh
Ishan
Shuham
Visal
user@user-virtual-machine:~$
```

cut -b

It uses a special form for selecting bytes from beginning upto the end of the line.

(1-) indicate from first byte to end byte of a line.

(3-) indicate from first byte to third byte of a line.

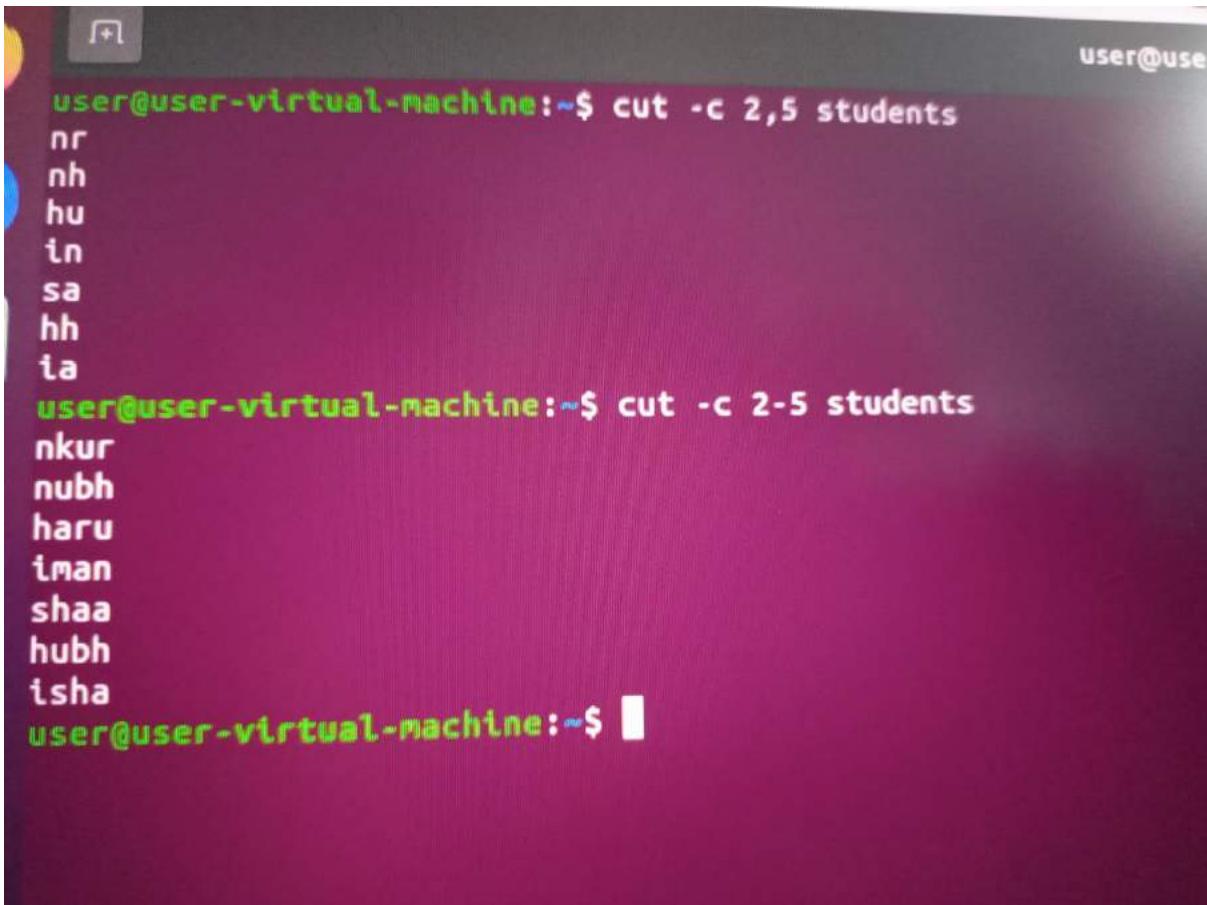


```
user@user-virtual-machine:~$ cut -b 1- students
Ankur
Anubhav
Charul
Himanshu
Ishaan
Shubham
Vishal
user@user-virtual-machine:~$ cut -b -3 students
Ank
Anu
Cha
Him
Ish
Shu
Vis
user@user-virtual-machine:~$
```

`cut -c`

To cut by character use the `-c` option. This selects the characters given to the `-c` option.

This can be a list of numbers separated comma or a range of numbers or a range of numbers separated by hyphen(-).



The screenshot shows a terminal window with a dark background and light-colored text. At the top right, it says "user@use". The terminal displays two commands and their outputs:

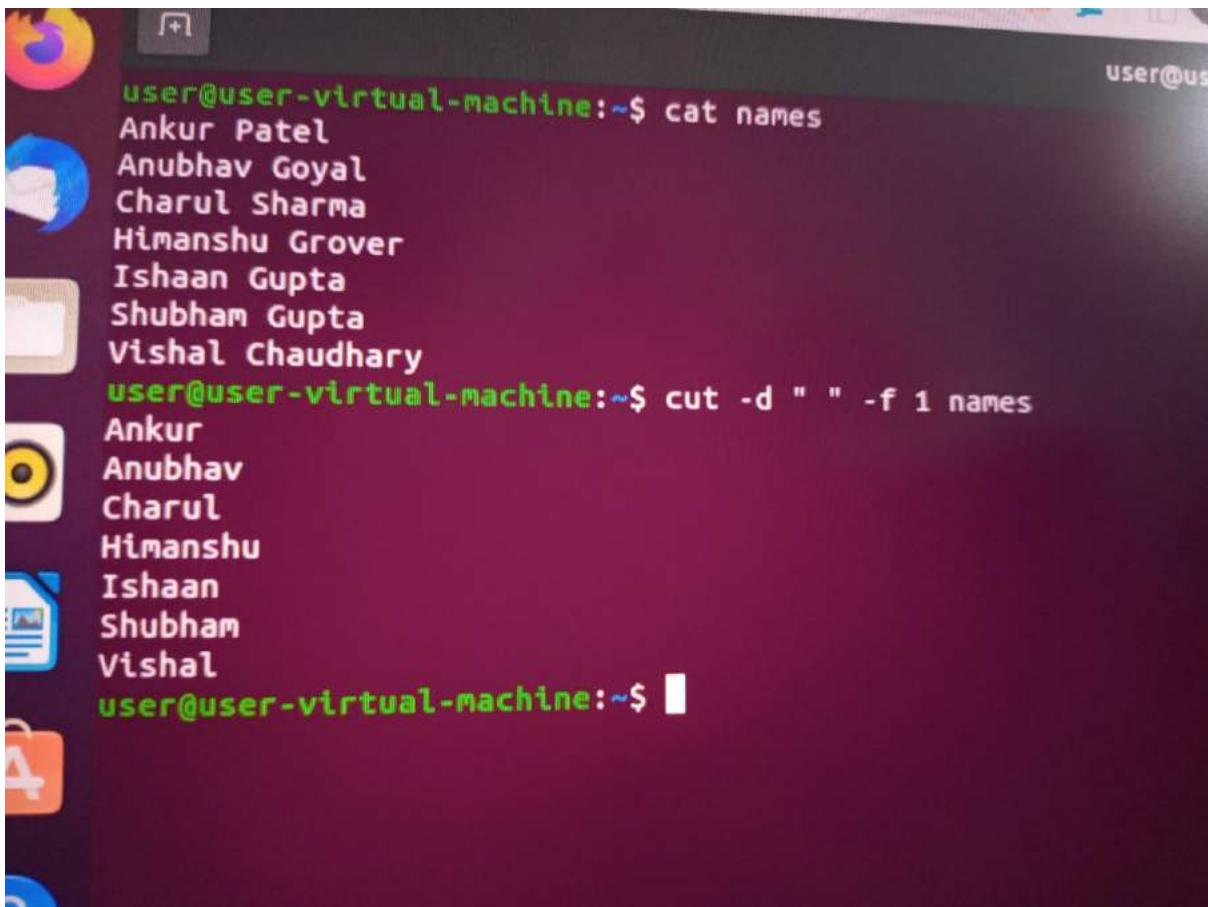
```
user@user-virtual-machine:~$ cut -c 2,5 students
nr
nh
hu
in
sa
hh
ia
user@user-virtual-machine:~$ cut -c 2-5 students
nkur
nubh
haru
iman
shaa
hubh
isha
user@user-virtual-machine:~$ █
```

`cut -f`

To extract the useful information you need to cut by fields rather than columns. List of the field number specified must be separated by comma. Ranges are not described with `-f` option.

Cut uses tab as a default field delimiter but can also work with other delimiter by using `-d` option.

If `-d` option is used then it consider space as a field separator or delimiter:



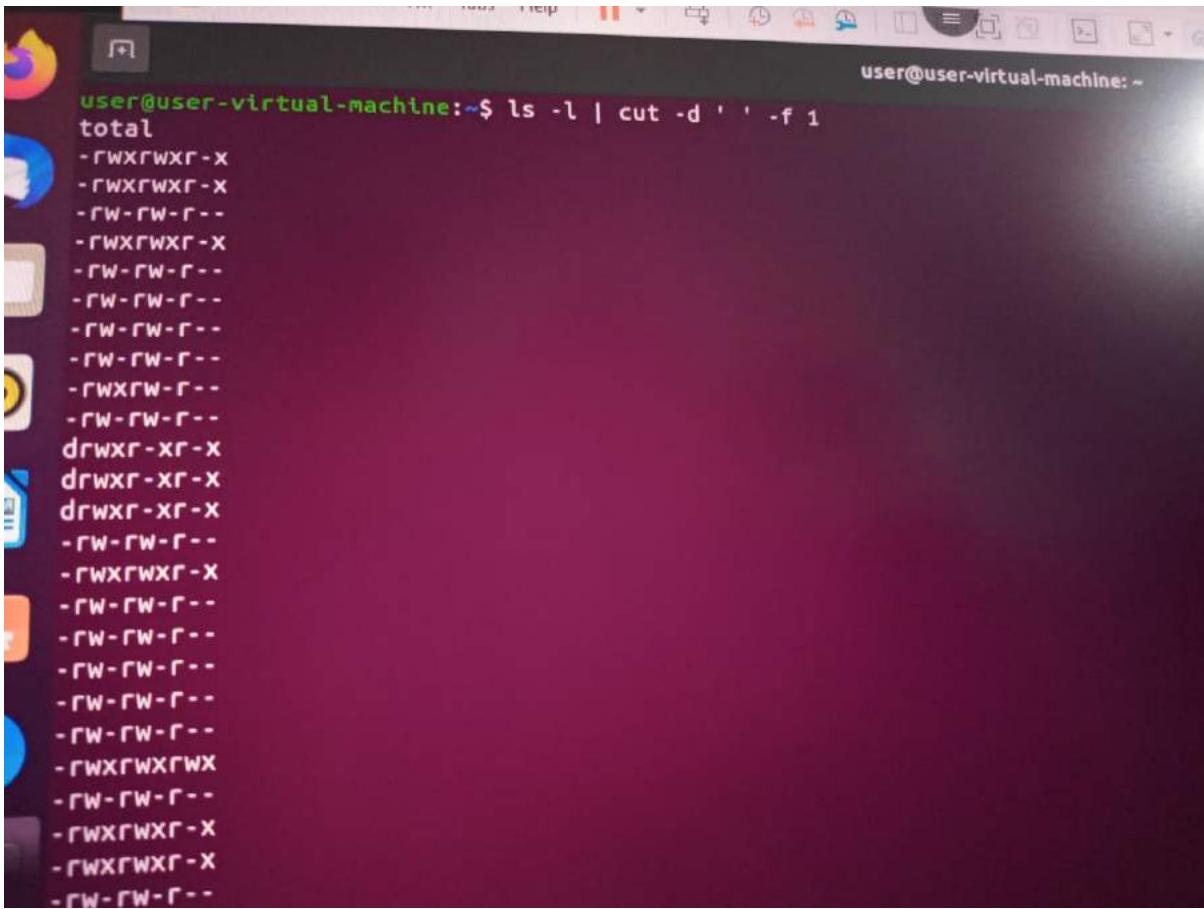
The screenshot shows a terminal window on a Linux desktop environment. The terminal has a dark background with light-colored text. It displays the following command-line session:

```
user@user-virtual-machine:~$ cat names
Ankur Patel
Anubhav Goyal
Charul Sharma
Himanshu Grover
Ishaan Gupta
Shubham Gupta
Vishal Chaudhary
user@user-virtual-machine:~$ cut -d " " -f 1 names
Ankur
Anubhav
Charul
Himanshu
Ishaan
Shubham
Vishal
user@user-virtual-machine:~$
```

Using cut with pipeline

Output of the ls -l command acts as input to cut command and we get only the first field of ls -l command.

```
I+I user@user-virtual-machine:~$ ls -l
total 304
-rwxrwxr-x 1 user user 16696 Sep  6 20:52 a.out
-rwxrwxr-x 1 user user    66 Sep  8 23:36 binnn.sh
-rw-rw-r-- 1 user user     57 Sep  8 23:42 binn.txt
-rwxrwxr-x 1 user user    66 Sep  8 23:28 bin.sh
-rw-rw-r-- 1 user user    26 Sep 10 14:53 blank.txt
-rw-rw-r-- 1 user user   253 Sep 11 14:34 capitals.txt
-rw-rw-r-- 1 user user    28 Sep  9 00:03 carrs.txt
-rw-rw-r-- 1 user user    24 Sep  9 00:01 cars.txt
-rwxrwxr-- 1 user user   286 Sep  9 16:16 check.sh
-rw-rw-r-- 1 user user    35 Sep  6 20:39 dd.txt
drwxr-xr-x 2 user user  4096 Sep  9 14:32 Desktop
drwxr-xr-x 2 user user  4096 Aug 21 17:21 Documents
drwxr-xr-x 2 user user  4096 Aug 21 17:21 Downloads
-rw-rw-r-- 1 user user    80 Sep  8 23:57 employee.txt
-rwxrwxr-x 1 user user   102 Sep  9 15:31 factorial.sh
-rw-rw-r-- 1 user user    80 Sep  8 16:07 file1.txt
-rw-rw-r-- 1 user user    16 Sep  8 16:12 file2.txt
-rw-rw-r-- 1 user user    42 Sep  8 16:38 file3.txt
-rw-rw-r-- 1 user user    35 Sep  8 23:23 filee2.txt
-rw-rw-r-- 1 user user    35 Sep  8 16:03 file.txt
-rw-rw-r-- 1 user user      0 Sep  8 14:42 first.txt
-rwxrwxrwx 1 user user    26 Sep  8 16:44 g1.txt
-rw-rw-r-- 1 user user    63 Sep  8 23:15 hhh.sh
-rwxrwxr-x 1 user user    45 Sep  8 22:58 hh.sh
-rw-rw-r-- 1 user user    33 Sep  8 22:49 t.txt
-rwxrwxr-- 1 user user    86 Sep  9 14:25 t.sh
-rw-rw-r-- 1 user user   42 Sep 11 14:28 list.txt
-rw-rw-r-- 1 user user    23 Sep  9 22:16 mix.txt
```



The screenshot shows a terminal window with a dark background and light-colored text. The command entered was 'ls -l | cut -d ' ' -f 1'. The output consists of a single column of file permissions, each preceded by a dash (-). The permissions are repeated multiple times, likely due to the command being run on a system with many files.

```
user@user-virtual-machine:~$ ls -l | cut -d ' ' -f 1
total
- rwxrwxr-x
- rwxrwxr-x
- rw-rw-r--
- rwxrwxr-x
- rw-rw-r--
- rwxrwxr-x
- rw-rw-r--
- rwxrwxr-x
- rw-rw-r--
drwxr-xr-x
drwxr-xr-x
drwxr-xr-x
- rw-rw-r--
- rwxrwxr-x
- rw-rw-r--
- rwxrwxr-x
- rw-rw-r--
- rwxrwxr-x
- rw-rw-r--
- rwxrwxr-x
- rw-rw-r--
```

cut -c 4

To print characters in a line by specifying the position of the characters.

cut -c 4,6

To print more than one character at a time.

cut -c 4-7

To print a range of characters in a line by specifying the start and end position of the characters.

```
user@user-virtual-machine:~$ cut -c 4 students
u
b
r
a
a
b
h
user@user-virtual-machine:~$ cut -c 4,6 students
u
ba
rl
as
an
ba
hl
user@user-virtual-machine:~$ cut -c 4-7 students
ur
bhav
rul
ansh
aan
bhamb
hal
user@user-virtual-machine:~$
```

cut -c -6 students

Prints first six characters

cut --complement -c 5

Except 5th character all will be there.

```
user@user-virtual-machine:~$ cut -c -6 students
Ankur
Anubha
Charul
Himans
Ishaan
Shubha
Vishal
user@user-virtual-machine:~$ cut --complement -c 5 students
Anku
Anubav
Charl
Himashu
Ishan
Shubam
Vishl
user@user-virtual-machine:~$
```

Sort Command

Sort command is used to sort a file, arranging the records in a particular order. By default, sort command sorts file assuming the contents are ASCII. Using options in sort command, it can also be used to sort numerically.

```
user@user-virtual-machine:~$ cat name
Shubham Gupta
Ankur Patel
Himanshu Grover
Ishaan Gupta
Anubhav Goyal
Diksha Singhal
Vishal Chaudhary
Charul Sharma
Manika Kaushik
Shriyam Verma
Piyush Tyagi
user@user-virtual-machine:~$ sort name
Ankur Patel
Anubhav Goyal
Charul Sharma
Diksha Singhal
Himanshu Grover
Ishaan Gupta
Manika Kaushik
Piyush Tyagi
Shriyam Verma
Shubham Gupta
Vishal Chaudhary
user@user-virtual-machine:~$
```

This command will not change the original content of the file i.e if we do cat name again,we will get the previous content only and not the sorted content.

sort -t

It is followed by the delimiter in quotes, overrides the default.

sort -r

```
user@user-virtual-machine:~$ sort -r name
Vishal Chaudhary
Shubham Gupta
Shriyam Verma
Piyush Tyagi
Manika Kaushik
Ishaan Gupta
Himanshu Grover
Diksha Singhal
Charul Sharma
Anubhav Goyal
Ankur Patel
user@user-virtual-machine:~$
```

sort -n

The screenshot shows a terminal window with a dark background and light-colored text. It displays the following command-line session:

```
user@user-virtual-machine:~$ cat file2.txt
50
39
15
89
200
user@user-virtual-machine:~$ sort -n file2.txt
15
39
50
89
200
user@user-virtual-machine:~$ sort -nr file2.txt
200
89
50
39
15
user@user-virtual-machine:~$
```

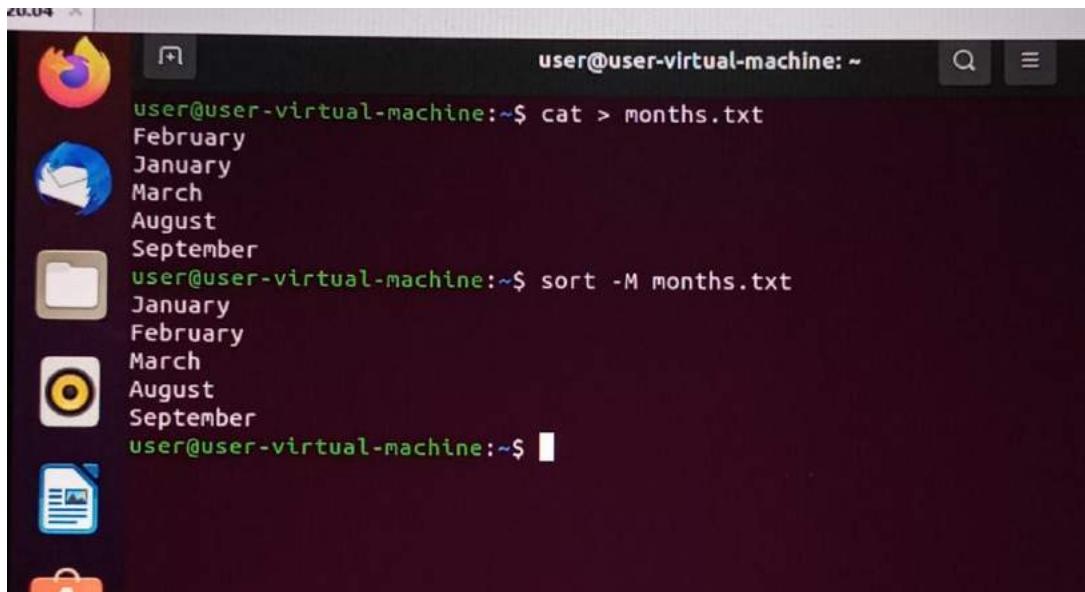
sort -u

The screenshot shows a terminal window with a dark background and light-colored text. It displays the following command-line session:

```
user@user-virtual-machine:~$ cat > carrs.txt
Audi
BMW
Cadillac
BMW
Dodge
user@user-virtual-machine:~$ sort -u carrs.txt
Audi
BMW
Cadillac
Dodge
user@user-virtual-machine:~$
```

sort and removes duplicate.

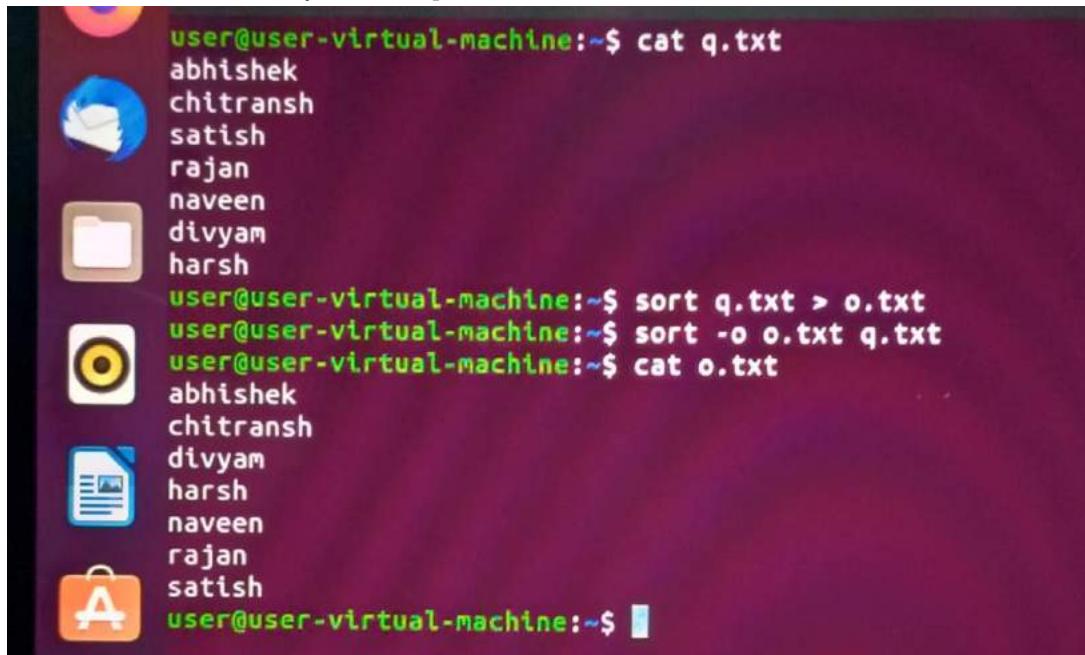
sort -M



A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window. The terminal window has a dark background and contains the following text:

```
user@user-virtual-machine:~$ cat > months.txt
February
January
March
August
September
user@user-virtual-machine:~$ sort -M months.txt
January
February
March
August
September
user@user-virtual-machine:~$
```

[sort inputfile.txt > filename.txt
sort -o filename.txt inputfile.txt]



A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window. The terminal window has a dark background and contains the following text:

```
user@user-virtual-machine:~$ cat q.txt
abhishek
chitransh
satish
rajan
naveen
divyam
harsh
user@user-virtual-machine:~$ sort q.txt > o.txt
user@user-virtual-machine:~$ sort -o o.txt q.txt
user@user-virtual-machine:~$ cat o.txt
abhishek
chitransh
divyam
harsh
naveen
rajan
satish
user@user-virtual-machine:~$
```

sort -k 2

Sorting a file based on the basis of any column number by using -k option.

The screenshot shows a terminal window with a dark background. At the top, there are several icons: a blue circle with a white hand, a yellow folder, a blue document, an orange folder with a white letter 'A', and a blue circle with a white question mark. The terminal prompt is "user@user-virtual-machine:~\$". The first command entered is "cat name", followed by a list of names:

```
user@user-virtual-machine:~$ cat name
Shubham Gupta
Ankur Patel
Himanshu Grover
Ishaan Gupta
Anubhav Goyal
Diksha Singhal
Vishal Chaudhary
Charul Sharma
Manika Kaushik
Shriyam Verma
Piyush Tyagi
```

The second command entered is "sort -t | -k 2 name", followed by the same list of names, sorted by the second column (last name):

```
user@user-virtual-machine:~$ sort -t | -k 2 name
Vishal Chaudhary
Anubhav Goyal
Himanshu Grover
Ishaan Gupta
Shubham Gupta
Manika Kaushik
Ankur Patel
Charul Sharma
Diksha Singhal
Piyush Tyagi
Shriyam Verma
```

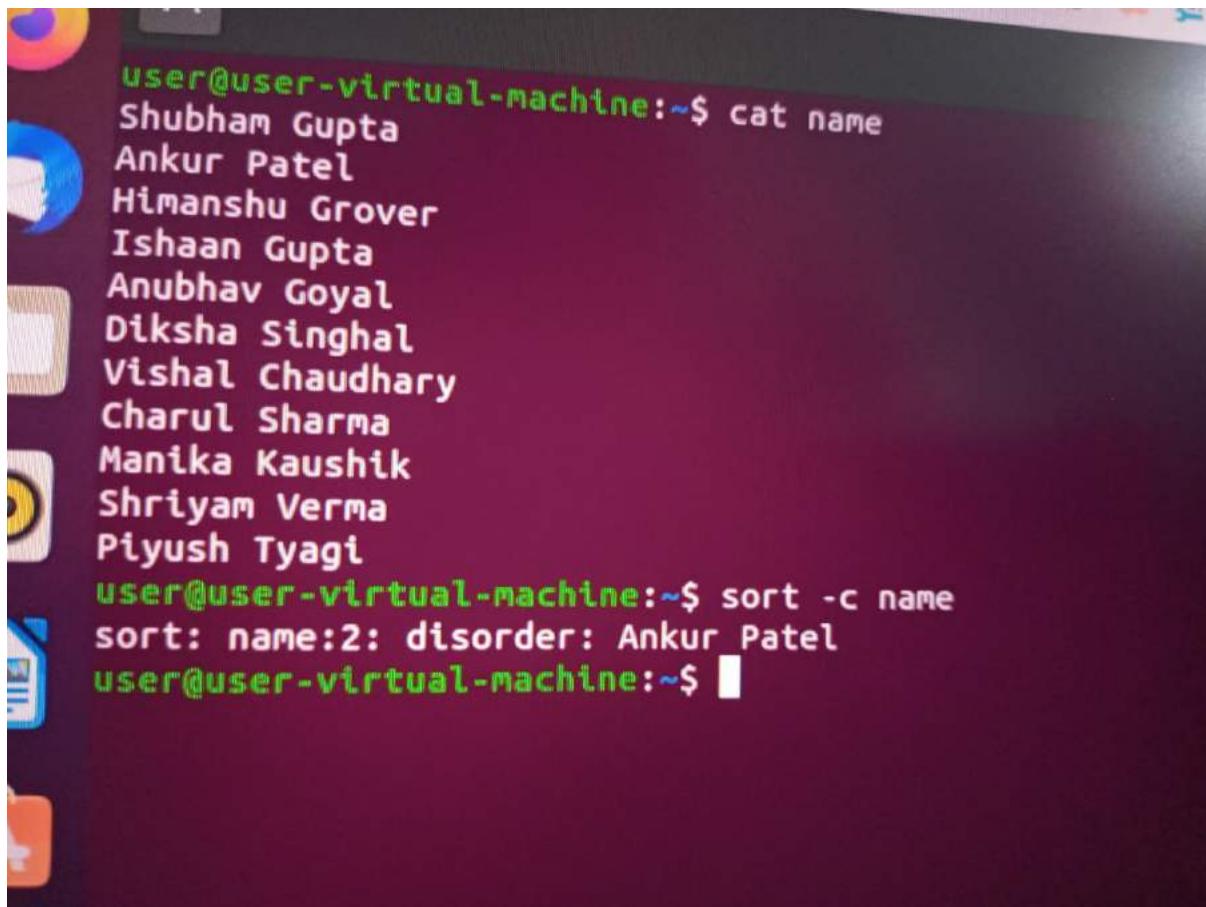
```
sort -t "|" -k 2
```

When string is separated by pipeline and we need to sort by any column .

```
user@user-virtual-machine:~$ cat filep
Shubham|Gupta
Ankur|Patel
Himanshu|Grover
Ishaan|Gupta
Anubhav|Goyal
Diksha|Singhal
Vishal|Chaudhary
Charul|Sharma
Manika|Kaushik
Shriyam|Verma
Piyush|Tyagi
user@user-virtual-machine:~$ sort -t "|" -k 2 filep
Vishal|Chaudhary
Anubhav|Goyal
Himanshu|Grover
Ishaan|Gupta
Shubham|Gupta
Manika|Kaushik
Ankur|Patel
Charul|Sharma
Diksha|Singhal
Piyush|Tyagi
Shriyam|Verma
user@user-virtual-machine:~$
```

sort -c

It is used to check if the file given is already sorted or not and checks if a file is already sorted pass -c option to sort.



```
user@user-virtual-machine:~$ cat name
Shubham Gupta
Ankur Patel
Himanshu Grover
Ishaan Gupta
Anubhav Goyal
Diksha Singhal
Vishal Chaudhary
Charul Sharma
Manika Kaushik
Shriyam Verma
Piyush Tyagi
user@user-virtual-machine:~$ sort -c name
sort: name:2: disorder: Ankur Patel
user@user-virtual-machine:~$
```

Find command

It is a command line utility for walking a file hierarchy.

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.

`find . -name filename`

Find a file in current and sub directories.

`-name demo`

Search for files named demo

`-iname demo`

Search for files named demo irrespective of upper/lower case.

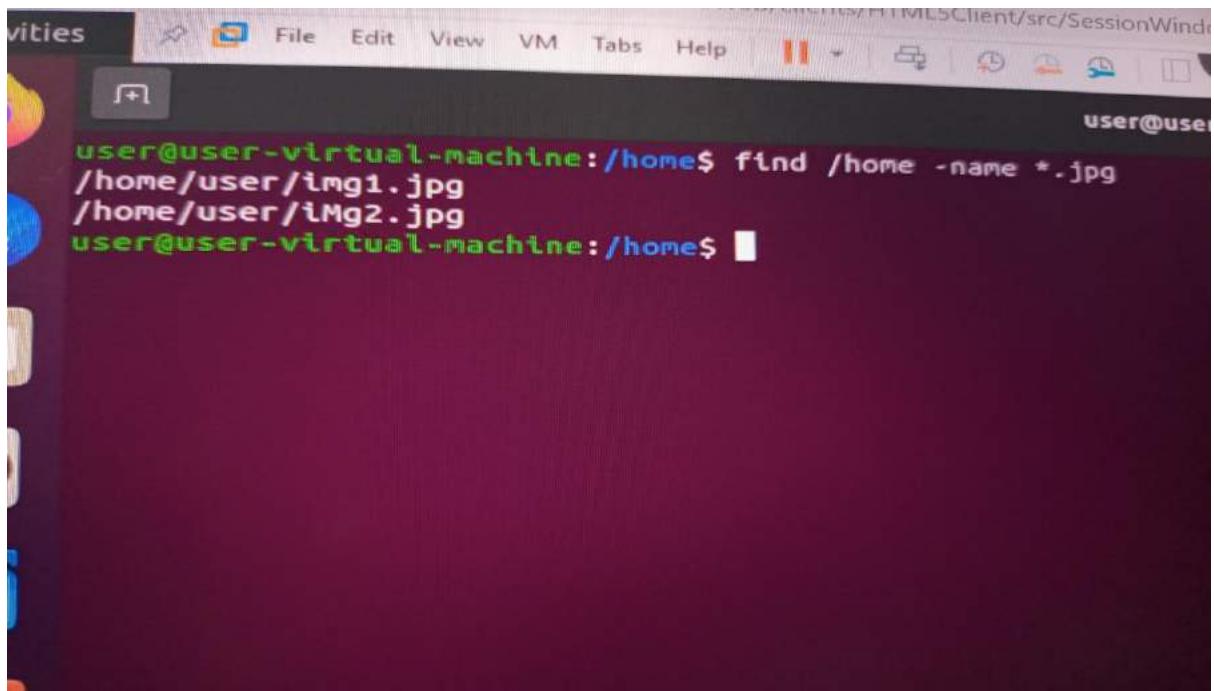
```
user@user-virtual-machine:~$ touch file1.txt file2.php img1.jpg iMg2.jpg  
user@user-virtual-machine:~$ find . -name file1.txt  
.file1.txt  
user@user-virtual-machine:~$ find . -name file10.txt  
user@user-virtual-machine:~$ find . -name img2.jpg  
user@user-virtual-machine:~$ find . -name iMg2.jpg  
.iMg2.jpg  
user@user-virtual-machine:~$
```

find . -empty

```
user@ubuntu:~/Desktop$ find . -empty  
.file1.txt  
.new.txt  
.img1.jpg  
.file2.php  
.iMg2.jpg  
user@ubuntu:~/Desktop$
```

find /home -name *.jpg

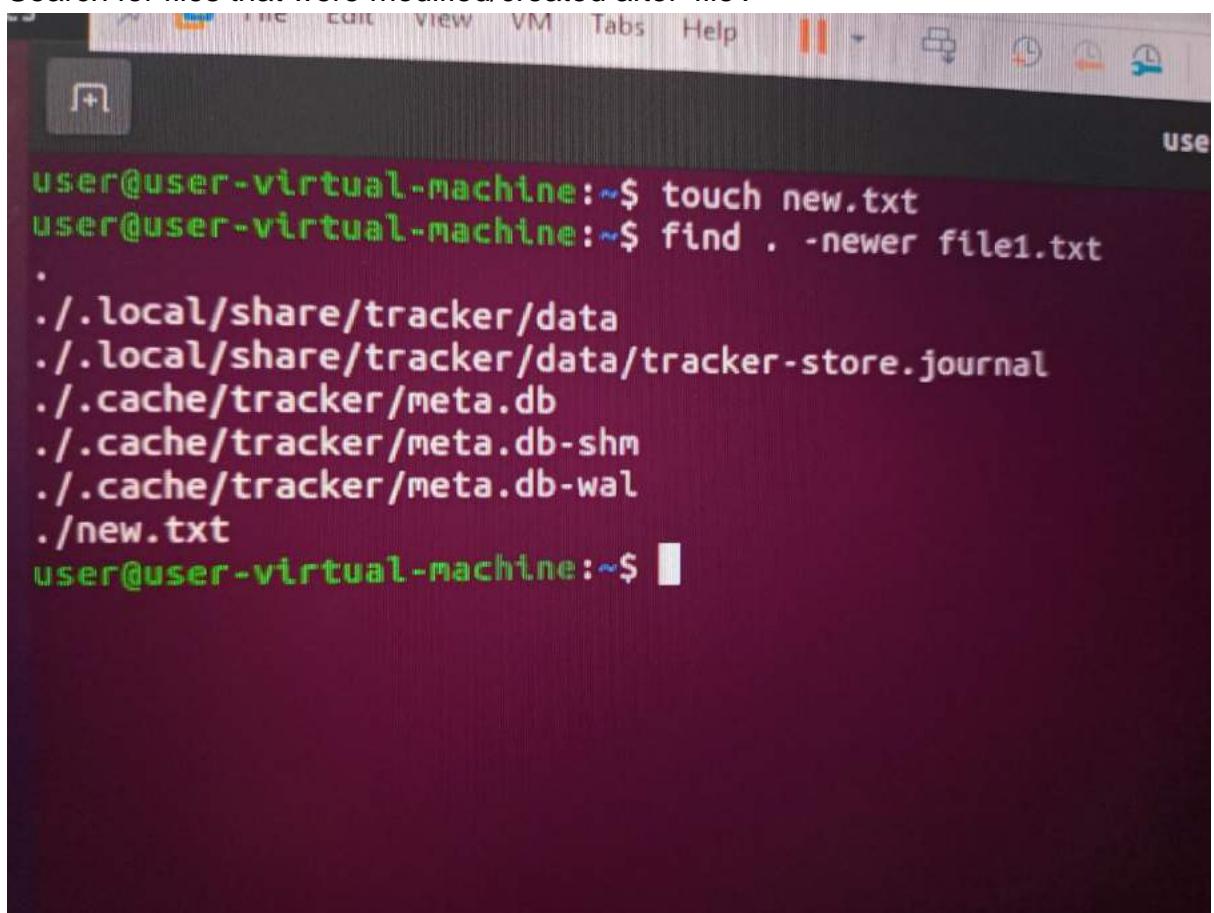
Find all .jpg files in the /home and its sub directories.



```
user@user-virtual-machine:/home$ find /home -name *.jpg
/home/user/img1.jpg
/home/user/iMg2.jpg
user@user-virtual-machine:/home$
```

-newer filename

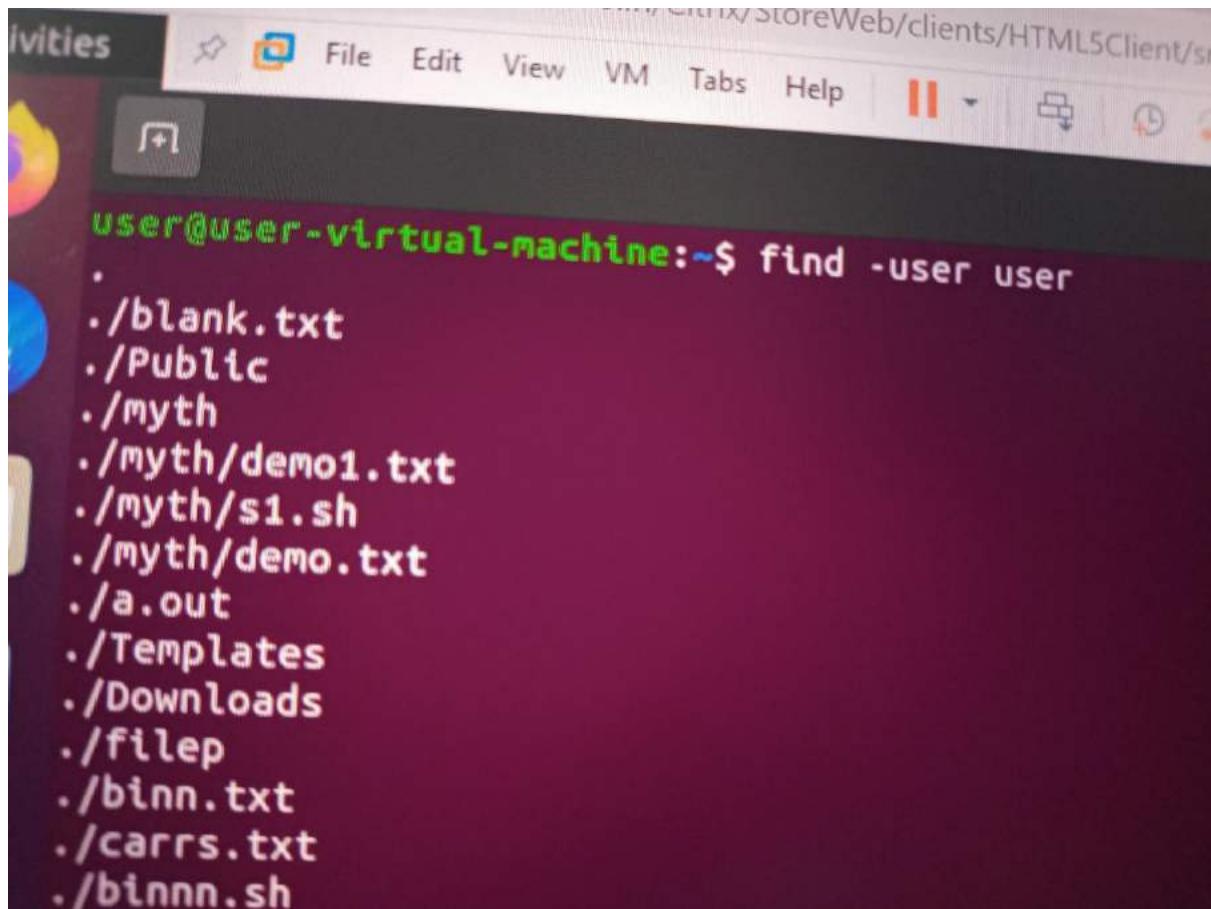
Search for files that were modified/created after 'file'.



```
user@user-virtual-machine:~$ touch new.txt
user@user-virtual-machine:~$ find . -newer file1.txt
.
./.local/share/tracker/data
./.local/share/tracker/data/tracker-store.journal
./.cache/tracker/meta.db
./.cache/tracker/meta.db-shm
./.cache/tracker/meta.db-wal
./new.txt
user@user-virtual-machine:~$
```

-user name

Search for files owned by the user name or ID 'name'.



A screenshot of a Linux terminal window titled "Activities". The terminal window has a dark background and displays the following command and its results:

```
user@user-virtual-machine:~$ find -user user
.
./blank.txt
./Public
./myth
./myth/demo1.txt
./myth/s1.sh
./myth/demo.txt
./a.out
./Templates
./Downloads
./filep
./binn.txt
./carrs.txt
./binnn.sh
```

find ./ -name filename -exec rm -i {} \;

When this command is entered a prompt will come for confirmation, if you want to delete the file or not.

If you enter 'Y/y' it will delete the file. Instead, if you enter 'N/n' it will not delete the file.

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Harris" and the tab title is "Ubuntu-20.04". The terminal content shows the following session:

```
user@user-virtual-machine:~$ find ./ -name mix.txt -exec rm -i {} \;
rm: remove regular file './mix.txt'? y
user@user-virtual-machine:~$ ls
a.out      cars.txt   factorial.sh  file.txt  iMg2.jpg  myth     oo.txt    Public   states.txt user.sh
binn.sh    check.sh   file1.txt    first.txt i.sh      name     o.txt     q.txt    students  u.sh
binn.txt   dd.txt     file2.php   g1.txt    list.txt  names    output1.txt reverse.sh Templates  Videos
bin.sh     Desktop    file2.txt   hhh.sh    marks    new.txt  output2.txt s1.sh    test    yay.sh
blank.txt  Documents  file3.txt   hh.sh    months.txt ooooo.txt output.txt s2.sh    test.txt
capitals.txt Downloads  filee2.txt  hh.txt    m.txt    oooo.txt Pictures  s3.sh    tt.c
carrs.txt  employee.txt filep    img1.jpg Music    ooo.txt  pscommand.txt shell1.sh until.sh
user@user-virtual-machine:~$
```

The desktop interface includes a search bar at the bottom left, a dock with icons for Home, File Manager, Terminal, and others, and a system tray showing weather (31°C Light rain), battery level (ENG 12-4), and other status indicators.

Find and delete a file without confirmation

\$ find / -name file_name -exec rm {} \;

When this command is entered, the file will be deleted directly without asking for any confirmation.

Ex: **\$find /home/user/Public -name new.txt -exec rm {} \;**

Note: The file gets deleted

Search for empty files and directories

\$ find / -empty

This command finds all empty folders and files in the entered directory or sub-directories.

Ex: **\$find /home/user/Public -empty**

/home/user/Public/new.txt

/home/user/Public/new2.txt

/home/user/Public/abc

Find all Empty Files

\$ find / -type f -empty

This command is used to find all empty files under a certain path.

Ex: \$find /home/user/Public -type f -empty

/home/user/Public/new.txt

/home/user/Public/new2.txt

Find all Empty Directories

\$ find / -type d -empty

This command is used to find all empty directories under a certain path.

Ex: \$find /home/user/Public -type d -empty

/home/user/Public/abc

Search text within multiple files

\$ find / -type f -name "*.txt" -exec grep 'any_word' {} \;

This command print lines which have the word specified in them and '-type f' specifies the input type is a file.

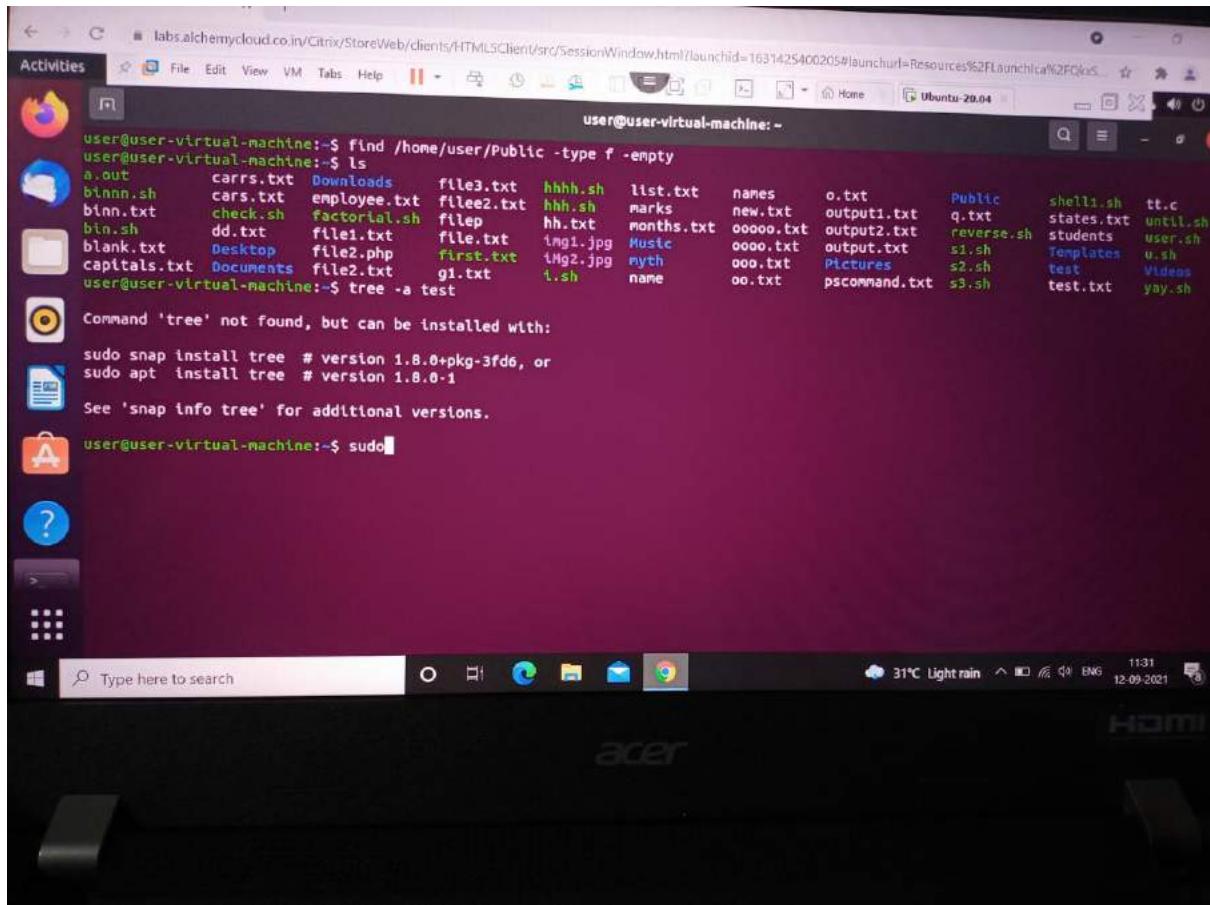
Ex: \$ find /home/user/Public -type f -name "*.txt" -exec grep 'cat' {} \;

I'm a cat.

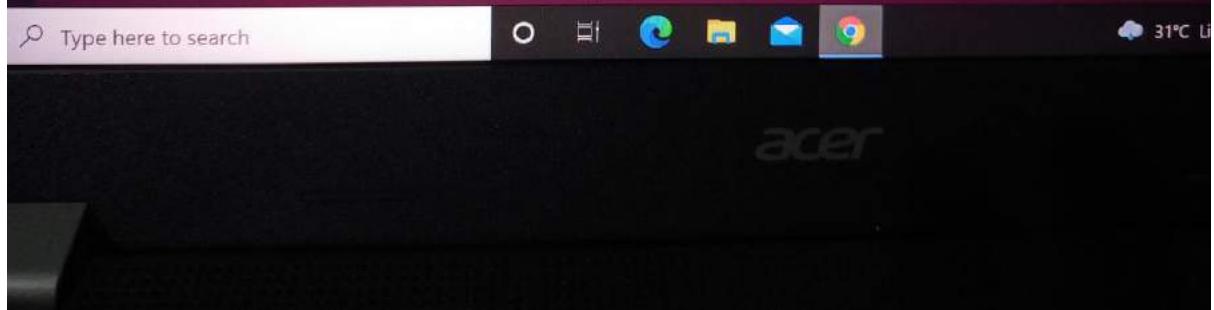
cat

Tree command

–This command helps you list all files and directories under a specified directory.
\$ tree -a directory_name

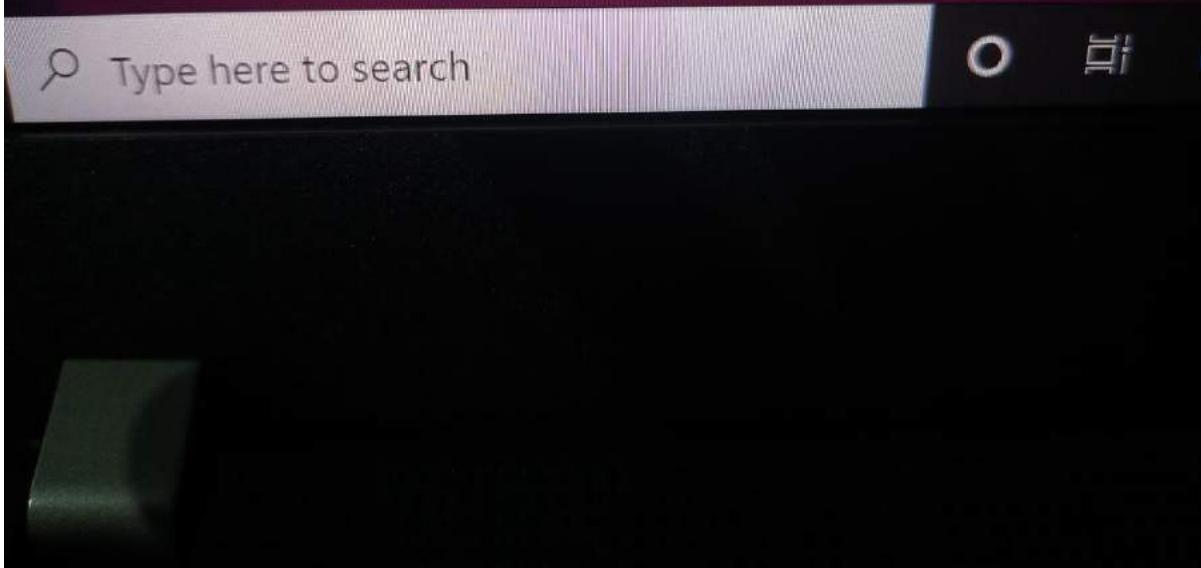


```
user@user-virtual-machine:~$ sudo apt install tree
[sudo] password for user:
Sorry, try again.
[sudo] password for user:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
tree
0 upgraded, 1 newly installed, 0 to remove and 192 not upgraded.
Need to get 43.0 kB of archives.
After this operation, 115 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 tree amd64 1.8.0-1 [43.0 kB]
Fetched 43.0 kB in 0s (129 kB/s)
Selecting previously unselected package tree.
(Reading database ... 188141 files and directories currently installed.)
Preparing to unpack .../tree_1.8.0-1_amd64.deb ...
Unpacking tree (1.8.0-1) ...
Setting up tree (1.8.0-1) ...
Processing triggers for man-db (2.9.1-1) ...
user@user-virtual-machine:~$
```

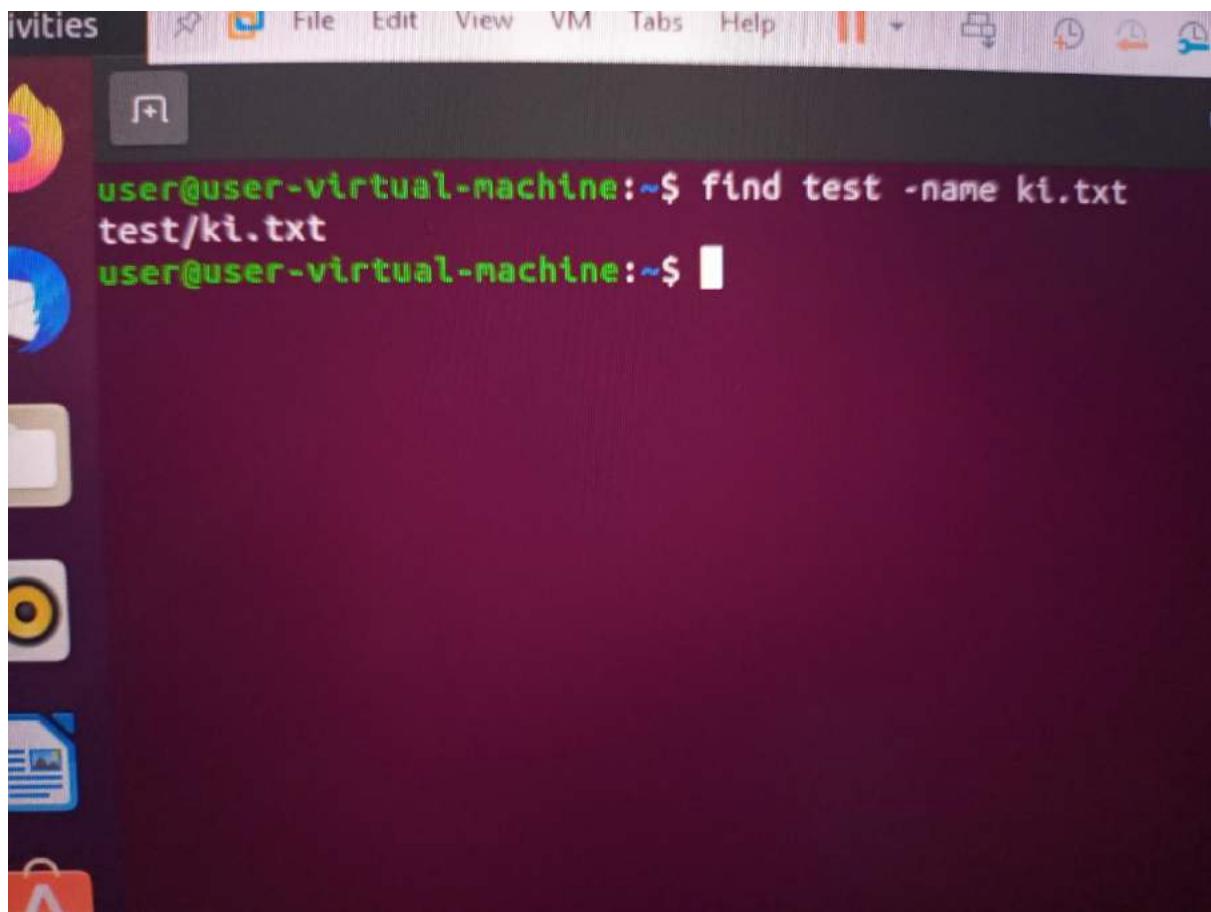


```
user@user-virtual-machine:~$ tree -a test
test
└── ki.txt

0 directories, 1 file
user@user-virtual-machine:~$
```



Find a specified file – This command shows the path where the specified file is present
\$ find directory_name -name file_name.txt



The screenshot shows a terminal window with a dark red background. At the top, there's a menu bar with options like File, Edit, View, VM, Tabs, Help, and several icons. Below the menu, there's a toolbar with icons for file operations. The main area of the terminal displays the following text:

```
user@user-virtual-machine:~$ find test -name ki.txt
test/ki.txt
user@user-virtual-machine:~$
```

Find a list of files - This command shows the path where a list of files is present

\$ find directory_name -name “*.txt”

Finds all the text files under a particular directory

The screenshot shows a Linux desktop environment with an Acer laptop model. A terminal window is open, displaying a command-line session. The session starts with listing files in the current directory using `ls -p`, followed by three `find` commands: `find Desktop -name "*.txt"`, `find Public -name "*.txt"`, and `find myth -name "*.txt"`. The terminal window has a dark background with light-colored text. The desktop interface includes a taskbar with icons for file manager, browser, and other applications, and a system tray showing battery status at 31% and a light rain icon.

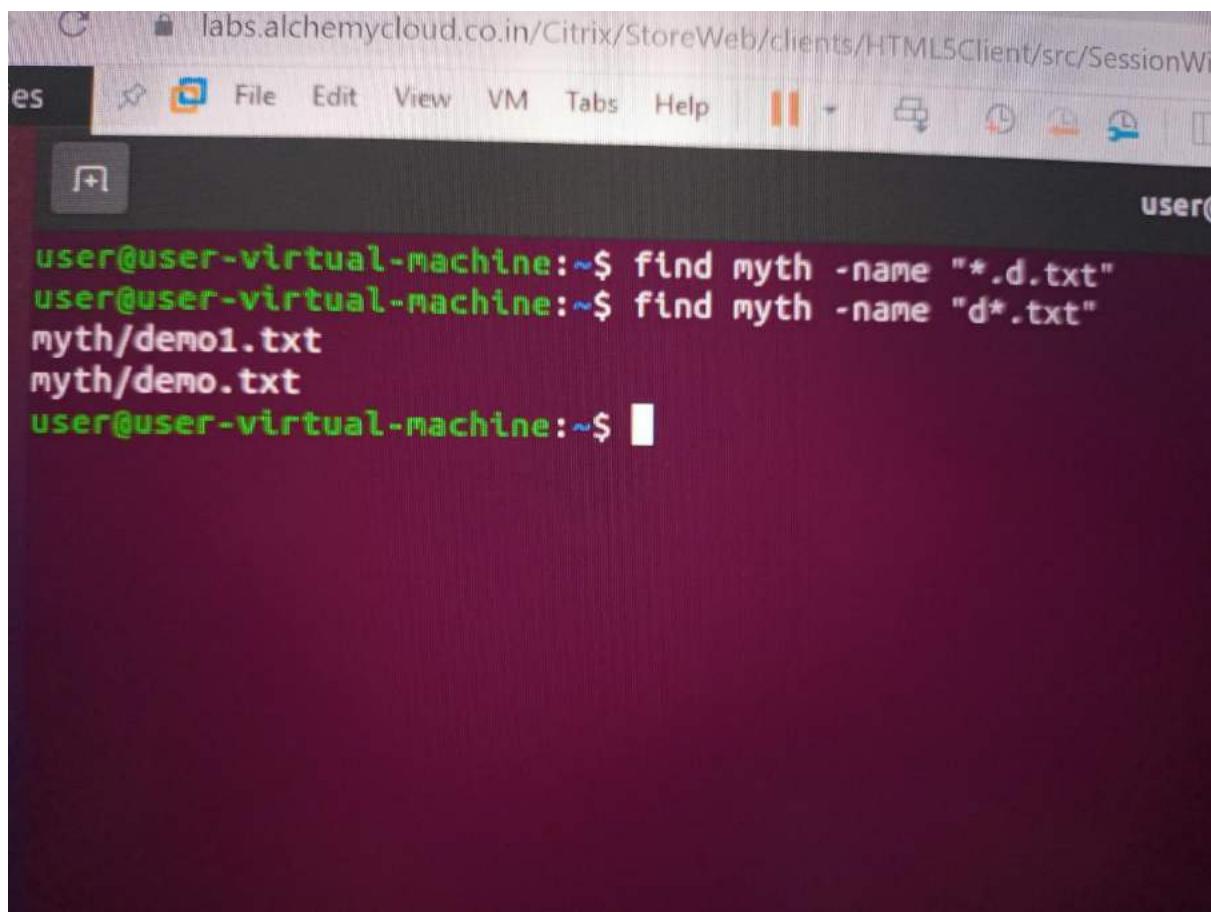
```
user@user-virtual-machine:~$ ls -p
a.out      cars.txt    factorial.sh   file.txt   iMg2.jpg   name     o.txt      q.txt      students   u.sh
binnn.sh   check.sh    file1.txt    first.txt  i.sh       names    output1.txt  reverse.sh  Templates/  Videos/
binn.txt   dd.txt      file2.php   g1.txt    list.txt   new.txt   output2.txt  s1.sh      test/      yay.sh
bin.sh     Desktop/   file2.txt   hhhh.sh   marks    ooooo.txt  output.txt  s2.sh      test.txt
blank.txt  Documents/ file3.txt   hh.sh     months.txt  oooo.txt  Pictures/   s3.sh      tt.c
capitals.txt Downloads/ filee2.txt hh.txt    Music/    ooo.txt   pscmd.txt  shell1.sh  states.txt user.sh
carrs.txt  employee.txt filep     img1.jpg  myth/    oo.txt    Public/    sPublic/   until.sh
user@user-virtual-machine:~$ find Desktop -name "*.txt"
Desktop/k.txt
user@user-virtual-machine:~$ find Public -name "*.txt"
user@user-virtual-machine:~$ find myth -name "*.txt"
myth/demo1.txt
myth/demo.txt
user@user-virtual-machine:~$
```

\$ find directory_name -name “*t.txt”

Finds all the text files ending with “t” under a particular directory.

\$ find directory_name -name “t*.txt”

Finds all the text files starting with “t” under a particular directory.

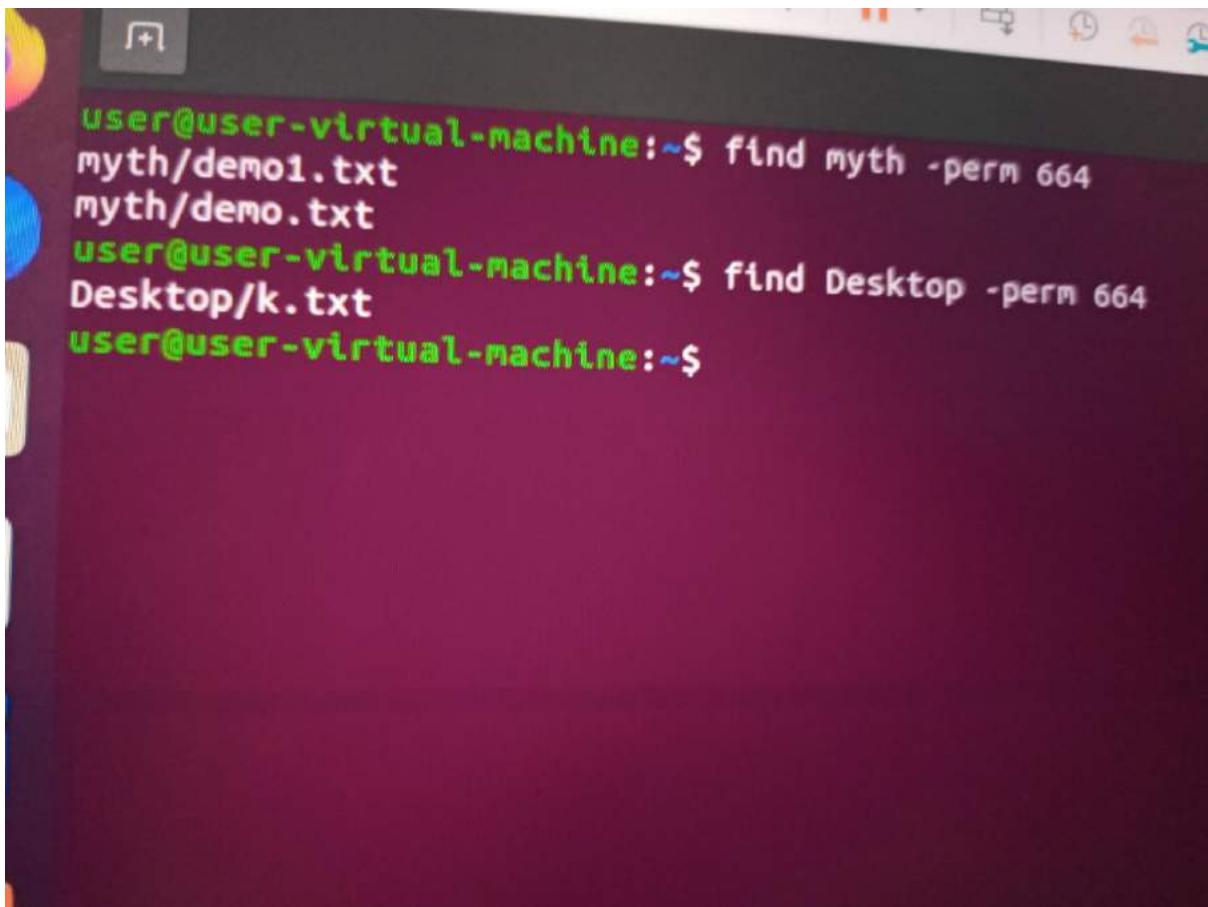


A screenshot of a terminal window titled "Session 1". The window has a dark background and light-colored text. At the top, there's a menu bar with "File", "Edit", "View", "VM", "Tabs", and "Help". Below the menu is a toolbar with various icons. The main area of the terminal shows the following command-line session:

```
user@user-virtual-machine:~$ find myth -name "*.d.txt"
user@user-virtual-machine:~$ find myth -name "d*.txt"
myth/demo1.txt
myth/demo.txt
user@user-virtual-machine:~$
```

Find list of files having specified permissions – Displays all the files under the specified directory having the specified permission

\$ find directory_name -perm 664



A screenshot of a Linux terminal window. The terminal has a dark background with light-colored text. At the top, there are several icons: a yellow circle, a plus sign, a square, a downward arrow, a clock, a person icon, and a gear icon. The terminal prompt is "user@user-virtual-machine:~\$". The user runs two "find" commands. The first command, "find myth -perm 664", finds files named "myth/demo1.txt" and "myth/demo.txt". The second command, "find Desktop -perm 664", finds a file named "Desktop/k.txt". Both commands are run from the root directory (~). The terminal ends with the prompt "user@user-virtual-machine:~\$".

Cp command

The cp command copies a file or a group of files.

The syntax requires two filenames to be specified in the command line.

The first line is copied to the second.

If the destination file(file2) does not exist, it will first be created before copying takes place.

If the destination file already exists and we run cp command, it will be simply overridden without any warning from the system.

```
cp file1 file2
```

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "user@user-virtual-machine:~". The user has run several commands:

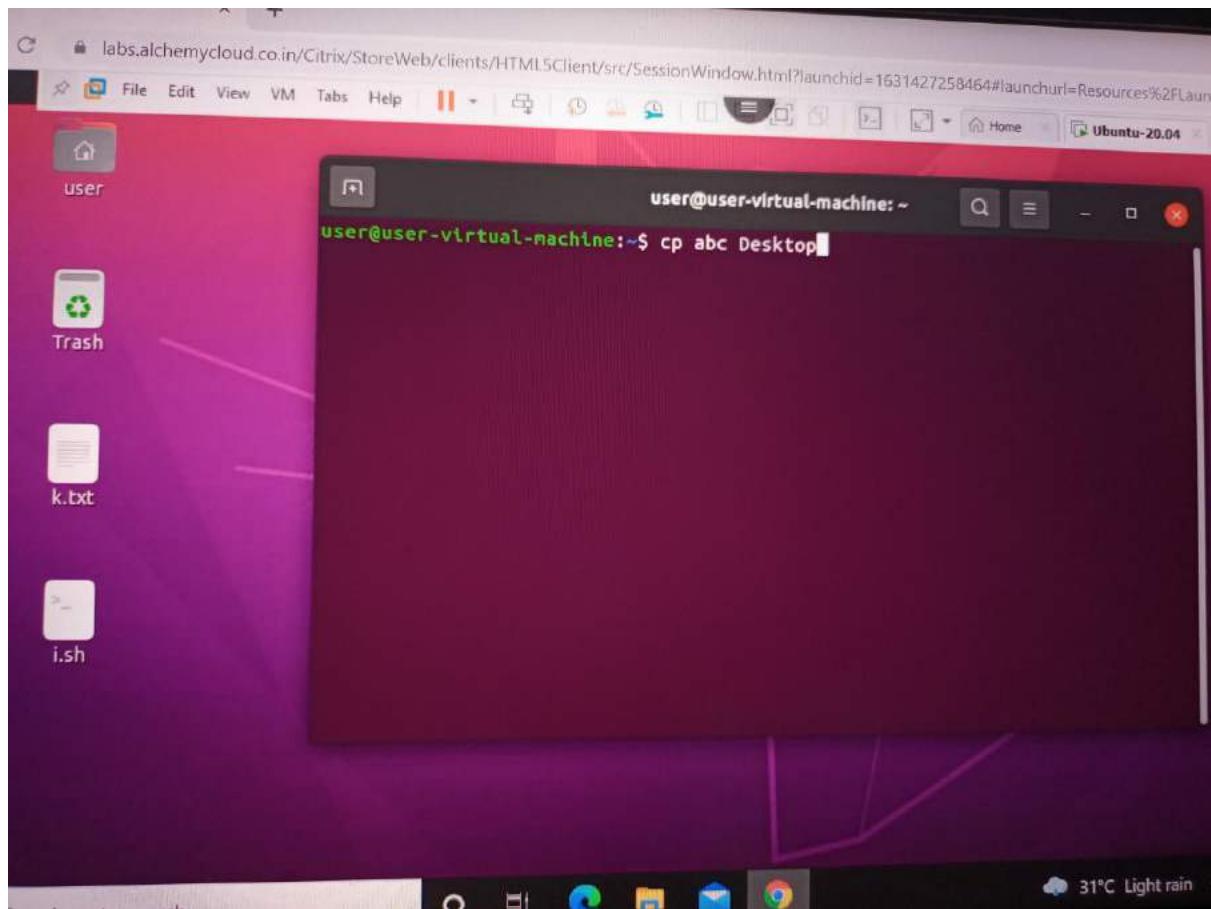
```
user@user-virtual-machine:~$ cat abc
This is file
I am adding more content.
user@user-virtual-machine:~$ cp abc cde
user@user-virtual-machine:~$ ls
abc      cars.txt  Downloads  filee2.txt  hh.txt    Music    ooo.txt  pscmd.txt  shellish  until.sh
a.out    cars.txt  employee.txt filep     img1.jpg  myth    oo.txt   Public    states.txt user.sh
binnn.sh  cde      factorial.sh file.txt  img2.jpg  name    o.txt    q.txt    students  u.sh
binn.txt  check.sh  file1.txt   first.txt  l.sh     names   output1.txt reverse.sh Templates Videos
bin.sh    dd.txt    file2.php   file2.txt  hh.txt   marks   ooooo.txt output2.txt s1.sh    test    yay.sh
blank.txt Desktop  file3.txt   hhh.sh    list.txt new.txt  output.txt s2.sh    test.txt tt.c
capitals.txt Documents file3.txt   hhh.sh    marks   months.txt oooo.txt Pictures  s3.sh
user@user-virtual-machine:~$ cat cde
This is file
I am adding more content.
user@user-virtual-machine:~$
```

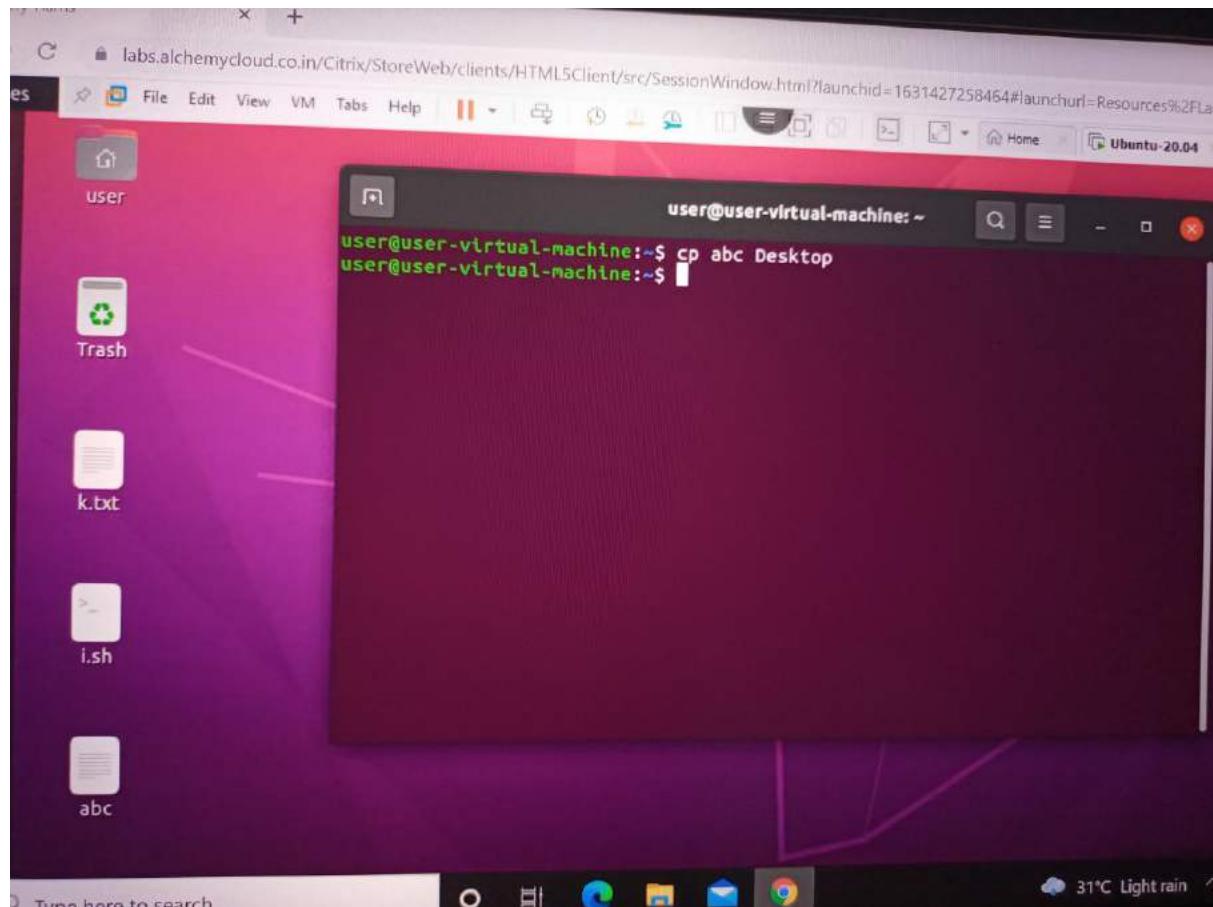
The desktop interface includes a dock with icons for Home, Dash, Activities, and other applications. The system tray shows the date and weather.

If there is only one file to be copied, the destination can be either an ordinary file or directory.

`cp file1 dir1/file2`

`cp file1 dir1`





It can also be used to copy more than one file with a single command.

For this case, the last argument must be a directory which already exists.

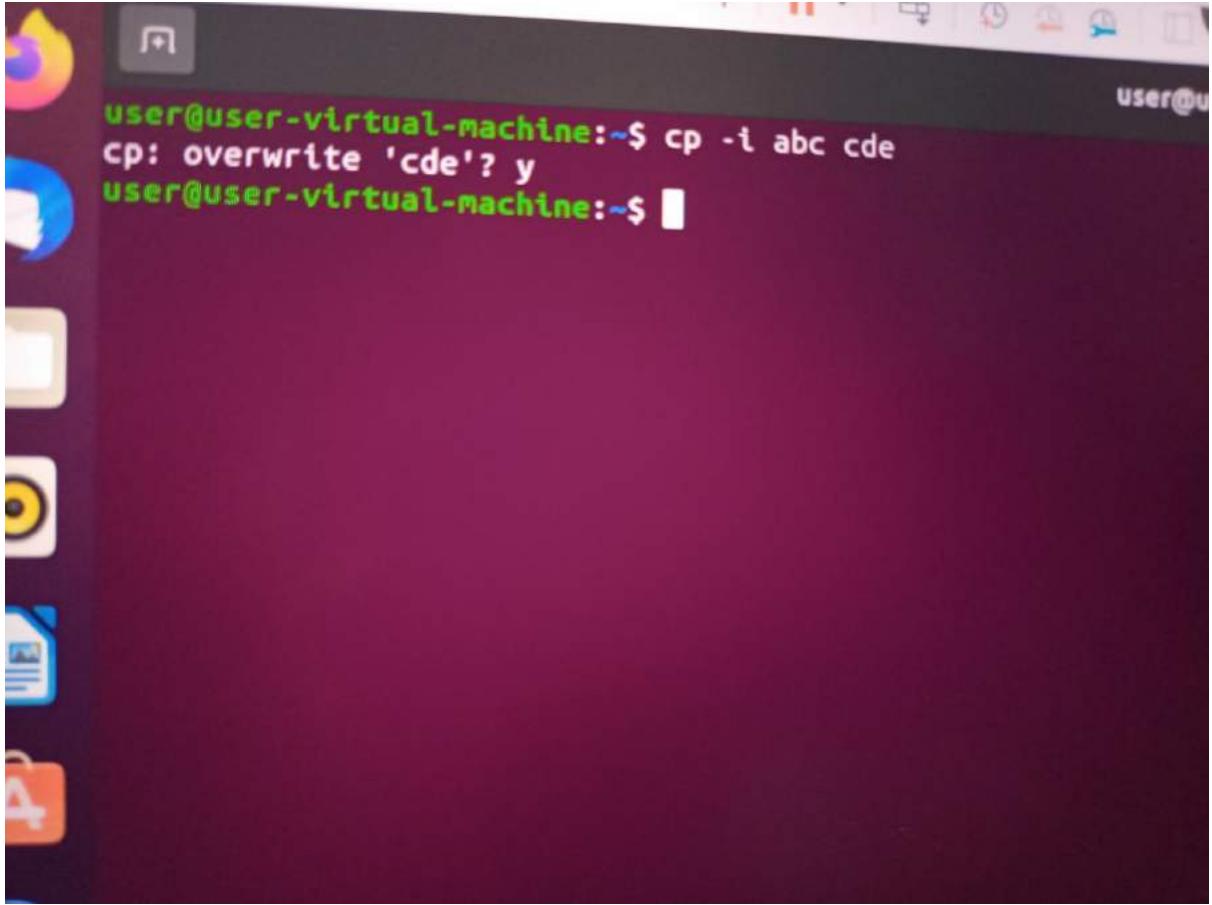
```
cp file1 file2 file3 dir1
```

A screenshot of a Linux desktop environment, likely Ubuntu 20.04, showing a terminal window and several files on the desktop. The terminal window displays the command 'ls' followed by a list of files and directories. The desktop has icons for 'user', 'cde', 'Trash', 'blank.txt', 'k.txt', 'i.sh', and 'abc'. A pink arrow points from the 'blank.txt' icon to the terminal window, indicating it is the source file being copied.

```
user@user-virtual-machine:~$ ls
abc          Desktop   first.txt  myth      Pictures    test
a.out        Documents g1.txt     name      pscommand.txt test.txt
binnn.sh    Downloads  Downloads  hhh.sh    names      Public     tt.c
binn.txt    employee.txt hh.txt    new.txt   q.txt      until.sh
bin.sh      factorial.sh i.sh     ooooo.txt reverse.sh user.sh
blank.txt   file1.txt   img1.jpg  ooo.txt  s1.sh      u.sh
capitals.txt file2.php  img2.jpg  oo.txt   s2.sh      Videos
cars.txt    file2.txt   i.sh     oo.txt   s3.sh      yay.sh
cars.txt    file3.txt   list.txt  o.txt    shell1.sh
cde         filee2.txt  marks    output1.txt states.txt
check.sh    filep      months.txt output2.txt students
dd.txt      file.txt   Music    output.txt Templates
user@user-virtual-machine:~$ cp cde blank.txt Desktop
user@user-virtual-machine:~$
```

`cp -i(interactive)`

This option warns the user before overwriting the destination file.

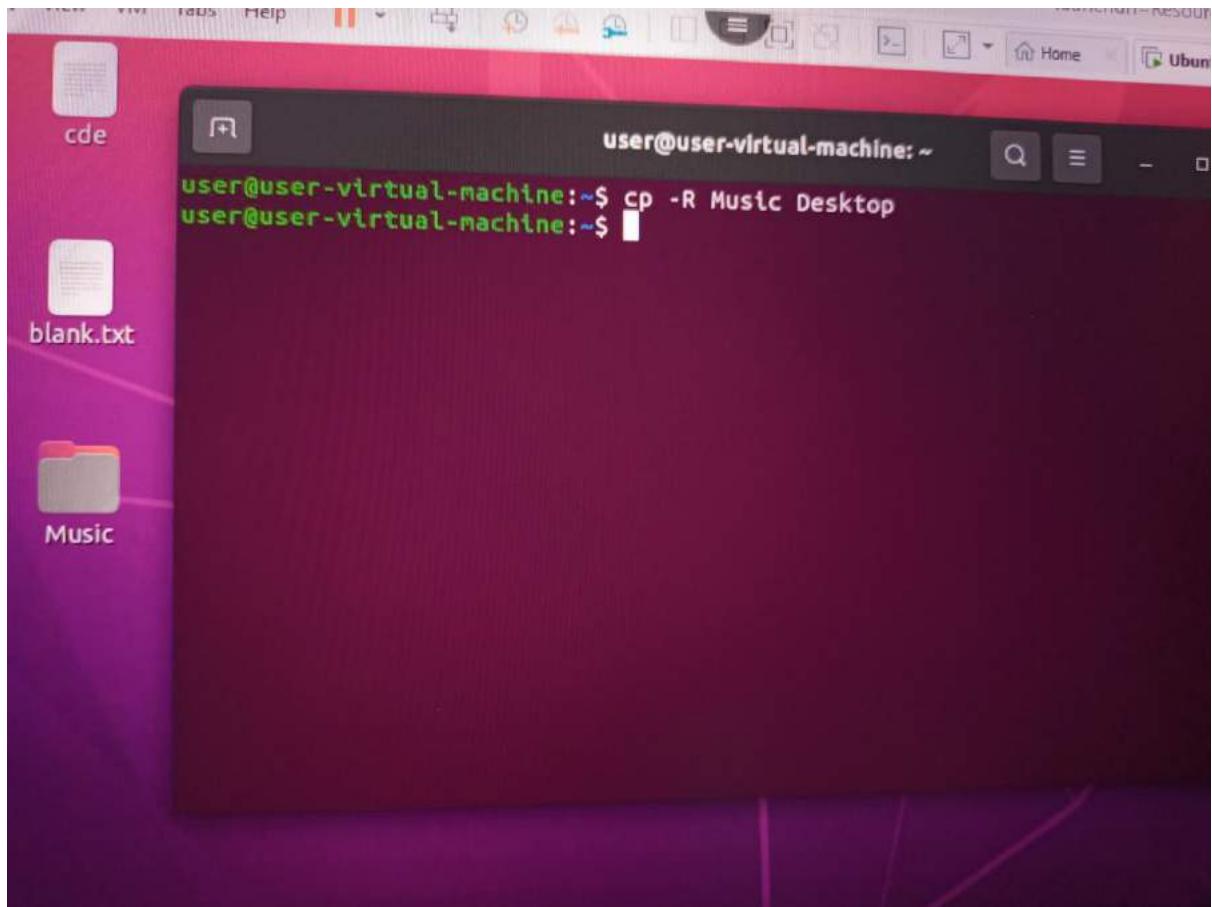


A screenshot of a Linux terminal window titled "User@user-virtual-machine:~\$". The terminal shows the command "cp -i abc cde" being run, followed by a confirmation prompt "cp: overwrite 'cde'? y". The terminal has a dark background with light-colored text. The window title bar includes the user name and session information. To the left of the terminal, there is a vertical dock with several icons, including a yellow flower, a blue folder, a white document, and a red folder.

```
user@user-virtual-machine:~$ cp -i abc cde
cp: overwrite 'cde'? y
user@user-virtual-machine:~$
```

cp -R

This option behaves recursively to copy an entire directory structure.



mv command

It renames a file or a directory.

It moves a group of files to a different directory.

`mv src dest`

The screenshot shows a terminal window on an Acer laptop with a dark background. The terminal output is as follows:

```
user@user-virtual-machine:~$ cat abc
This is file
I am adding more content.
user@user-virtual-machine:~$ cat xyz
cat: xyz: No such file or directory
user@user-virtual-machine:~$ mv abc xyz
user@user-virtual-machine:~$ ls
a.out      cars.txt    employee.txt   filep   img1.jpg   myth     oo.txt      Public   states.txt  user.sh
binnn.sh   cde        factorial.sh  file.txt img2.jpg   name     o.txt       q.txt    students   u.sh
binn.txt   check.sh   file1.txt    first.txt i.sh     names    output1.txt reverse.sh Templates  Videos
bin.sh     dd.txt     file2.php   file2.txt g1.txt   list.txt new.txt    output2.txt s1.sh    test     xyz
blank.txt  Desktop   file2.txt   hh.txt   hhhh.sh   marks   ooooo.txt output.txt s2.sh    test.txt  yay.sh
capitals.txt Documents file3.txt   hh.txt   hhh.sh   months.txt oooo.txt Pictures   s3.sh    pscommand.txt shell1.sh until1.sh
carrs.txt  Downloads filee2.txt  hh.txt   Hustic   oo.txt      Hustic   oo.txt   Hustic   oo.txt
user@user-virtual-machine:~$ cat xyz
This is file
I am adding more content.
user@user-virtual-machine:~$ mv xyz abc
user@user-virtual-machine:~$ ls
abc      carrs.txt  Downloads   file2.txt hh.txt   Hustic   oo.txt      pscommand.txt shell1.sh until1.sh
a.out    cars.txt   employee.txt filep   img1.jpg   myth     oo.txt      Public   states.txt  user.sh
binnn.sh cde        factorial.sh file.txt img2.jpg   name     o.txt       q.txt    students   u.sh
binn.txt check.sh   file1.txt   first.txt i.sh     names    output1.txt reverse.sh Templates  Videos
bin.sh   dd.txt     file2.php   file2.txt g1.txt   list.txt new.txt    output2.txt s1.sh    test     xyz
blank.txt Desktop   file2.txt   hh.txt   hhhh.sh   marks   ooooo.txt output.txt s2.sh    test.txt  yay.sh
capitals.txt Documents file3.txt   hh.txt   hhh.sh   months.txt oooo.txt Pictures   s3.sh    tt.c
user@user-virtual-machine:~$ cat abc
This is file
I am adding more content.
```

It doesn't create a copy of file, it merely renames it.

No additional space is consumed on disk during renaming.

mv can also be used to move a file to a directory.

mv src dirname

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "user@user-virtual-machine:~". Inside the terminal, the user runs several commands:

```
user@user-virtual-machine:~$ ls
a.out    cars.txt   Downloads  File3.txt  hh.sh    marks    new.txt  output1.txt  q.txt    states.txt  until.sh
binnn.sh  cde      employee.txt filee2.txt hh.txt   months.txt  ooooo.txt  output2.txt  reverse.sh  students   user.sh
binn.txt  check.sh factorial.sh  file.txt   img1.jpg  Music     oooo.txt   output.txt   s1.sh    Templates  u.sh
bin.sh    dd.txt   file1.txt    first.txt  iMg2.jpg  myth     ooo.txt    Pictures   s2.sh    test      Videos
capitals.txt Desktop  file2.php  g1.txt   i.sh     name     oo.txt    pscommand.txt s3.sh    test.txt  yay.sh
carrs.txt Documents file2.txt  hhhh.sh  list.txt  names    o.txt     Public    shelli.sh  tt.c
user@user-virtual-machine:~$ mv cars.txt Desktop
user@user-virtual-machine:~$ ls
a.out    cde      employee.txt filee2.txt  hh.txt   months.txt  ooooo.txt  output2.txt  reverse.sh  students   user.sh
binnn.sh  check.sh factorial.sh  file.txt   img1.jpg  Music     oooo.txt   output.txt   s1.sh    Templates  u.sh
binn.txt  dd.txt   file1.txt    first.txt  iMg2.jpg  myth     ooo.txt    Pictures   s2.sh    test      Videos
capitals.txt Desktop  file2.php  g1.txt   i.sh     name     oo.txt    pscommand.txt s3.sh    test.txt  yay.sh
carrs.txt Documents file2.txt  hhhh.sh  list.txt  names    o.txt     Public    shelli.sh  tt.c
user@user-virtual-machine:~$ mv -i cars.txt Desktop
user@user-virtual-machine:~$
```

The desktop interface includes a search bar at the top, a dock with icons for Home, Dash, Applications, and Help, and a system tray showing weather (31°C Light rain), battery level (48%), and date/time (12-09-2021 12:44).

mv -i

This command ask the user for confirmation before moving or renaming each file.

The screenshot shows a terminal window titled "Ubuntu-20.04" running on an Acer laptop. The terminal displays the following command-line session:

```
user@user-virtual-machine:~$ cat abc
user@user-virtual-machine:~$ cat you
user@user-virtual-machine:~$ mv -i abc you
mv: overwrite 'you'? y
user@user-virtual-machine:~$ ls
a.out      dd.txt     file2.php   hhhh.sh  marks    ooooo.txt  output.txt  s2.sh    test.txt  you
binnn.sh   Desktop    file2.txt   hhh.sh   months.txt  oooo.txt  Pictures   s3.sh    tt.c
binn.txt   Documents  file3.txt  hh.txt   Music     ooo.txt   pscommand.txt shell1.sh until.sh
bin.sh     Downloads  filee2.txt img1.jpg myth     oo.txt    Public    states.txt user.sh
capitals.txt employee.txt file.txt  img2.jpg name     o.txt     q.txt    students  u.sh
carrs.txt  factorial.sh first.txt i.sh    names    output1.txt reverse.sh Templates Videos
check.sh   file1.txt  g1.txt   list.txt new.txt  output2.txt si.sh    test    yay.sh
user@user-virtual-machine:~$
```

mv -f(force)

It prompts for confirmation overwriting the destination file if a file is write protected .

mv -b

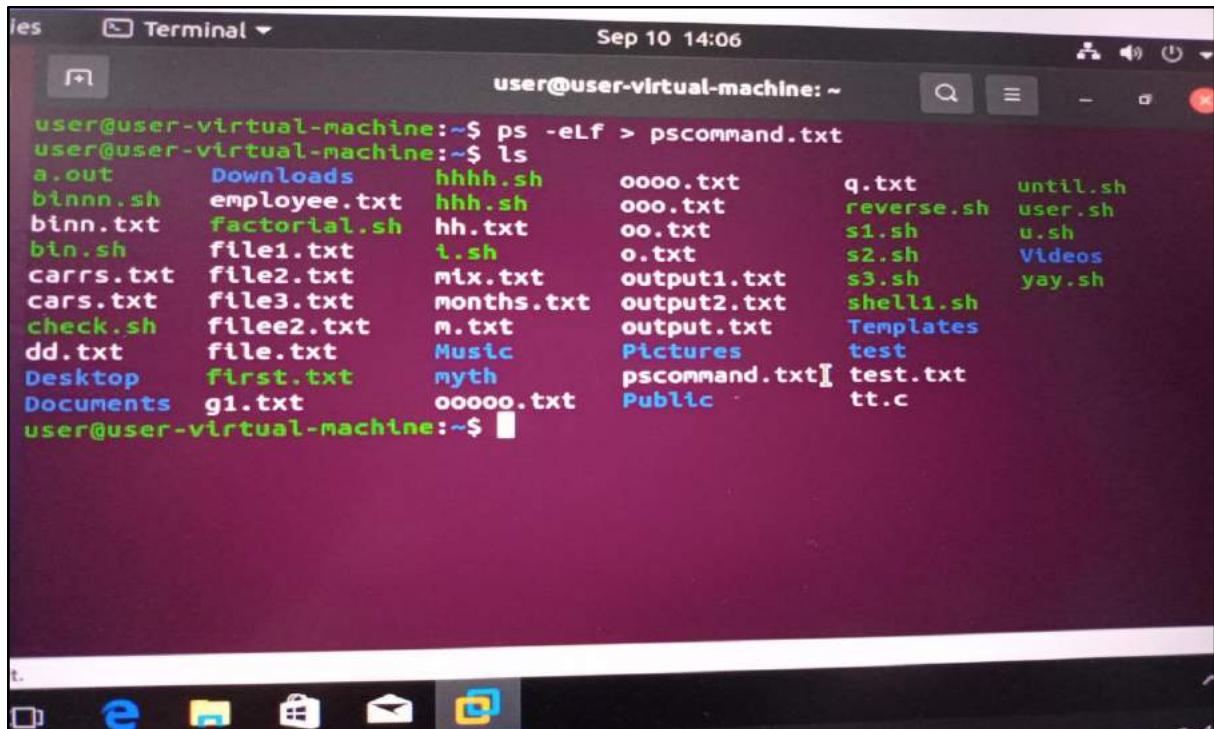
With this option it is easier to take a backup of an existing file that will be overwritten as a result of the mv command.

Less Command

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.

```
ps -eLf > pscommand.txt
```

By this command, we are entering the data of "ps -eLf" in the file(pscommand.txt)



```
Terminal Sep 10 14:06 user@user-virtual-machine: ~
user@user-virtual-machine:~$ ps -eLf > pscommand.txt
user@user-virtual-machine:~$ ls
a.out      Downloads      hhh.sh      oooo.txt      q.txt      until.sh
binnn.sh   employee.txt  hhh.sh      ooo.txt      reverse.sh  user.sh
binnn.txt  factorial.sh hh.txt      oo.txt      s1.sh      u.sh
bin.sh     file1.txt    i.sh       o.txt       s2.sh      Videos
carrs.txt  file2.txt    mix.txt    months.txt  output1.txt  s3.sh      yay.sh
cars.txt   file3.txt    mix.txt    months.txt  output2.txt  shell1.sh
check.sh   filee2.txt   m.txt     myth        output.txt   Templates
dd.txt     file.txt     Music      Public      pscommand.txt test
Desktop   first.txt    myth      Public      Public      tt.c
Documents g1.txt      ooooo.txt
user@user-virtual-machine:~$
```

```
cat pscommand.txt
```

To view the contents of the file.

```
root      20625      1  20632  0   3 00:00 ?    00:00:00 /usr/sbin/cups
-browsed  20625      1  20633  0   3 00:00 ?    00:00:00 /usr/sbin/cups
-browsed  29026      2  29026  0   1 06:17 ?    00:00:00 [xfsalloc]
root      29027      2  29027  0   1 06:17 ?    00:00:00 [xfs_mru_cache]
]
root      29031      2  29031  0   1 06:17 ?    00:00:00 [jfsIO]
root      29032      2  29032  0   1 06:17 ?    00:00:00 [jfsCommit]
root      29033      2  29033  0   1 06:17 ?    00:00:00 [jfsCommit]
root      29034      2  29034  0   1 06:17 ?    00:00:00 [jfsSync]
root      32654      2  32654  0   1 10:06 ?    00:00:37 [kworker/1:1-c
group_destroy]
root      32767      2  32767  0   1 11:34 ?    00:00:04 [kworker/0:2-e
vents]
root      32922      2  32922  0   1 13:44 ?    00:00:01 [kworker/u256:
0-events_power_efficient]
root      32925      2  32925  0   1 13:49 ?    00:00:00 [kworker/u256:
1-events_unbound]
root      32938      2  32938  0   1 13:58 ?    00:00:00 [kworker/0:1-m
pt_poll_0]
root      32939      2  32939  0   1 13:58 ?    00:00:01 [kworker/1:0-e
vents]
user     32992  19759  32992  0   1 14:05 pts/0  00:00:00 ps -elf
user@user-virtual-machine:~$
```

We will be redirected to the end of line .Also,we don't have any feature to go to any particular line or word. But in less command all these features are available.

less pscommand.txt

It will view the content of the file.

```
Ubuntu-20.04 x
root      3      2      3  0   1 Sep09 ?    00:00:00 [rcu_gp]
root      4      2      4  0   1 Sep09 ?    00:00:00 [rcu_par_gp]
root      6      2      6  0   1 Sep09 ?    00:00:00 [kworker/0:0-H-
events_highpri]
root      8      2      8  0   1 Sep09 ?    00:00:00 [mm_percpu_wq]
root      9      2      9  0   1 Sep09 ?    00:00:00 [rcu_tasks_rud
e_]
root     10      2     10  0   1 Sep09 ?    00:00:00 [rcu_tasks_tr
ce]
root     11      2     11  0   1 Sep09 ?    00:00:03 [ksoftirqd/0]
root     12      2     12  0   1 Sep09 ?    00:01:41 [rcu_sched]
root     13      2     13  0   1 Sep09 ?    00:00:01 [migration/0]
root     14      2     14  0   1 Sep09 ?    00:00:00 [idle_inject/0
]
root     16      2     16  0   1 Sep09 ?    00:00:00 [cpuhp/0]
root     17      2     17  0   1 Sep09 ?    00:00:00 [cpuhp/1]
root     18      2     18  0   1 Sep09 ?    00:00:00 [idle_inject/1
]
root     19      2     19  0   1 Sep09 ?    00:00:12 [migration/1]
root     20      2     20  0   1 Sep09 ?    00:00:05 [ksoftirqd/1]
root     22      2     22  0   1 Sep09 ?    00:00:00 [kworker/1:0-H-
events_highpri]
root     23      2     23  0   1 Sep09 ?    00:00:00 [kdevtmpfs]
root     24      2     24  0   1 Sep09 ?    00:00:00 [netns]
pscommand.txt
```

2. Exit from less

If you are not used to of less command, you might struggle to find how to exit less. Trust me it's not at all complicated. Simply press 'q' at any given point to exit from less.

3. Moving around in less

The output of less is divided into sort of pages. You'll see only the text that fills up to your terminal screen.

Up arrow – Move one line up

Down arrow – Move one line down

Space or PgDn – Move one page down

b or PgUp – Move one page up

g – Move to the beginning of the file

G – Move to the end of the file

ng – Move to the nth line

If we want to go page by page downwards, we can press space.

If we want to go page by page upwards, we can press b key.

Every time space is pressed different content on the page is shown.

5. Finding text in less

If you have a large text file, it's better to search for a specific piece of text rather than reading it line by line in order to find it manually.

To find a word or phrase or even a regex pattern, press / and type whatever you want to find.

/pattern

Example

/usr (type /pattern and press enter,keep on pressing n it will take us downwards and we will get that pattern ,once the pattern finishes,it will show END)

```

top - 3 Sep 09 16:00:00 up 1 day, 10:00, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 101 total, 1 running, 100 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0%us, 0.0%sy, 0.0%ni, 0.0%id, 100.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem:  16384K total,  15984K used,   400K free,  1024K buffers
Swap:  1024K total,      0K used,  1024K free,   128K cached

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+   COMMAND
  704 root      20   0    704     0   3 Sep09 ?        00:00:15 /usr/sbin/Netw
  704 root      20   0    704     0   3 Sep09 ?        00:00:04 /usr/sbin/Netw
  704 root      20   0    704     0   3 Sep09 ?        00:00:03 /usr/sbin/Netw
  704 root      20   0    704     0   3 Sep09 ?        00:00:02 /usr/sbin/Netw
  710 root      20   0    710     0   2 Sep09 ?        00:00:12 /usr/sbin/irqb
  710 lance    20   0    710     0   2 Sep09 ?        00:00:00 /usr/sbin/irqb
  711 root      20   0    711     0   1 Sep09 ?        00:00:01 /usr/bin/python
n3 /usr/bin/networkd-dispatcher --run-startup-triggers 713 root      20   0    713     0   3 Sep09 ?        00:00:05 /usr/lib/polici
  713 ykit-1/polkitd --no-debug 713 root      20   0    713     0   3 Sep09 ?        00:00:00 /usr/lib/polici
  713 ykit-1/polkitd --no-debug 713 root      20   0    713     0   3 Sep09 ?        00:00:01 /usr/lib/polici
  713 ykit-1/polkitd --no-debug 716 syslog   20   0    716     0   4 Sep09 ?        00:00:00 /usr/sbin/rsys
  716 logd -n -INONE 716 syslog   20   0    716     0   4 Sep09 ?        00:00:00 /usr/sbin/rsys
  716 logd -n -INONE 716 syslog   20   0    716     0   4 Sep09 ?        00:00:00 /usr/sbin/rsys
  716 syslog   20   0    716     0   4 Sep09 ?        00:00:00 /usr/sbin/rsys
:

```

Now if we want to search a pattern from bottom to upward direction, we use

?pattern

Example

?usr

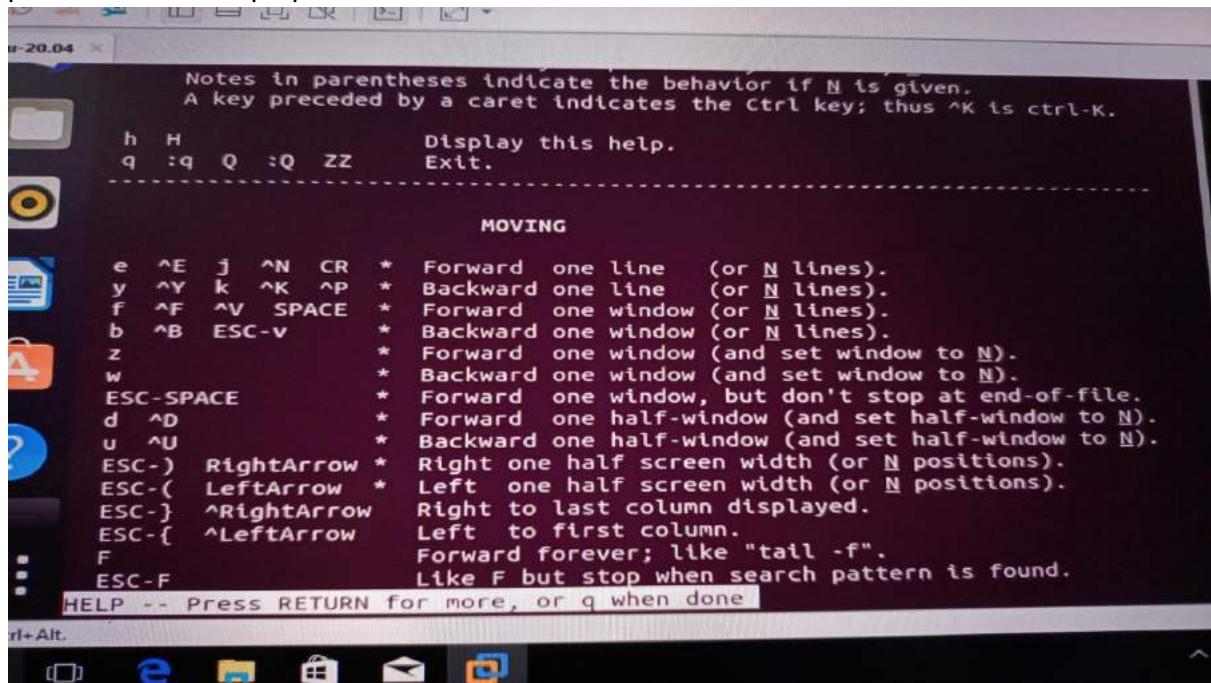
```

top - 3 Sep 09 16:00:00 up 1 day, 10:00, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 101 total, 1 running, 100 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0%us, 0.0%sy, 0.0%ni, 0.0%id, 100.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem:  16384K total,  15984K used,   400K free,  1024K buffers
Swap:  1024K total,      0K used,  1024K free,   128K cached

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+   COMMAND
  20625 root      20   0    20633     0   3 00:00:00 ?        00:00:00 /usr/sbin/cups
  29026 root      20   0    29026     0   1 06:17 ?        00:00:00 [xfsalloc]
  29027 root      20   0    29027     0   1 06:17 ?        00:00:00 [xfs_mru_cache]
  29031 root      20   0    29031     0   1 06:17 ?        00:00:00 [jfsIO]
  29032 root      20   0    29032     0   1 06:17 ?        00:00:00 [jfsCommit]
  29033 root      20   0    29033     0   1 06:17 ?        00:00:00 [jfsCommit]
  29034 root      20   0    29034     0   1 06:17 ?        00:00:00 [jfsSync]
  32654 root      20   0    32654     0   1 10:06 ?        00:00:37 [kworker/1:1-c
group_destroy]
  32767 root      20   0    32767     0   1 11:34 ?        00:00:04 [kworker/0:2-e
vents]
  32922 root      20   0    32922     0   1 13:44 ?        00:00:01 [kworker/u256:
0-events_power_efficient]
  32925 root      20   0    32925     0   1 13:49 ?        00:00:00 [kworker/u256:
1-events_unbound]
  32938 root      20   0    32938     0   1 13:58 ?        00:00:00 [kworker/0:1-m
pt_poll_0]
  32939 root      20   0    32939     0   1 13:58 ?        00:00:01 [kworker/1:0-e
vents]
  32992 user      20   0    32992     0   1 14:05 pts/0        00:00:00 ps -elf
~
~
~
```

We will get the last line where this pattern is present nad now by pressing n we can go upwards.

press h to view help option



The screenshot shows a terminal window with the title "u-20.04". The screen displays the help menu for the less command. At the top, it says "Notes in parentheses indicate the behavior if N is given. A key preceded by a caret indicates the Ctrl key; thus ^K is ctrl-K." Below this, there is a table of keyboard shortcuts for moving around the text. The table has two columns: the key or sequence of keys and their corresponding actions. The actions often include descriptions involving "Forward" or "Backward" movement, "one line", "one window", or "half-screen width". Some entries also mention "or N lines", "or N positions", or "and set window to N". At the bottom of the help menu, there is a prompt: "HELP -- Press RETURN for more, or q when done".

h H	Display this help.
q :q Q :Q ZZ	Exit.
MOVING	
e ^E j ^N CR	* Forward one line (or N lines).
y ^Y k ^K ^P	* Backward one line (or N lines).
f ^F ^V SPACE	* Forward one window (or N lines).
b ^B ESC-v	* Backward one window (or N lines).
z	* Forward one window (and set window to N).
w	* Backward one window (and set window to N).
ESC-SPACE	* Forward one window, but don't stop at end-of-file.
d ^D	* Forward one half-window (and set half-window to N).
u ^U	* Backward one half-window (and set half-window to N).
ESC-) RightArrow	* Right one half screen width (or N positions).
ESC-(LeftArrow	* Left one half screen width (or N positions).
ESC-] ^RightArrow	Right to last column displayed.
ESC-[^LeftArrow	Left to first column.
F	Forward forever; like "tail -f".
ESC-F	Like F but stop when search pattern is found.

6. View multiple files with less command

To open multiple files with less, simply input the file names one by one:

less <filename1> <filename2> <filename3>

:n to move next file

:p to move to previous file

7. less -X file name

save the file name on screen when file exits

8.less -S file name

long line can be seen by side wrapping.

Example

less -S pscommand.txt

A screenshot of a Linux desktop environment. In the top right corner, there's a system tray icon showing the date "Sep 10 14:48". Below the desktop icons, a weather widget displays "30°C Light rain". The main focus is a terminal window titled "Terminal" with the command "user@user-virtual-machine: ~". The terminal is displaying a process list with columns: UID, PID, PPID, LWP, C, NLWP, STIME, TTY, TIME, and CMD. The processes listed are all owned by "root" and have a start time of "Sep09 ?". The commands listed include "/sbin/init", "[kthreadd]", "[rcu_gp]", "[rcu_par_gp]", "[kworker/0:0H]", "[mm_percpu_wq]", "[rcu_tasks_ru]", "[rcu_tasks_tr]", "[ksoftirqd/0]", "[rcu_sched]", "[migration/0]", "[idle_inject/0]", "[cpuhp/0]", "[cpuhp/1]", "[idle_inject/1]", "[migration/1]", "[ksoftirqd/1]", "[kworker/1:0H]", "[kdevtmpfs]", and "[netns]".

9.less --version to know the version of less

A screenshot of a terminal window titled "Terminal" with the command "user@user-virtual-machine: ~". The user runs the command "less --version". The output shows the following information:
less 551 (GNU regular expressions)
Copyright (C) 1984-2019 Mark Nudelman

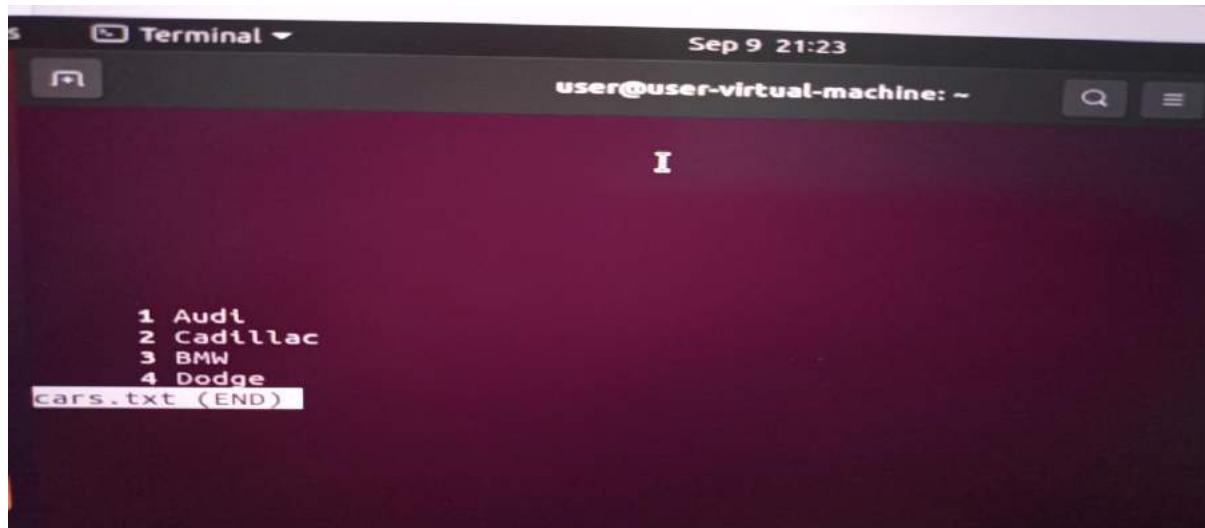
less comes with NO WARRANTY, to the extent permitted by law.
For information about the terms of redistribution,
see the file named README in the less distribution.
Home page: <http://www.greenwoodsoftware.com/less>
user@user-virtual-machine: ~

10. less -m file name
show more detailed prompt including file position

```
less -N filename
```

Example

```
less -N cars.txt
```



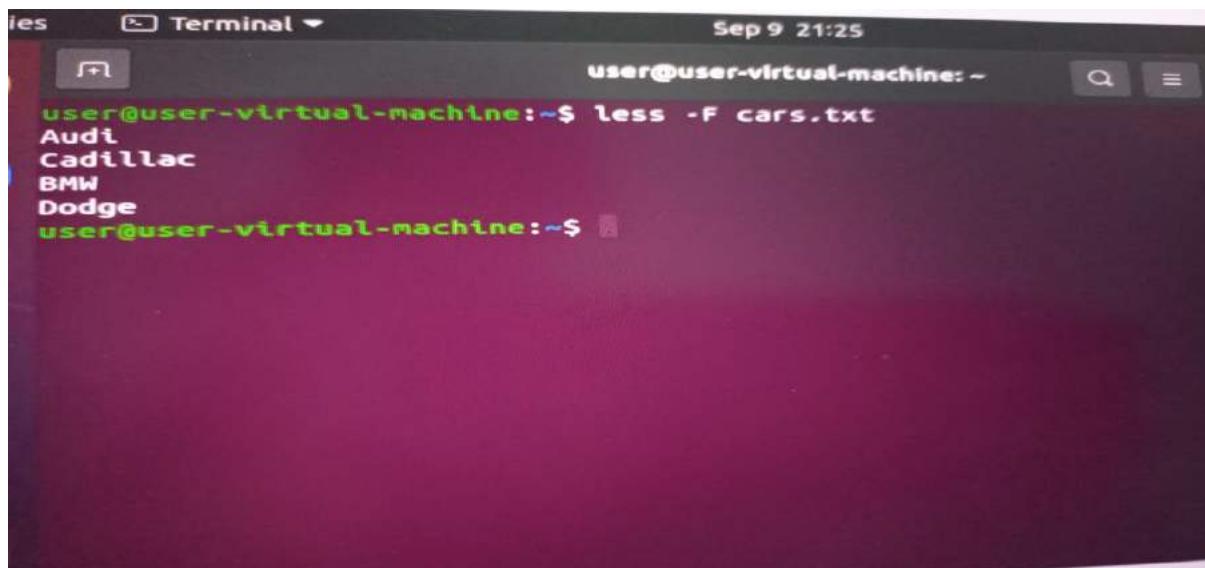
```
Terminal Sep 9 21:23 user@user-virtual-machine: ~ I
1 Audi
2 Cadillac
3 BMW
4 Dodge
cars.txt (END)
```

```
less -F filename
```

This option causes less to exit if entire file can be displayed on first screen.

Example

```
less -F cars.txt
```



```
Terminal Sep 9 21:25 user@user-virtual-machine: ~
user@user-virtual-machine:~$ less -F cars.txt
Audi
Cadillac
BMW
Dodge
user@user-virtual-machine:~$
```

The above cars.txt file contents is displayed on firsts screen so, the less exit the file automatically.

```
less -s filename
```

It causes consecutive blank lines to be squeezed into single blank line.

A screenshot of a terminal window titled "Terminal" on an Ubuntu 20.04 desktop. The window shows the command "user@user-virtual-machine: ~\$ cat blank.txt" and its output: "hello smriti" on one line, followed by two blank lines, then "i know you" on another line. The terminal interface includes a dock with various icons like Dash, Home, and Applications, and a system tray at the bottom.

after typing less -s blank.txt

A screenshot of a terminal window titled "Terminal" on an Ubuntu 20.04 desktop. The window shows the command "user@user-virtual-machine: ~\$ cat blank.txt" and its output: "hello smriti" on one line, followed by a single blank line, then "i know you" on another line. The terminal interface includes a dock with various icons like Dash, Home, and Applications, and a system tray at the bottom.

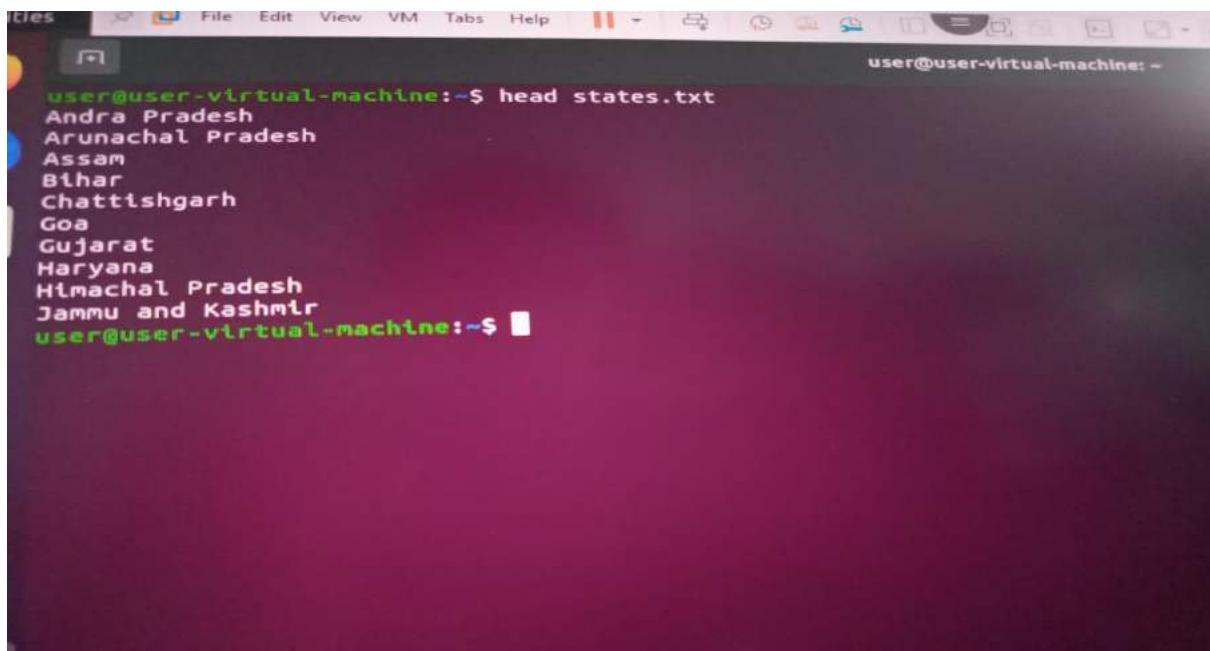
Head command

It prints the top N number of lines of the given input.

By default it prints the first 10 lines of the specified files.

If more than one file name is provided then data from each file is preceded by its file name.

```
head states.txt
```



```
user@user-virtual-machine:~$ head states.txt
Andra Pradesh
Arunachal Pradesh
Assam
Bihar
Chattishgarh
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
user@user-virtual-machine:~$
```

head file.txt file2.txt file3.txt

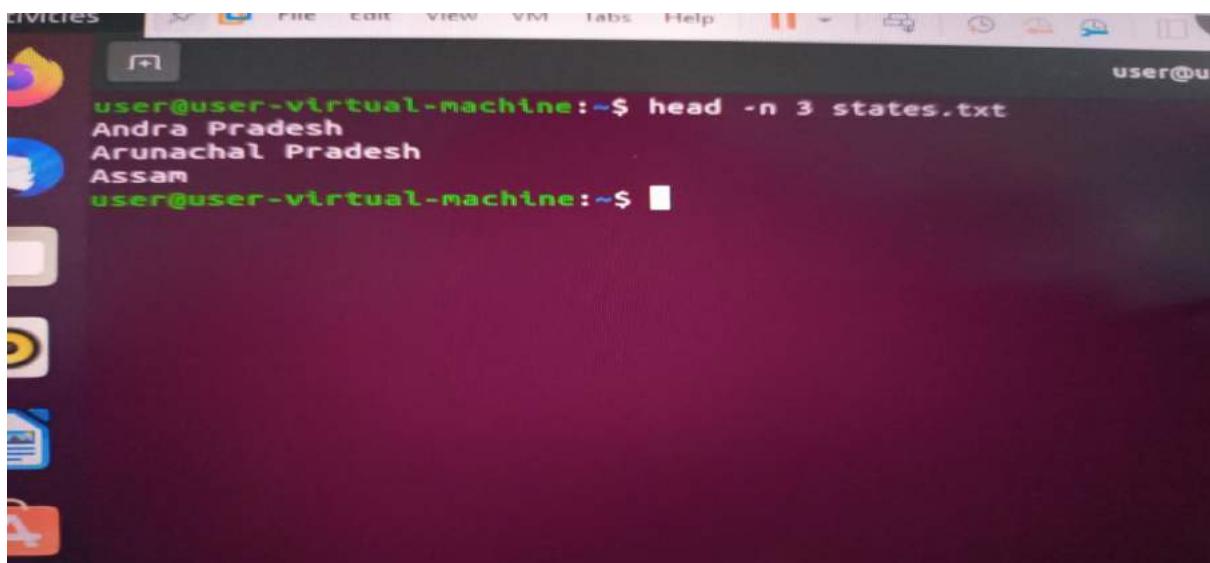
By default, it prints the first 10 lines of the specified files. If more than one file name is provided then data from each file is preceded by its file name.

head -n

It prints the first "num" lines instead of first 10 lines.

num is mandatory to be specified in command otherwise it displays an error.

head -n 3 states.txt



```
user@user-virtual-machine:~$ head -n 3 states.txt
Andra Pradesh
Arunachal Pradesh
Assam
user@user-virtual-machine:~$
```

\$head -n -5 file.txt ---It prints all rows except last 5 row in the file

```
user@user-virtual-machine:~$ head -n -5 states.txt
Andra Pradesh
Arunachal Pradesh
Assam
Bihar
Chattishgarh
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
Madhya pradesh
Maharastra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Punjab
Rajasthan
Sikkim
Tamil Nadu
user@user-virtual-machine:~$
```

head -n 0 file.txt

It prints nothing. Because we gave 0 lines. Hence it shows nothing.

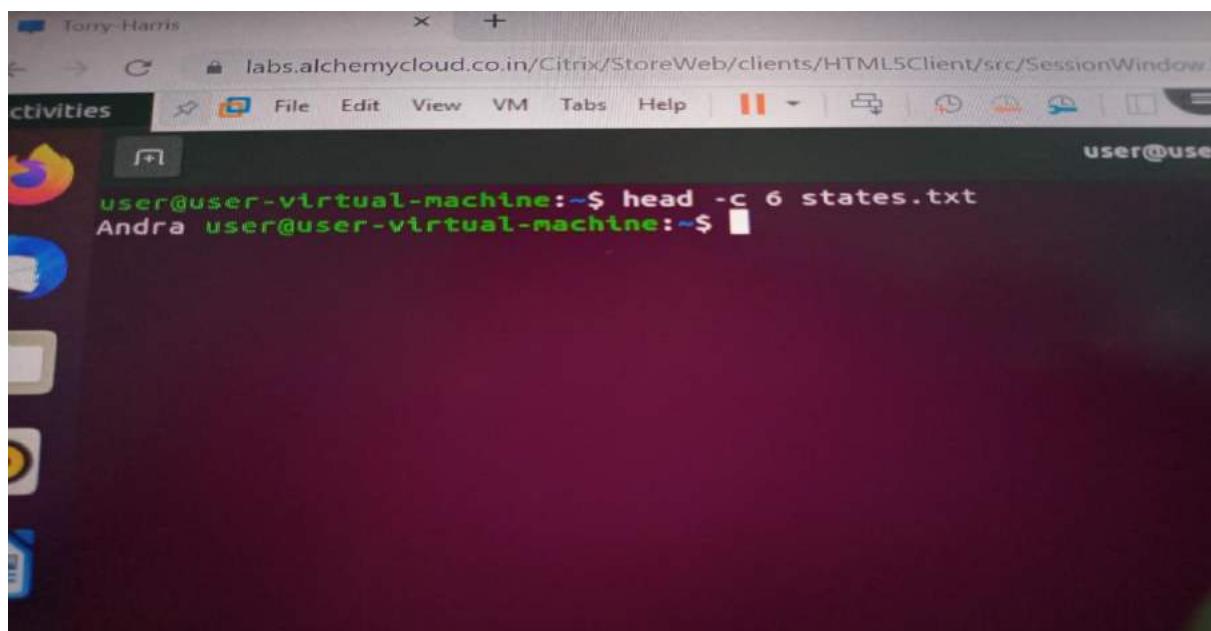
\$head -n -0 file.txt ---It prints all whatever the file consists of (inverse of - previous command)

```
user@user-virtual-machine:~$ head -n 0 states.txt
Andra Pradesh
Arunachal Pradesh
Assam
Bihar
Chattishgarh
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
Madhya pradesh
Maharastra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
user@user-virtual-machine:~$
```

```
head -c
```

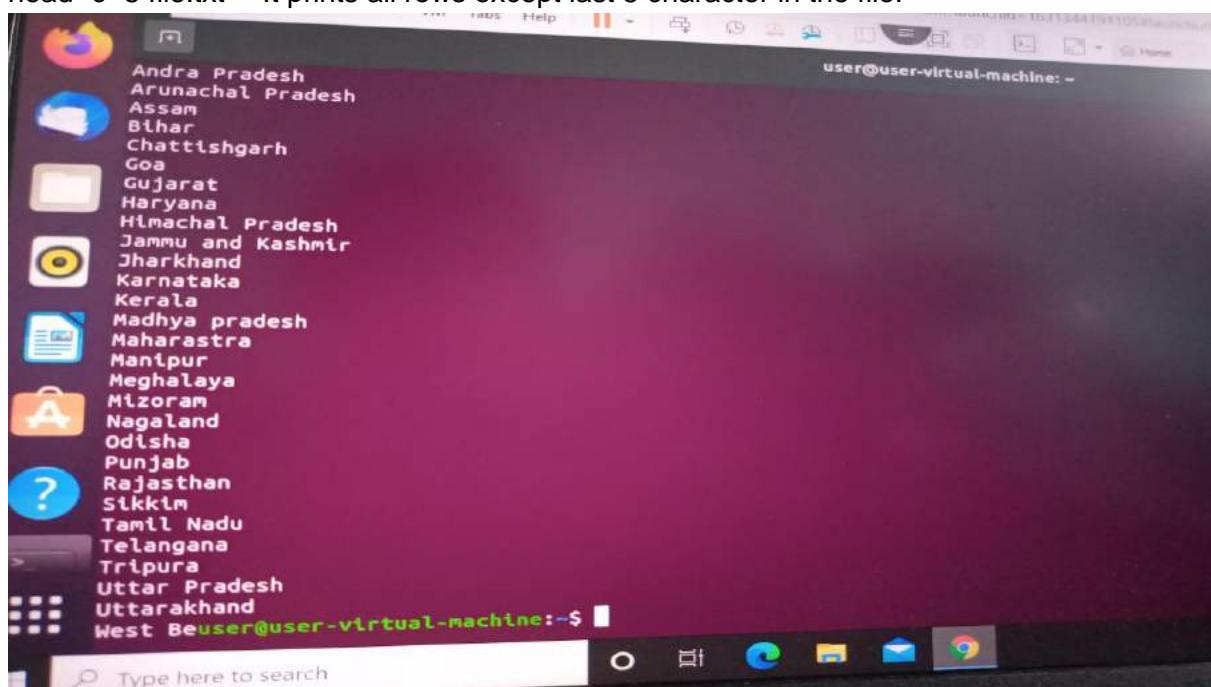
It prints the first "num" bytes from the file specified.

```
head -c 6 states.txt
```



A screenshot of a terminal window titled 'Terry-Harris'. The window shows a command-line interface with the following text:
user@user-virtual-machine:~\$ head -c 6 states.txt
Andra user@user-virtual-machine:~\$ █

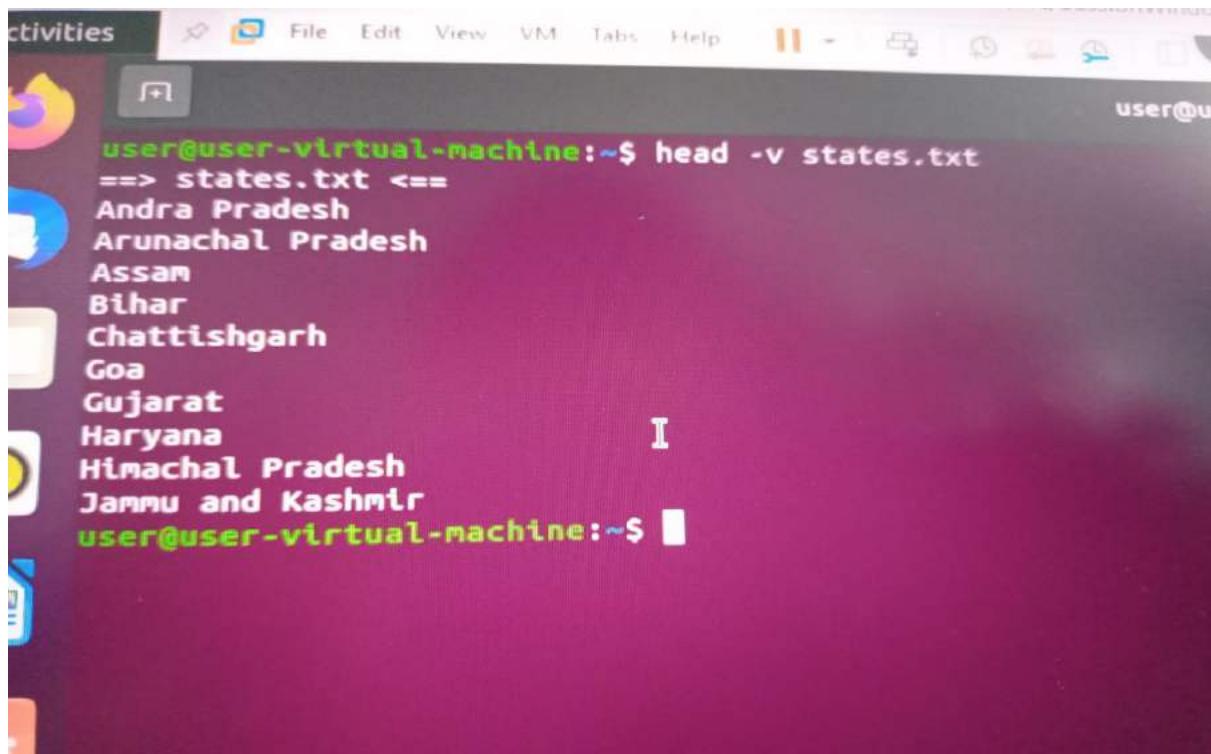
```
head -c -5 file.txt ---It prints all rows except last 5 character in the file.
```



A screenshot of a terminal window showing a list of Indian states. The list includes: Andhra Pradesh, Arunachal Pradesh, Assam, Bihar, Chattishgarh, Goa, Gujarat, Haryana, Himachal Pradesh, Jammu and Kashmir, Jharkhand, Karnataka, Kerala, Madhya pradesh, Maharashtra, Manipur, Meghalaya, Mizoram, Nagaland, Odisha, Punjab, Rajasthan, Sikkim, Tamil Nadu, Telangana, Tripura, Uttar Pradesh, Uttarakhand, West Bengal. The command 'head -c -5 file.txt' was run at the prompt.

```
head -v
```

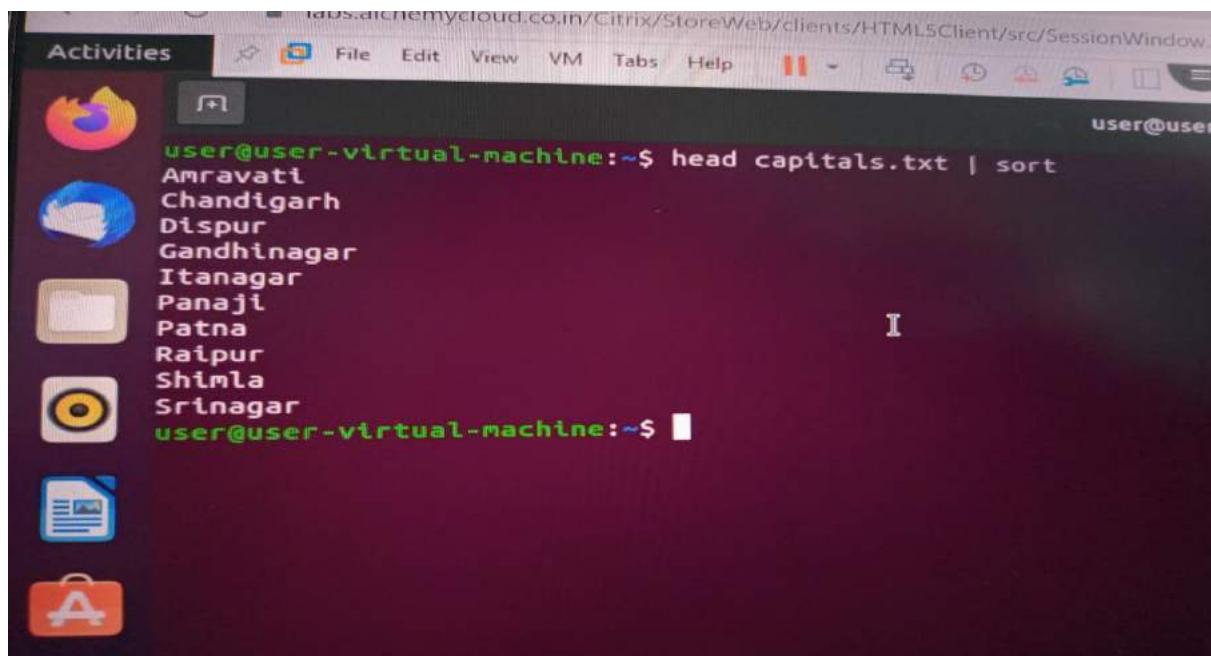
By using this option, data from the specified file is always preceded by its file name.



A screenshot of a Linux desktop environment showing a terminal window. The terminal window has a dark background and contains the following text:

```
Activities File Edit View VM Tabs Help || - + user@user-virtual-machine:~$ head -v states.txt
==> states.txt <==
Andra Pradesh
Arunachal Pradesh
Assam
Bihar
Chattishgarh
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
user@user-virtual-machine:~$
```

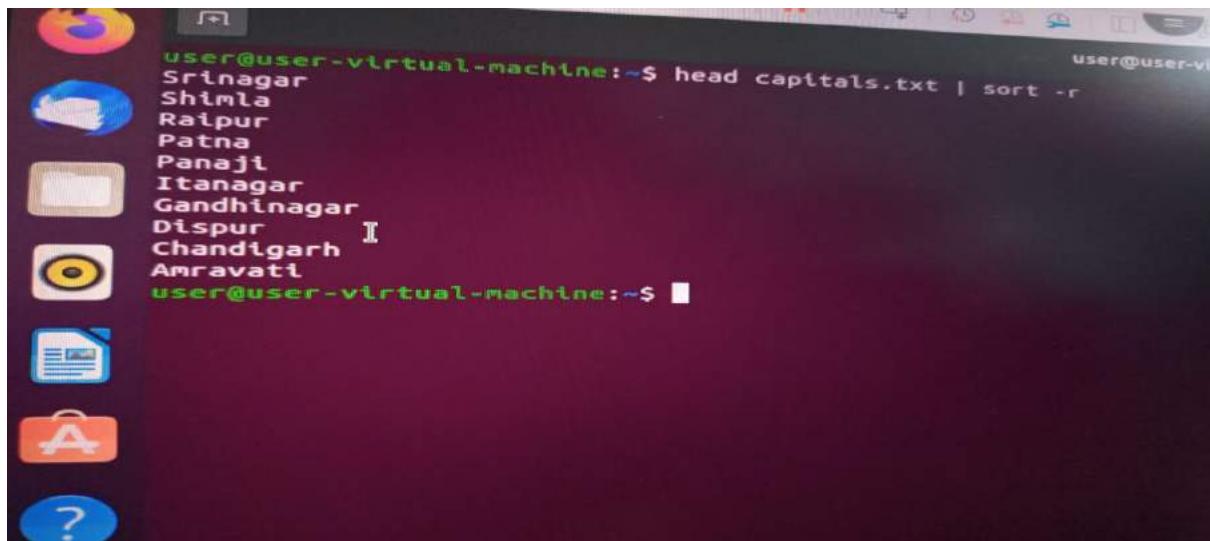
head file.txt | sort whatever the file.txt consist . It will print in sorted list.



A screenshot of a Linux desktop environment showing a terminal window. The terminal window has a dark background and contains the following text:

```
Activities File Edit View VM Tabs Help || - + user@user-virtual-machine:~$ head capitals.txt | sort
Amravati
Chandigarh
Dispur
Gandhinagar
Itanagar
Panaji
Patna
Raipur
Shimla
Srinagar
user@user-virtual-machine:~$
```

\$head file.txt | sort -r-- whatever the file.txt consist . It will print in sort in

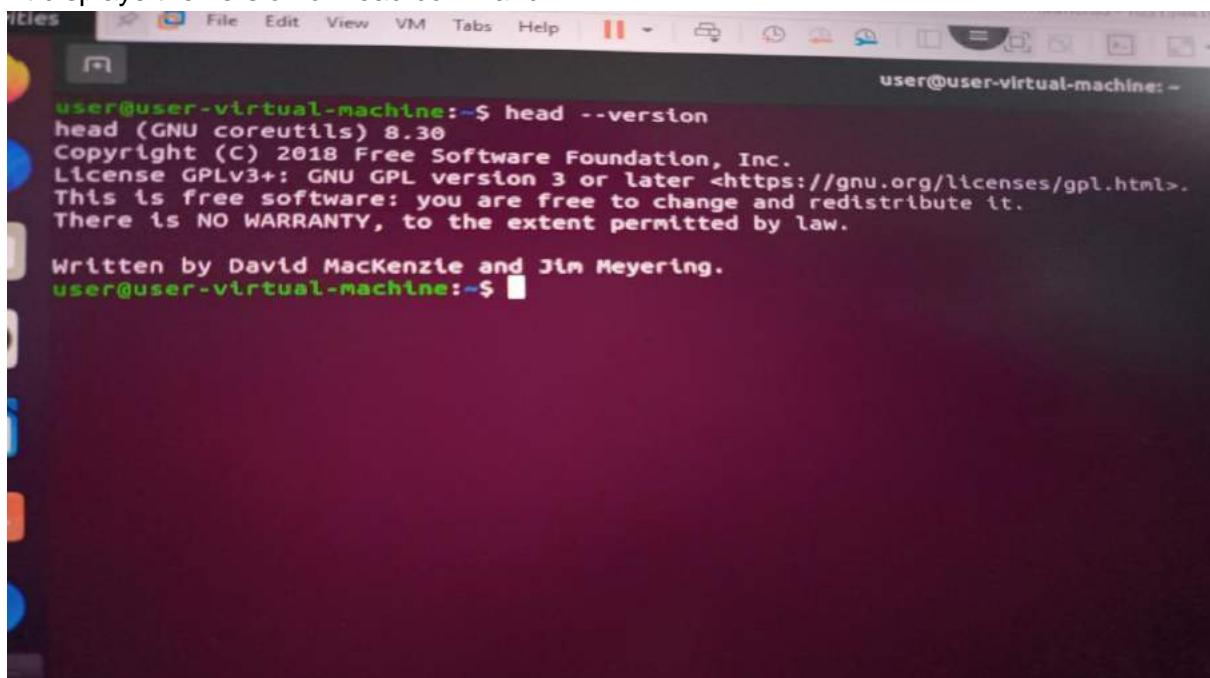


A screenshot of a Ubuntu desktop environment. On the left, there is a dock with several icons: a red folder, a blue folder, a white document, an orange folder with an 'A' icon, and a blue circle with a question mark. The main window is a terminal window titled 'user@user-virtual-machine'. The command entered is 'head capitals.txt | sort -r', and the output shows the names of Indian capital cities sorted in reverse alphabetical order:

```
user@user-virtual-machine:~$ head capitals.txt | sort -r
Srinagar
Shimla
Raipur
Patna
Panaji
Itanagar
Gandhinagar
Dispur
Chandigarh
Amravati
user@user-virtual-machine:~$
```

\$head –version

-It displays the version of head command



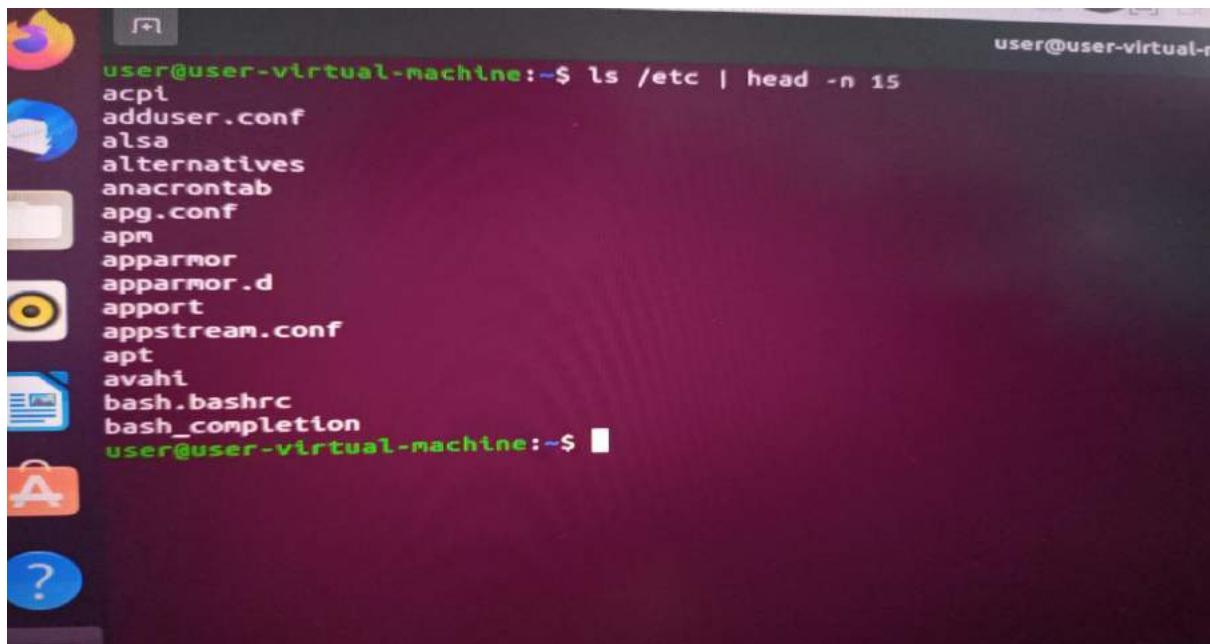
A screenshot of a Ubuntu desktop environment. On the left, there is a dock with several icons: a red folder, a blue folder, a white document, an orange folder with an 'A' icon, and a blue circle with a question mark. The main window is a terminal window titled 'user@user-virtual-machine'. The command entered is 'head --version', and the output displays the version information for the head command:

```
user@user-virtual-machine:~$ head --version
head (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 David MacKenzie and Jim Meyering.
user@user-virtual-machine:~$
```

ls /etc | head -n 15

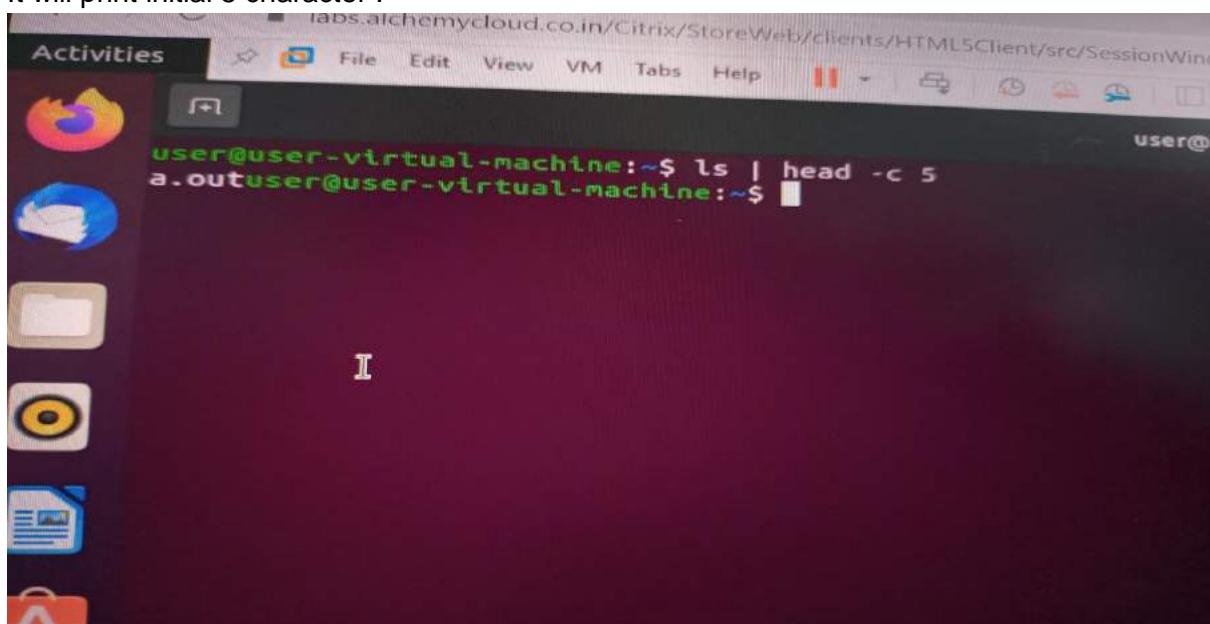
it will prints top 15 lines.



```
user@user-virtual-machine:~$ ls /etc | head -n 15
acpi
adduser.conf
alsa
alternatives
anacrontab
apg.conf
apm
apparmor
apparmor.d
apport
appstream.conf
apt
avahi
bash.bashrc
bash_completion
user@user-virtual-machine:~$
```

ls | head -c 5

It will print initial 5 character .



```
Activities File Edit View VM Tabs Help user@user-virtual-machine:~$ ls | head -c 5
a.outuser@user-virtual-machine:~$
```

ls | head -c -5

it will prints all except last 5

Using head with pipeline()

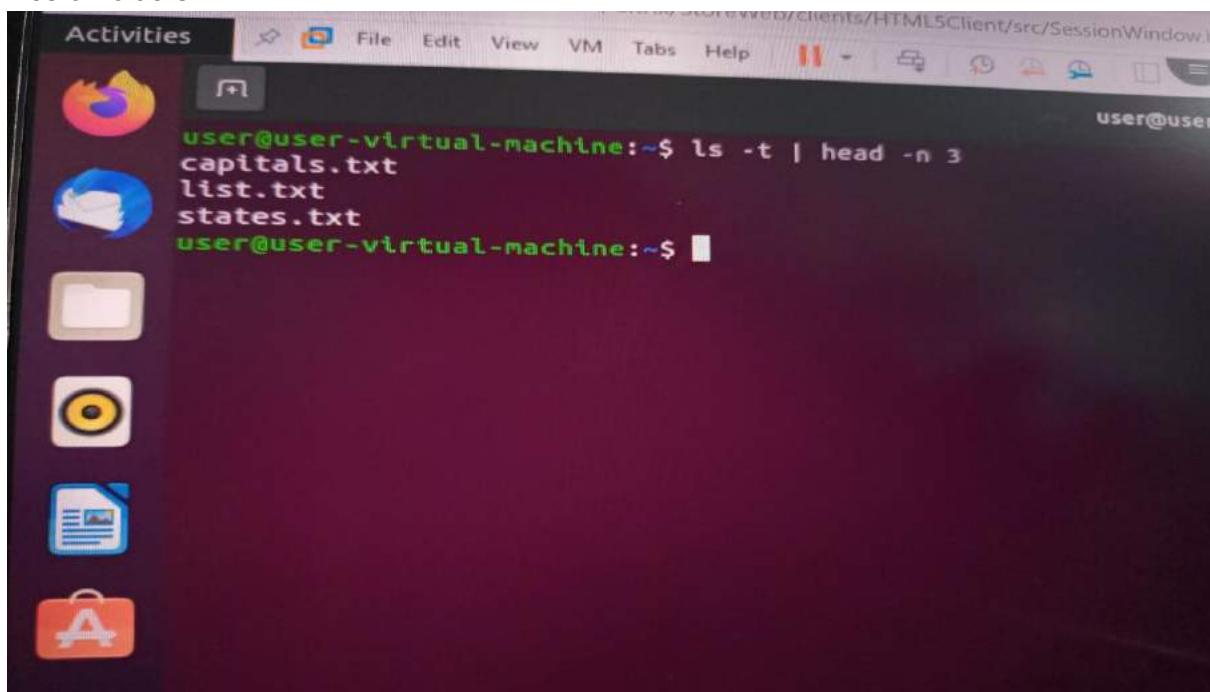
The head command can be piped with other commands.

Example

```
ls -t | head -n 3
```

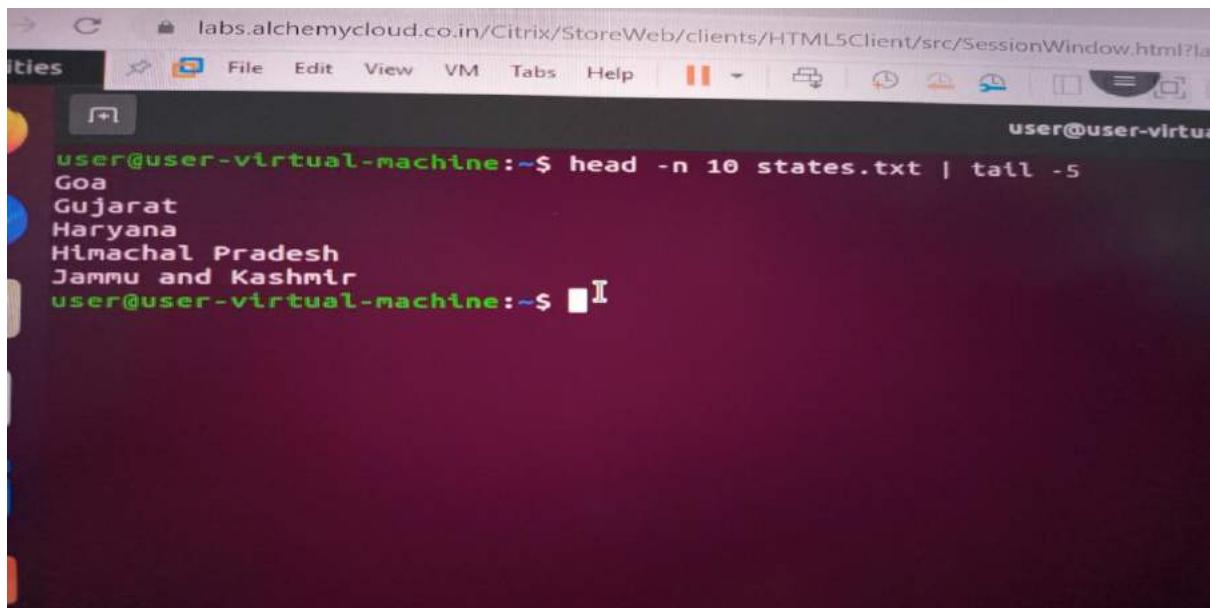
Cuts three most recently used files.

The output of the ls command is piped to head to show only the three most recently modified files or folders.



```
Activities File Edit View VM Tabs Help user@user-virtual-machine:~$ ls -t | head -n 3
capitals.txt
list.txt
states.txt
user@user-virtual-machine:~$
```

```
$head -n 10 file.txt | tail -5
```



```
Activities File Edit View VM Tabs Help user@user-virtual-machine:~$ head -n 10 states.txt | tail -5
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
user@user-virtual-machine:~$
```

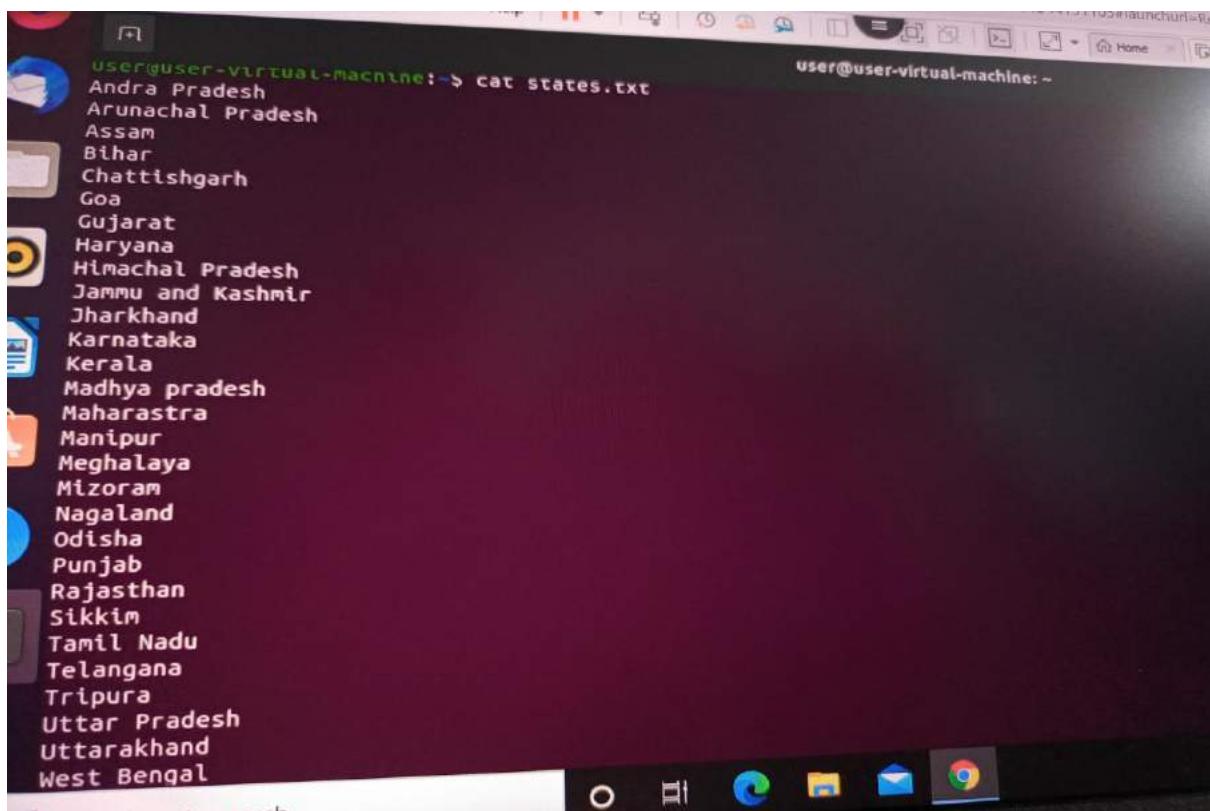
Tail command

It prints the last N number of data of the given input.

By default it prints the last 10 lines of the specified files.

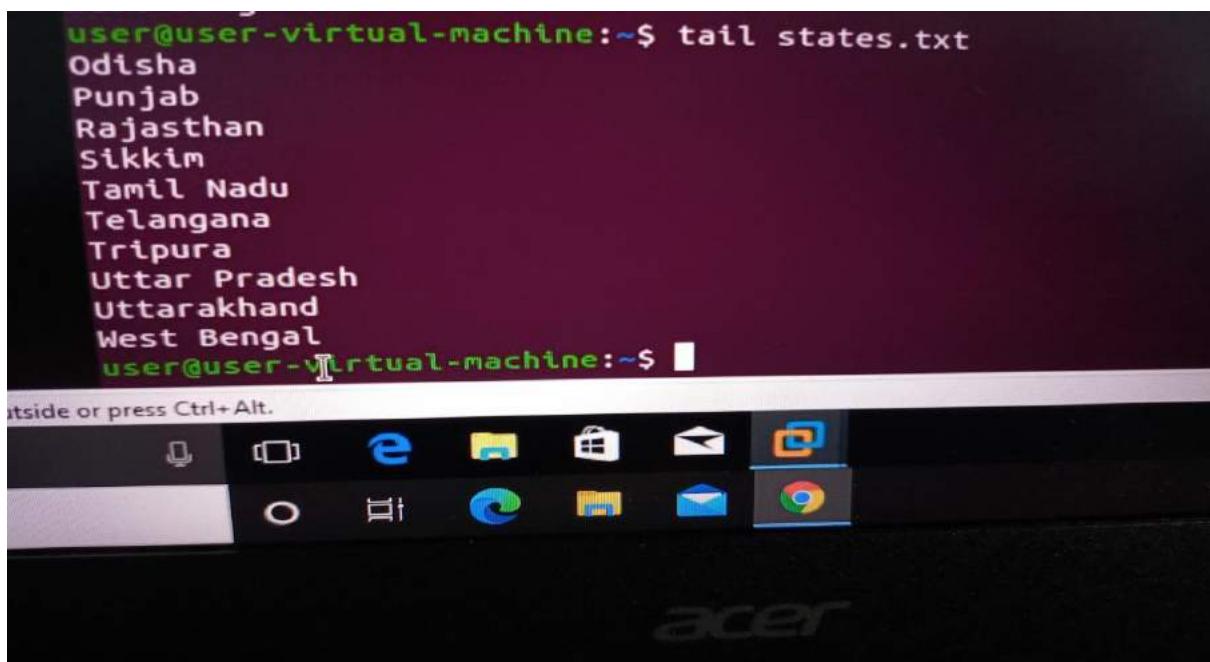
If more than one file name is provided then data from each file preceeded by its filename.

cat states.txt



```
user@user-virtual-machine:~$ cat states.txt
Andra Pradesh
Arunachal Pradesh
Assam
Bihar
Chattishgarh
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
Madhya pradesh
Maharastra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal
```

tail states.txt



```
user@user-virtual-machine:~$ tail states.txt
Odisha
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal
user@user-virtual-machine:~$
```

```
tail -n
```

It prints the last 'num' lines instead of last 10 lines.

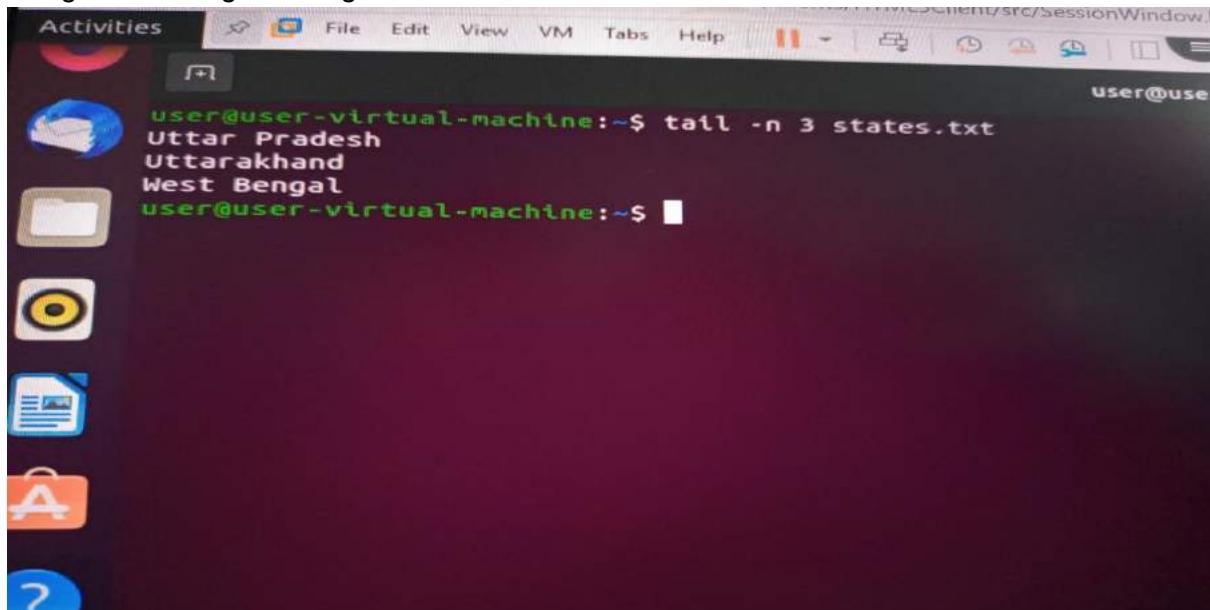
Example

```
tail -n 3 states.txt
```

or

```
tail -3 states.txt
```

This is also same as above command instead of using -n we can give “-“ sign in front of number



A screenshot of a Linux desktop environment showing a terminal window. The terminal window has a dark background and displays the following text:

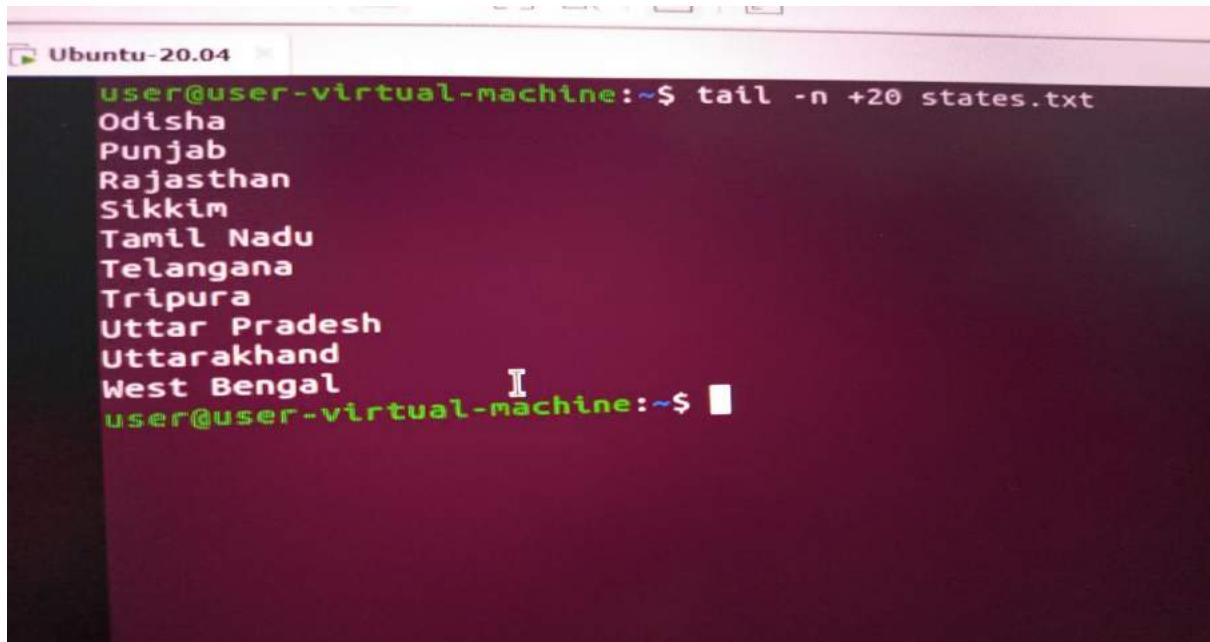
```
Activities File Edit View VM Tabs Help user@user-virtual-machine:~$ tail -n 3 states.txt
Uttar Pradesh
Uttarakhand
West Bengal
user@user-virtual-machine:~$
```

The terminal window is titled "Activities". The desktop interface includes a dock with various icons for applications like a file manager, browser, and system tools. The top bar shows the user's name and session information.

Tail command also comes with an '+' option which is not present in the head command. With this option tail command prints the data starting from specified line number of the file instead of end.

For command:

`tail -n +num file_name`, data will start printing from line number 'num' till the end of the file specified.

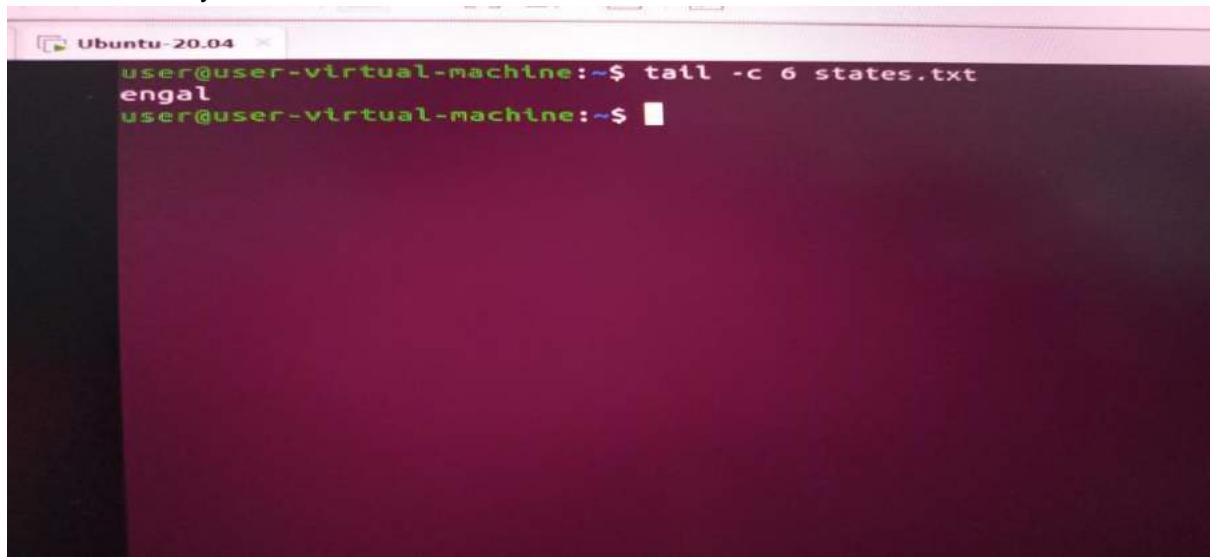


```
Ubuntu-20.04 <
user@user-virtual-machine:~$ tail -n +20 states.txt
Odisha
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal
I
user@user-virtual-machine:~$
```

`tail -c`

It prints the last 'num' bytes from the file specified.

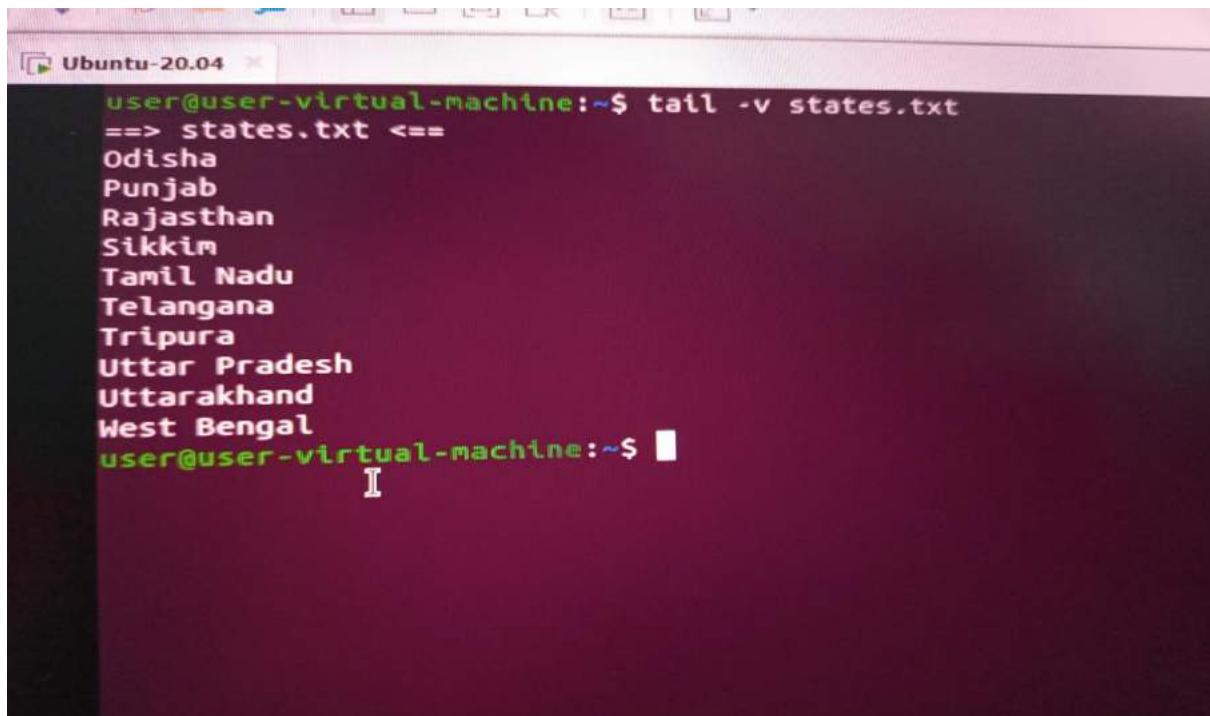
Note: New line is counted as a single character so if tail prints out a new line, it will be counted as a byte.



```
Ubuntu-20.04 <
user@user-virtual-machine:~$ tail -c 6 states.txt
engal
user@user-virtual-machine:~$
```

`tail -v`

By using this option data from the specified file is always preceded by its file name.



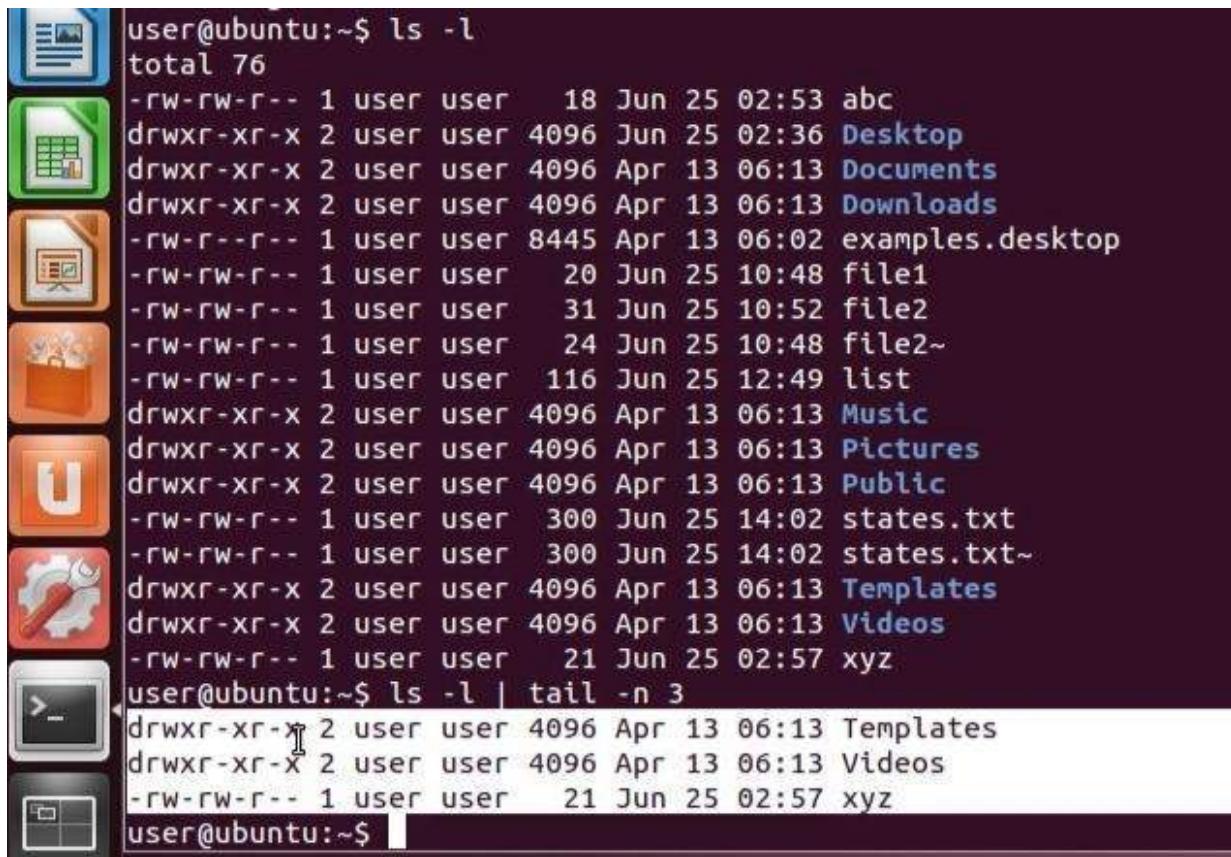
```
user@user-virtual-machine:~$ tail -v states.txt
==> states.txt <==
Odisha
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal
user@user-virtual-machine:~$ █
```

Using tail command with pipeline(|)

The tail command can be piped with other commands.

ls -l | tail -n 3

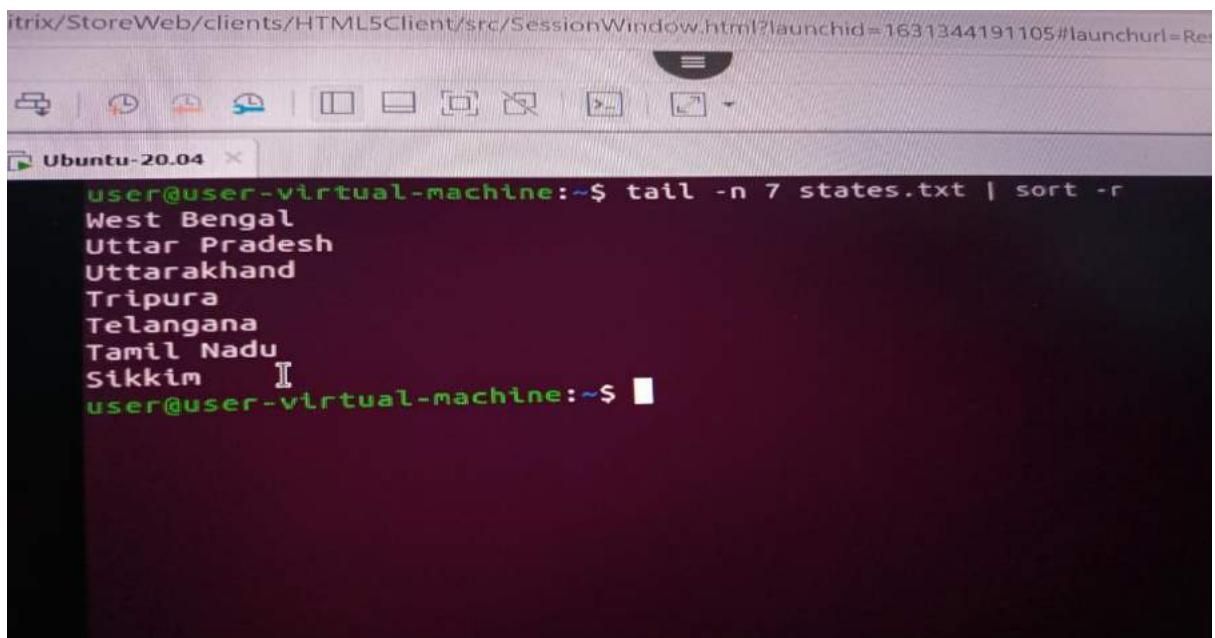
In this example, the output of the ls command is piped to the tail command which shows only the last three files.



```
user@ubuntu:~$ ls -l
total 76
-rw-rw-r-- 1 user user 18 Jun 25 02:53 abc
drwxr-xr-x 2 user user 4096 Jun 25 02:36 Desktop
drwxr-xr-x 2 user user 4096 Apr 13 06:13 Documents
drwxr-xr-x 2 user user 4096 Apr 13 06:13 Downloads
-rw-r--r-- 1 user user 8445 Apr 13 06:02 examples.desktop
-rw-rw-r-- 1 user user 20 Jun 25 10:48 file1
-rw-rw-r-- 1 user user 31 Jun 25 10:52 file2
-rw-rw-r-- 1 user user 24 Jun 25 10:48 file2~
-rw-rw-r-- 1 user user 116 Jun 25 12:49 list
drwxr-xr-x 2 user user 4096 Apr 13 06:13 Music
drwxr-xr-x 2 user user 4096 Apr 13 06:13 Pictures
drwxr-xr-x 2 user user 4096 Apr 13 06:13 Public
-rw-rw-r-- 1 user user 300 Jun 25 14:02 states.txt
-rw-rw-r-- 1 user user 300 Jun 25 14:02 states.txt~
drwxr-xr-x 2 user user 4096 Apr 13 06:13 Templates
drwxr-xr-x 2 user user 4096 Apr 13 06:13 Videos
-rw-rw-r-- 1 user user 21 Jun 25 02:57 xyz
user@ubuntu:~$ ls -l | tail -n 3
drwxr-xr-x 2 user user 4096 Apr 13 06:13 Templates
drwxr-xr-x 2 user user 4096 Apr 13 06:13 Videos
-rw-rw-r-- 1 user user 21 Jun 25 02:57 xyz
user@ubuntu:~$
```

tail -n 7 states.txt| sort -r

output of the tail command is given as input to the sort command with -r option to sort the last 7 state names coming from file states.txt in the reverse order.

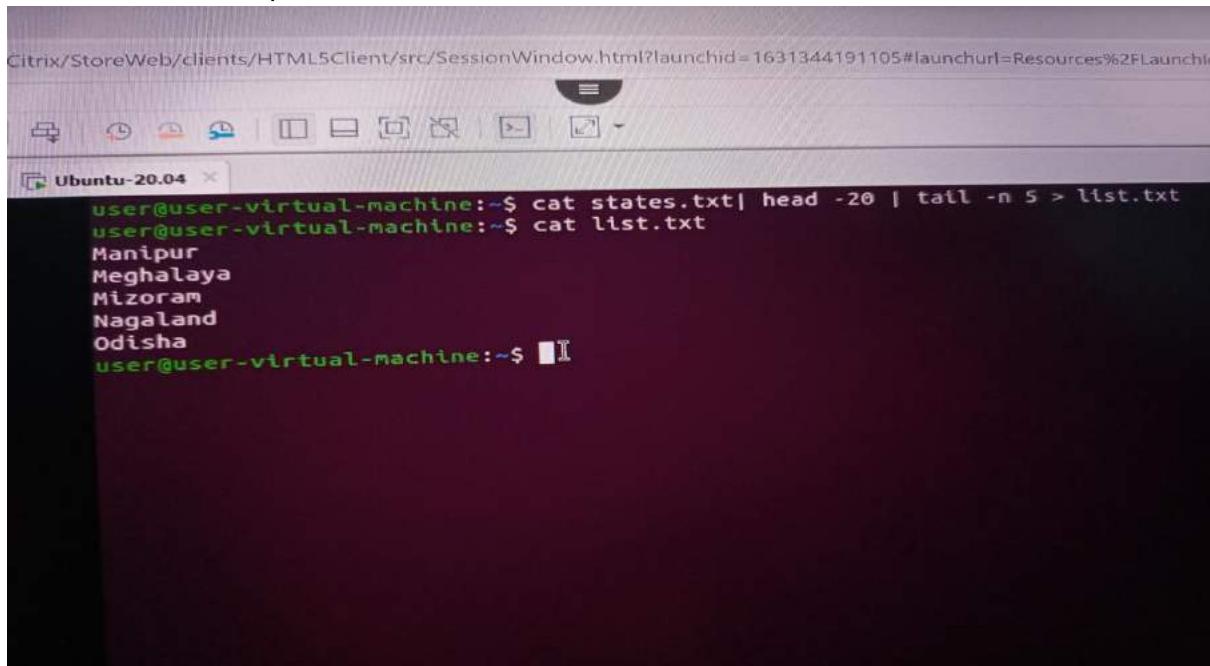


```
tail -n 7 states.txt | sort -r
West Bengal
Uttar Pradesh
Uttarakhand
Tripura
Telangana
Tamil Nadu
Sikkim
```

```
cat states.txt | head -n 20 | tail -n 5 > list.txt
```

The cat command gives

all the data present in the file states.txt and after that pipe transfers all the output coming from cat command to the head command. Head command gives all the data from start(line number 1) to the line number 20 and pipe transfer all the output coming from head command to tail command. Now, tail command gives last 5 lines of the data and the output goes to the file name list.txt via directive operator.



```
Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow.html?launchid=1631344191105#launchurl=Resources%2FLauncher%2Findex.html
```

```
Ubuntu-20.04 x
user@user-virtual-machine:~$ cat states.txt| head -20 | tail -n 5 > list.txt
user@user-virtual-machine:~$ cat list.txt
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
user@user-virtual-machine:~$ █
```

`tail -q`

It is used if more than 1 file is given.

Example

```
tail -q states.txt capitals.txt
```

This will combine the 2 specified files.

```
cat capitals.txt
```

A screenshot of an Ubuntu desktop environment. On the left, there is a dock with various icons: a document, a folder, a target, a file, a question mark, and a terminal. The main area shows a terminal window with the command `cat capitals.txt` and its output:

```
user@user-virtual-machine:~$ cat capitals.txt
Amravati
Itanagar
Dispur
Patna
Raipur
Panaji
Gandhinagar
Chandigarh
Shimla
Srinagar
Ranchi
Bengaluru
Thiruvananthapuram
Bhopal
Mumbai
Imphal
Shillong
Aizawl
Kohima
Bhuwaneswar
Chandigarh
Jaipur
Gangtok
Chennai
Hyderabad
Agartala
Lucknow
Dehradun
Kolkata
```

The terminal window has a dark background with light-colored text. The title bar says "user@user-virtual-machine: ~". Below the terminal is a search bar with the placeholder "Type here to search" and a dock with icons for Home, Dash, Activities, Applications, and the Dash search bar.

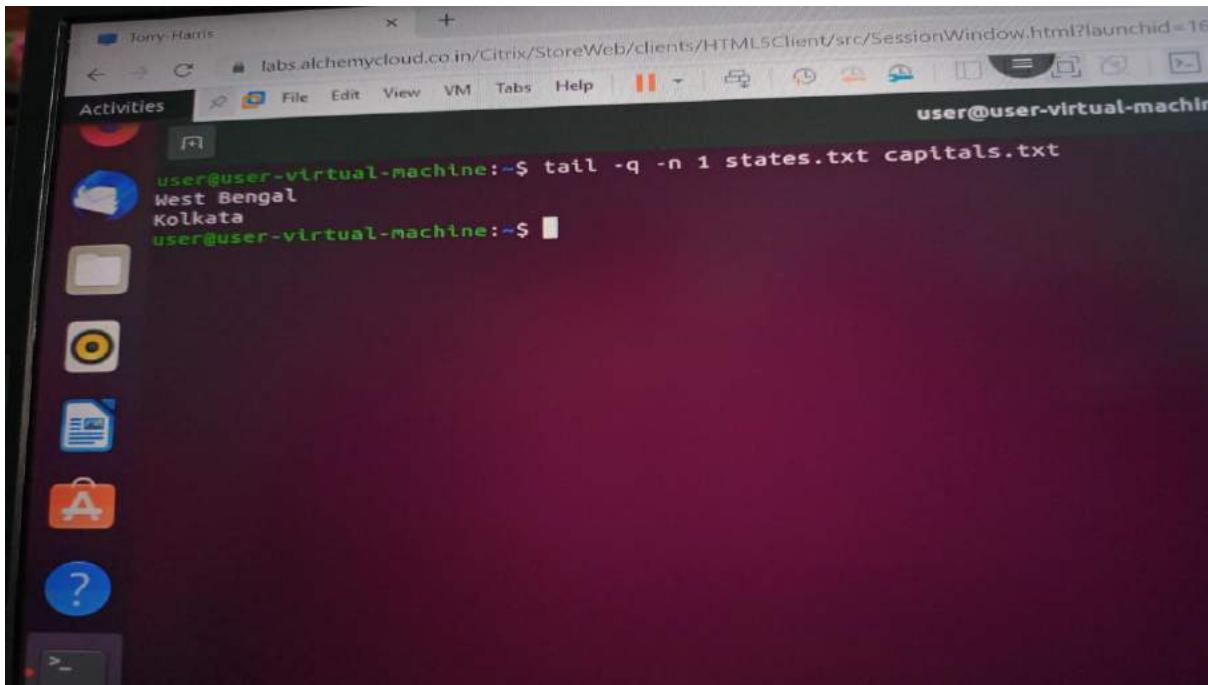
A screenshot of an Ubuntu desktop environment. On the left, there is a dock with various icons: a document, a folder, a target, a file, a question mark, and a terminal. The main area shows a terminal window with the command `tail -q states.txt capitals.txt` and its output:

```
user@user-virtual-machine:~$ tail -q states.txt capitals.txt
Odisha
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal
Bhuwaneswar
Chandigarh
Jaipur
Gangtok
Chennai
Hyderabad
Agartala
Lucknow
Dehradun
Kolkata
```

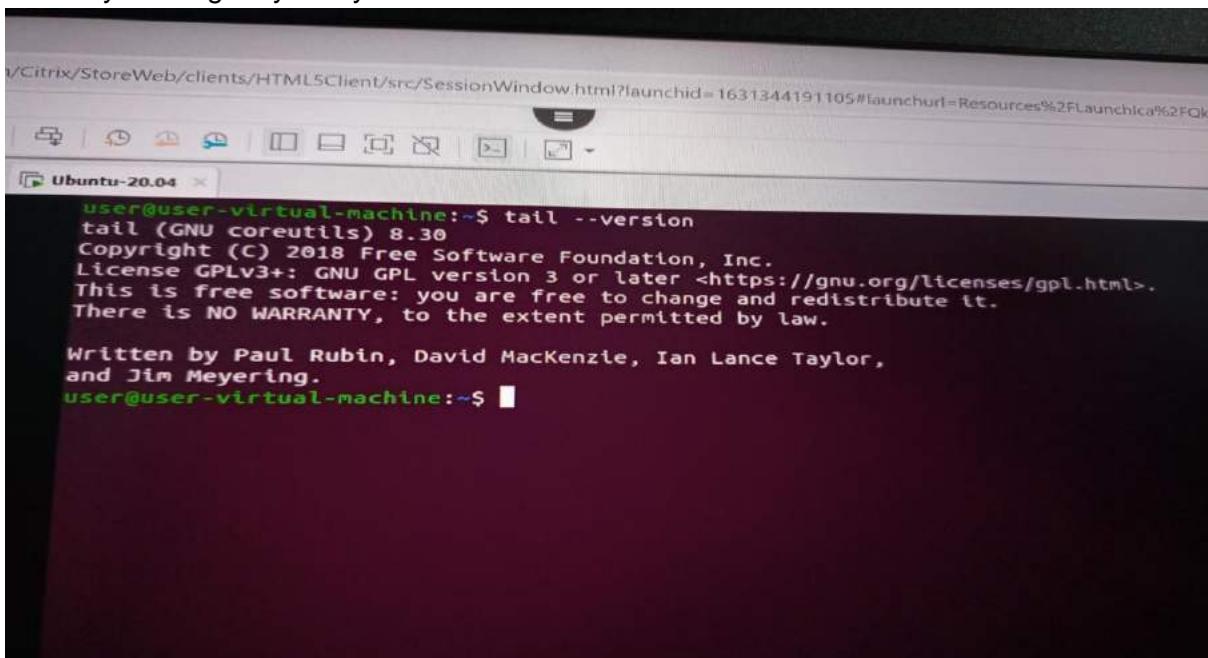
The terminal window has a dark background with light-colored text. The title bar says "user@user-virtual-machine: ~". Below the terminal is a dock with icons for Home, Dash, Activities, Applications, and the Dash search bar.

(odisha to west bengal) are states and (Bhuwaneswar to Kolkata) are capitals respectively.

```
tail -q -n 1 states.txt capitals.txt
```



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



Using gedit

```
user@user-virtual-machine:~$ cat > hh.txt
hyderabad
delhi
mumbai
ahmedabad
user@user-virtual-machine:~$ sort hh.txt
ahmedabad
delhi
hyderabad
mumbai
user@user-virtual-machine:~$ gedit hh.txt
user@user-virtual-machine:~$ gedit hhh.sh
user@user-virtual-machine:~$
```

sort filename

```
Activities Text Editor Sep 8 22:57 *hhh.sh
Open File
1 #!/bin/bash
2 cat hh.txt
3 sort hh.txt > oo.txt
4
```

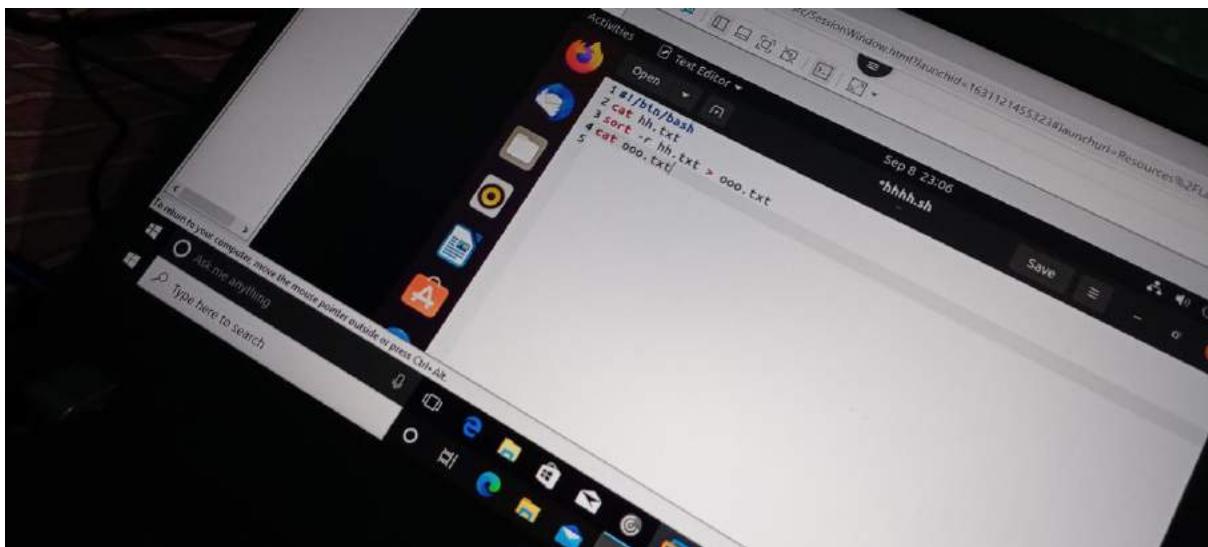
```
user@user-virtual-machine:~$ cat > hh.txt
hyderabad
delhi
mumbai
ahmedabad
user@user-virtual-machine:~$ sort hh.txt
ahmedabad
delhi
hyderabad
mumbai
user@user-virtual-machine:~$ gedit hh.txt
user@user-virtual-machine:~$ gedit hhh.sh
user@user-virtual-machine:~$ ./hhh.sh
bash: ./hhh.sh: Permission denied
user@user-virtual-machine:~$ chmod +x hhh.sh
user@user-virtual-machine:~$ ./hhh.sh
hyderabad
delhi
mumbai
ahmedabad
user@user-virtual-machine:~$
```

A screenshot of a Linux desktop environment. A terminal window is open in the top right corner, displaying a command-line session:

```
user@user-virtual-machine:~$ sort hh.txt
ahendabad
delhi
hyderabad
mumbai
user@user-virtual-machine:~$ gedit hh.txt
user@user-virtual-machine:~$ gedit hhh.sh
bash: ./hhh.sh: Permission denied
user@user-virtual-machine:~$ chmod +x hhh.sh
user@user-virtual-machine:~$ ./hhh.sh
ahendabad
delhi
hyderabad
mumbai
user@user-virtual-machine:~$ cat oo.txt
ahendabad
delhi
hyderabad
mumbai
user@user-virtual-machine:~$
```

The desktop interface includes a dock with various icons (File Explorer, Mail, etc.) and a system tray.

sort -r filename



or

Activities

Text Editor

Sep 8 23:14

Open

*hhhh.sh

```
1 #!/bin/bash
2 cat hh.txt | sort -r hh.txt > ooo.txt
3 cat ooo.txt
4
```

The image shows a screenshot of an Ubuntu desktop environment. On the left, there is a vertical dock with several icons: a flame (Dash), a document with a blue circle (Files), a folder (Files), a target (System Settings), a document with a blue border (Dash), and a shopping bag with a white letter 'A' (Applications). The main window is a terminal titled 'Text Editor' with the file name '*hhhh.sh'. The terminal contains the following text:1 #!/bin/bash
2 cat hh.txt | sort -r hh.txt > ooo.txt
3 cat ooo.txt
4

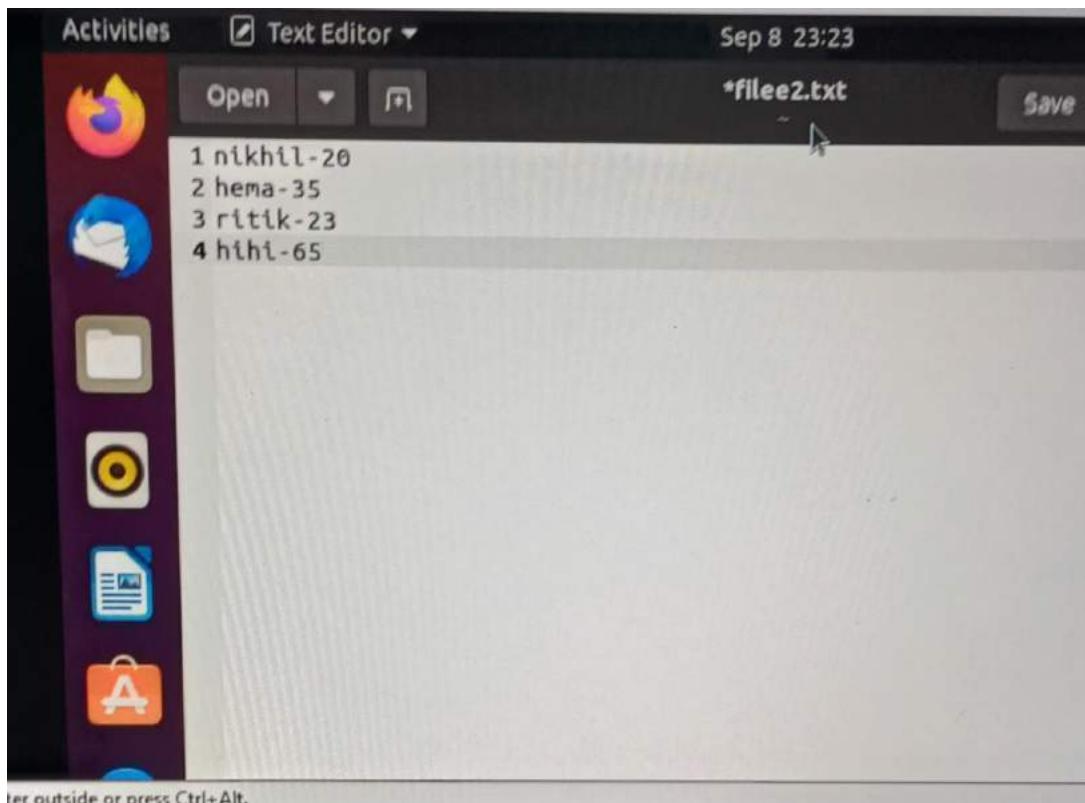
At the bottom of the terminal window, there is a message: "de or press Ctrl+Alt."

04

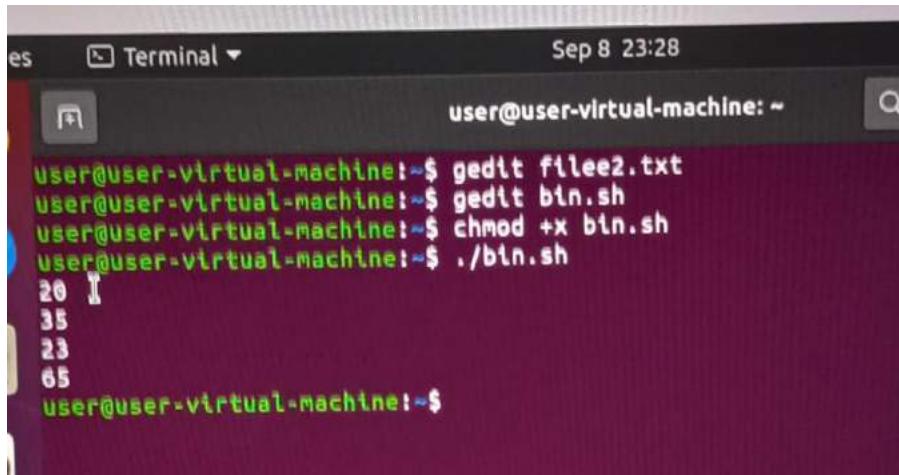
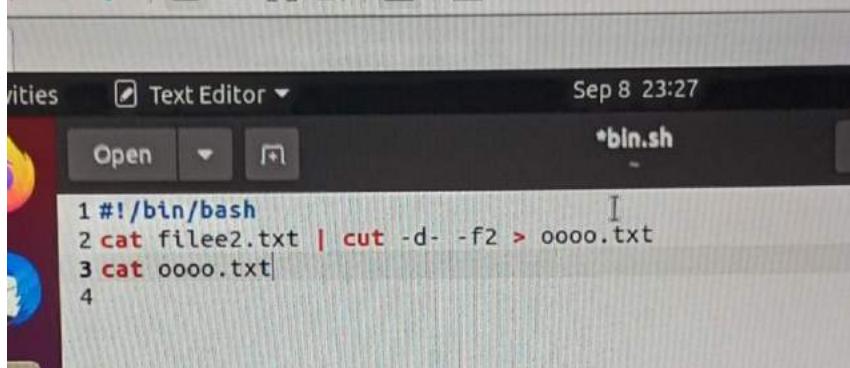
```
delhi
hyderabad
mumbai
user@user-virtual-machine:~$ cat hh.txt
hyderabad
delhi
mumbai
ahemdabad
user@user-virtual-machine:~$ gedit hhhh.sh
user@user-virtual-machine:~$ chmod +x hhhh.sh
user@user-virtual-machine:~$ ./hhhh.sh
hyderabad
delhi
mumbai
ahemdabad
mumbai
hyderabad
delhi
ahemdabad
user@user-virtual-machine:~$ cat ooo.txt
mumbai
hyderabad
delhi
ahemdabad
user@user-virtual-machine:~$
```

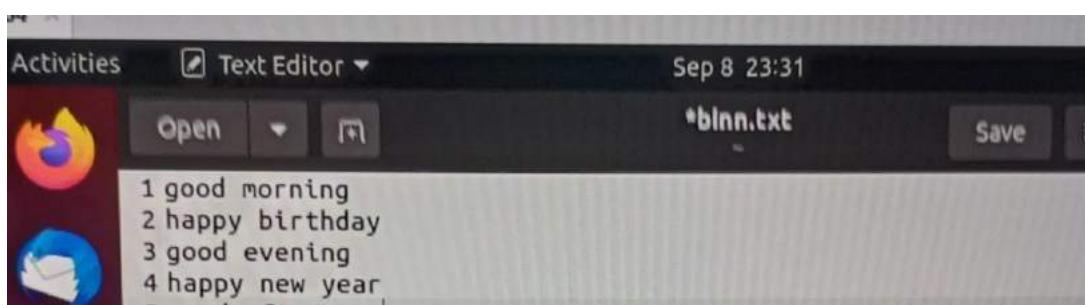
es Terminal ▾ Sep 8 23:24

```
user@user-virtual-machine:~$ gedit filee2.txt
user@user-virtual-machine:~$
```



Move the cursor outside or press Ctrl+Alt.

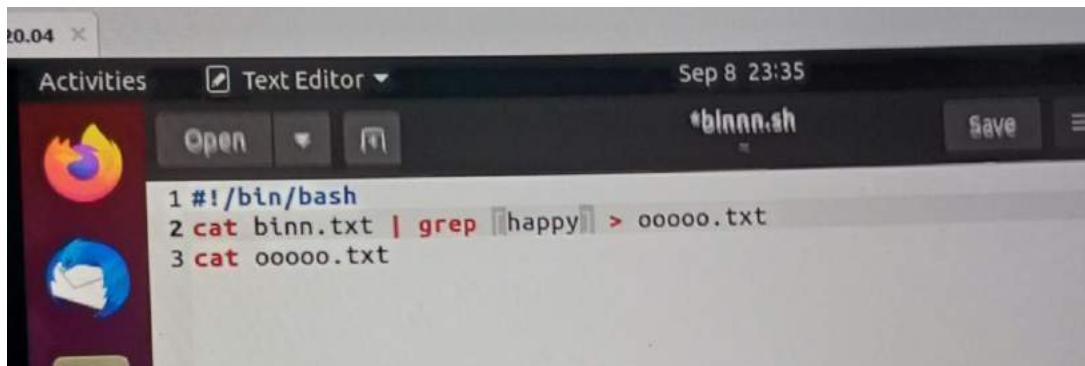




Activities Text Editor Sep 8 23:31

*binn.txt Save

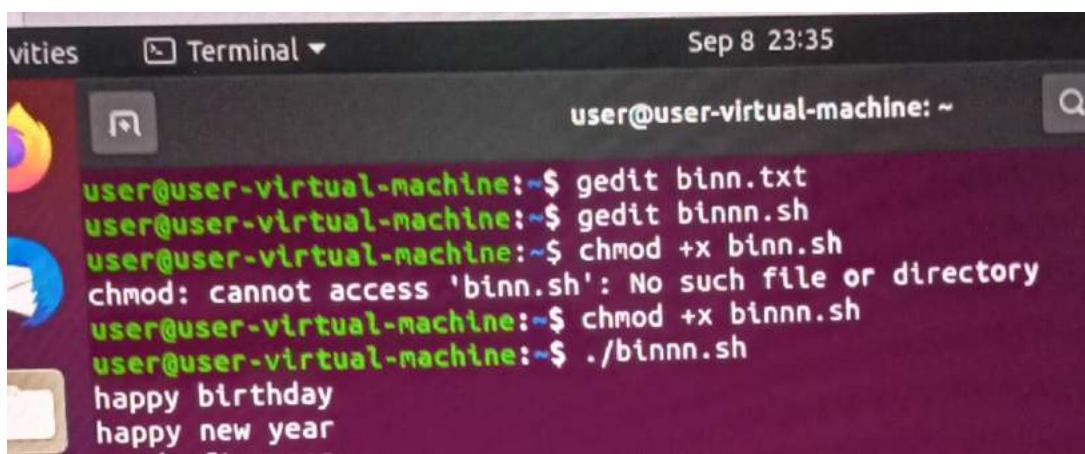
```
1 good morning
2 happy birthday
3 good evening
4 happy new year
```



Activities Text Editor Sep 8 23:35

*binnn.sh Save

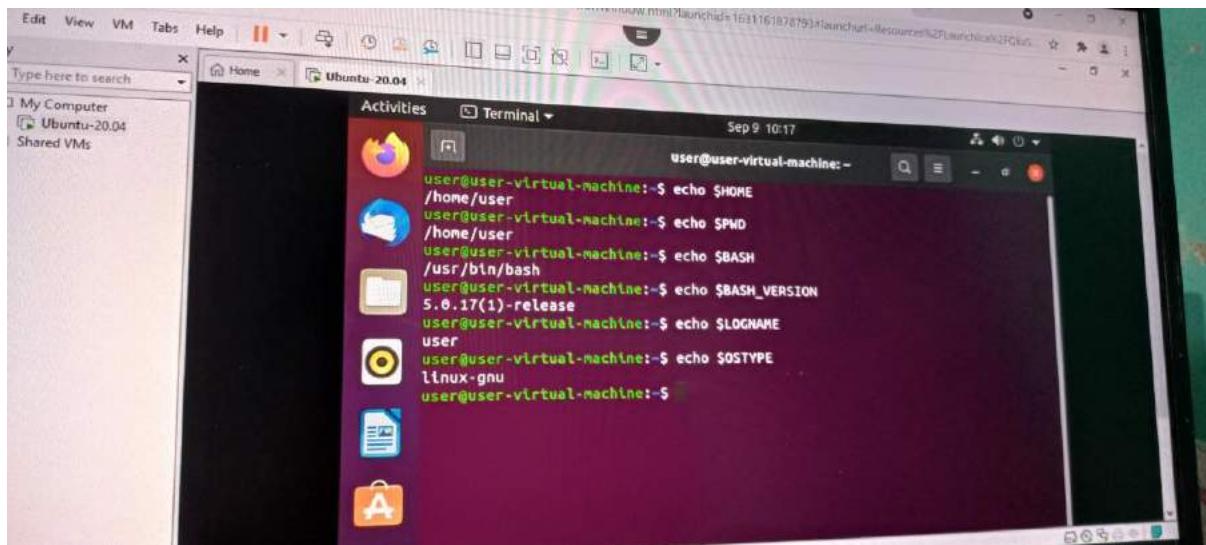
```
1 #!/bin/bash
2 cat binn.txt | grep happy > ooooo.txt
3 cat ooooo.txt
```



Activities Terminal Sep 8 23:35

user@user-virtual-machine: ~

```
user@user-virtual-machine:~$ gedit binn.txt
user@user-virtual-machine:~$ gedit binnn.sh
user@user-virtual-machine:~$ chmod +x binnn.sh
chmod: cannot access 'binnn.sh': No such file or directory
user@user-virtual-machine:~$ chmod +x binnn.sh
user@user-virtual-machine:~$ ./binnn.sh
happy birthday
happy new year
```

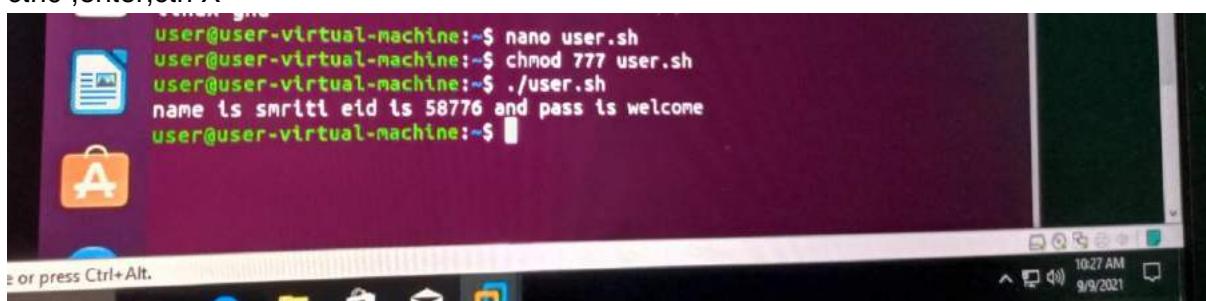


input student info and print it using echo



come out of nano by using

ctrlO ,enter,ctrl X



A screenshot of a terminal window titled "s1.sh". The window has a dark theme with a light gray background. The title bar shows the URL: "labs.alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow.html?launc". The menu bar includes "File", "Edit", "View", "VM", "Tabs", and "Help". Below the menu is a toolbar with various icons. The main area contains a bash script:

```
1 #!/bin/bash
2 echo "enter first name"
3 read fname;
4 echo "enter second name"
5 read sname
6 echo $fname $sname
7 read -p "enter the address" adds
8 echo $adds
9
10 d=`date +%m-%d-%Y`
11 echo $d #MM-DD-YYYY
```

```
Terry-Harris Bash Sleep - Javatpoint
File Edit View VM Tabs Help || + user@user-virtual-machine: ~
+ user@user-virtual-machine:~$ gedit s1.sh
user@user-virtual-machine:~$ ./s1.sh
enter first name
smriti
enter second name
bhagat
smriti bhagat
enter the addressjharkhand
jharkhand
09-12-2021
user@user-virtual-machine:~$
```

orri-Harris Bash Sleep - Javatpoint

File Edit View VM Tabs Help

Open yay.sh

```
1 #!/bin/bash
2 echo "file name: $0"
3 echo "first argument: $1"
4 echo "second argument: $2"
5 echo "argument values: $*"
6 echo "total no. of arguments: $#"
7 for token in $*
8 do
9 echo $token
10 done|
```

The screenshot shows a terminal window with a dark background and light-colored text. At the top, there's a header bar with icons for file operations and a menu bar with "File", "Edit", "View", "VM", "Tabs", and "Help". Below the header, there's a toolbar with a search icon, a refresh icon, and other small icons. The main area of the terminal displays the following command-line session:

```
user@user-virtual-machine:~$ gedit yay.sh
user@user-virtual-machine:~$ ./yay.sh smriti bhagat
file name: ./yay.sh
first argument: smriti
second argument: bhagat
argument values: smriti bhagat
total no. of arguments: 2
smriti
bhagat
user@user-virtual-machine:~$
```

My-Harris ✎ Bash Sleep - Javatpoint

File Edit View VM Tabs Help

Open s2.sh

```
1 #!/bin/bash
2 echo "processes started"
3 echo "process is going to sleep for 1 sec"
4 sleep 1s
5 echo "process wakeup"
6 echo "process completed"
```

A screenshot of a terminal window titled "Bash Sleep - Javatpoint". The terminal shows the following command-line interaction:

```
user@user-virtual-machine:~$ gedit s2.sh
user@user-virtual-machine:~$ ./s2.sh
processes started
process is going to sleep for 1 sec
process wakeup
process completed
user@user-virtual-machine:~$
```

largest of 3 numbers

A screenshot of a web-based terminal interface from <https://labs.alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow.html?laur>. The terminal window has a dark theme with a light gray background. The title bar shows the URL. The menu bar includes File, Edit, View, VM, Tabs, Help, and several icons. Below the menu is an 'Open' dropdown and a '+' button. The main area contains a bash script named 'u.sh' with line numbers 1 through 14. The script reads three integers from the user, compares them to find the largest, and then prints "hello \n hi".

```
1 #!/bin/bash
2 read -p "enter n1:" n1
3 read -p "enter n2:" n2
4 read -p "enter n3:" n3
5 if [ $n1 -gt $n2 -a $n1 -gt $n3 ]
6 then
7 echo '$n1 is largest'
8 elif [ $n2 -gt $n1 -a $n2 -gt $n3 ]
9 then
10 echo '$n2 is largest'
11 else
12 echo '$n3 is largest'
13 fi
14 echo -e " hello \n hi "
```

A screenshot of a Linux desktop environment showing a terminal window. The terminal window has a dark background and displays the following text:

```
user@user-virtual-machine:~$ gedit u.sh
user@user-virtual-machine:~$ ./u.sh
enter n1:90
enter n2:109
enter n3:60
109 is largest
hello
hi
user@user-virtual-machine:~$
```

Check number are equal or not.

v-Harris

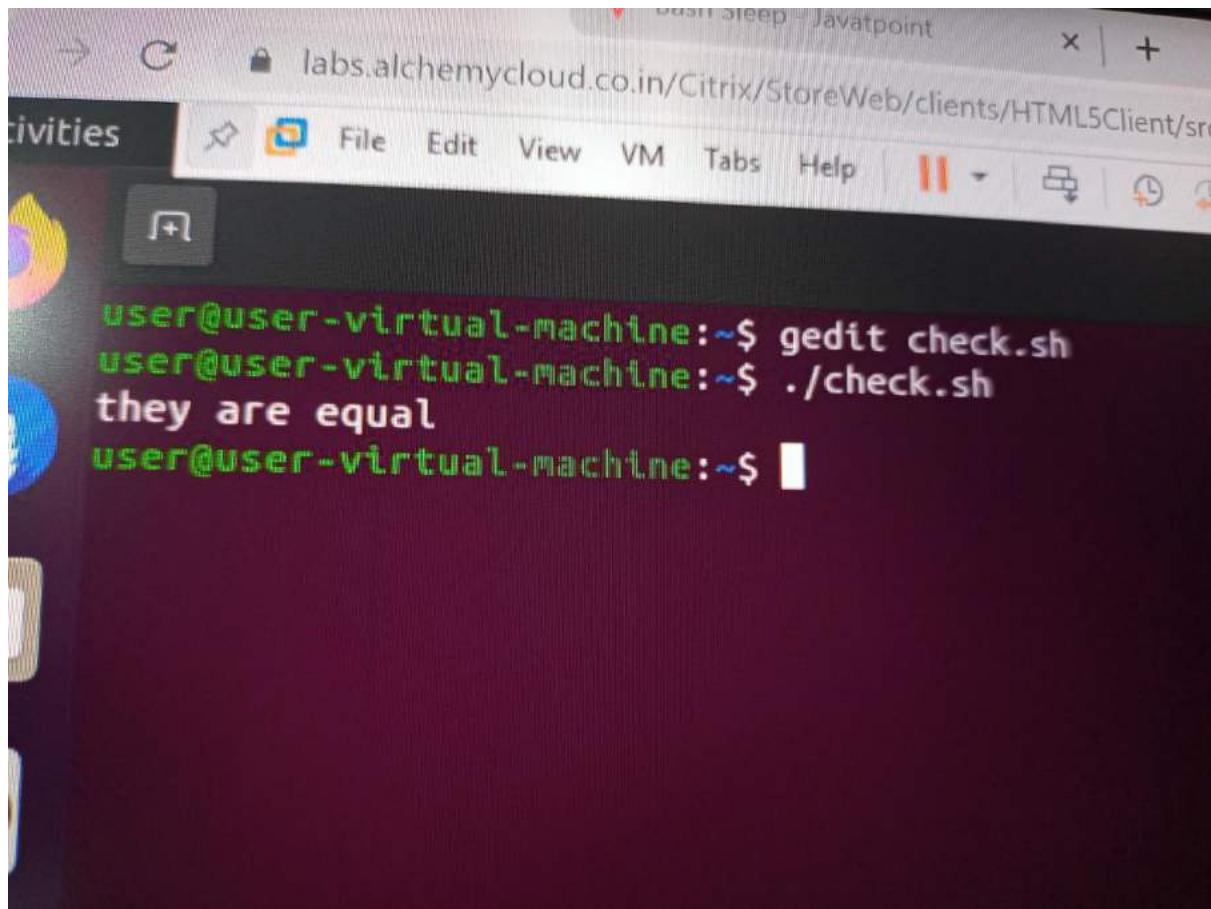
Bash Sleep - Javatpoint

labs.alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow.html?launchid=5

File Edit View VM Tabs Help

Open *check.sh

```
1 #!/bin/bash
2 str1="abc"
3 str2="abc"
4 if [ $str1 == $str2 ];
5 then
6 echo "they are equal"
7 else
8 echo "not equal"
9 fi
0
1
```



The screenshot shows a Linux terminal window titled "dash sleep - Javatpoint" with the URL "labs.alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/sr...". The terminal window has a dark background and displays the following text:

```
user@user-virtual-machine:~$ gedit check.sh
user@user-virtual-machine:~$ ./check.sh
they are equal
user@user-virtual-machine:~$
```

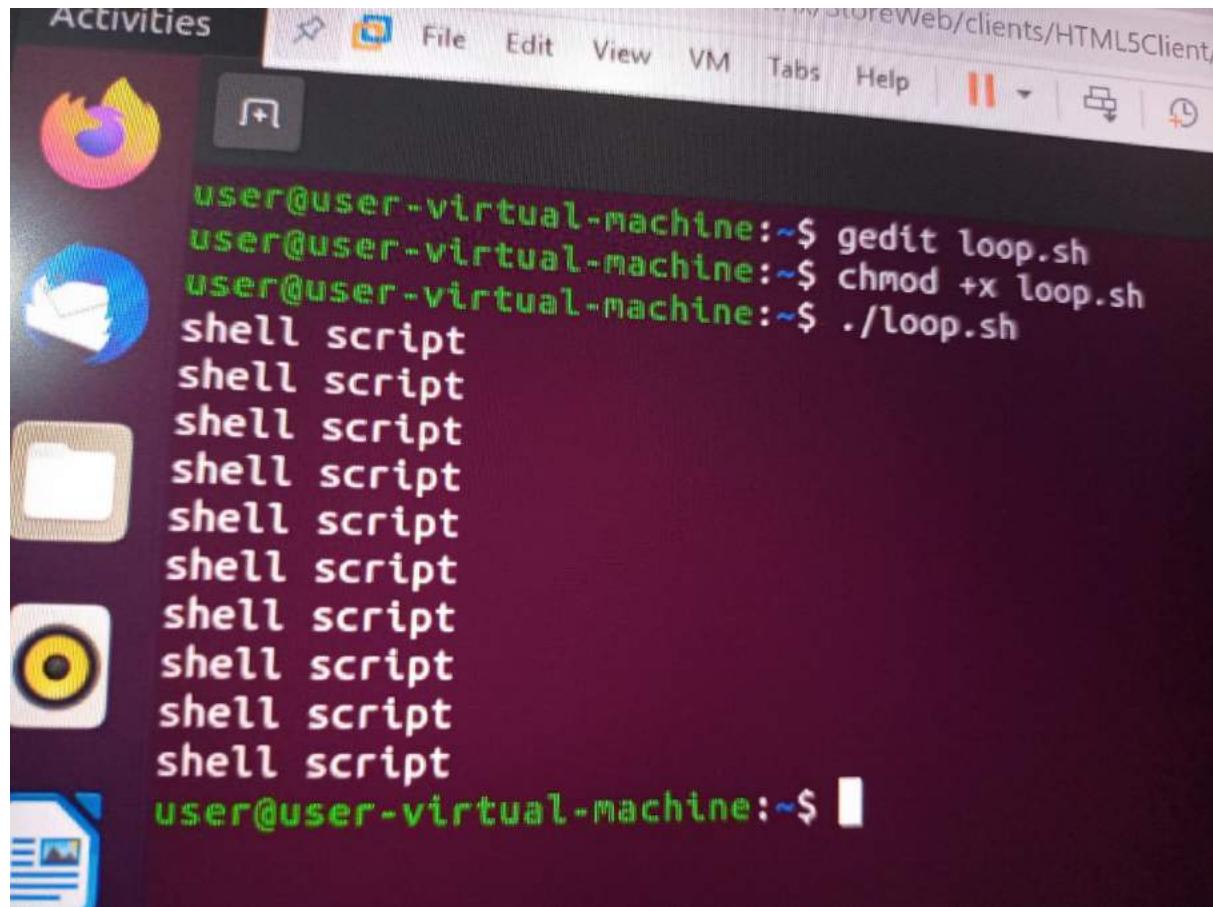
for loop

Bash Sleep - Javatpoint

File Edit View VM Tabs Help || +

Open *loop.sh

```
1#!/bin/bash
2for i in {1..10}
3do
4echo "shell script"
5done|
```



A screenshot of a Citrix session window titled "Bash Sleep - Javatpoint". The window shows a terminal interface with a shell script named "loop.sh" open. The script content is as follows:

```
1 #!/bin/bash
2 for i in {1..10}
3 do
4 echo "$i shell script"
5 done
```

A screenshot of a Linux desktop environment. On the left, there's an 'Activities' dock with icons for a browser, file manager, terminal, and other applications. The main window is a terminal emulator titled 'File' with the URL 'labs.alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/S'. The terminal window shows the following session:

```
user@user-virtual-machine:~$ gedit loop.sh
user@user-virtual-machine:~$ ./loop.sh
1 shell script
2 shell script
3 shell script
4 shell script
5 shell script
6 shell script
7 shell script
8 shell script
9 shell script
10 shell script
user@user-virtual-machine:~$ █
```

Torry-Harris Bash Sleep - Javaatpoint

→ C labs.alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow.html?laun

vities File Edit View VM Tabs Help || +

Open *loop.sh

```
1 #!/bin/bash
2 for i in [1..10..2]
3 do
4 echo "Si shell script"
5 done
```

A screenshot of a Linux desktop environment showing a terminal window. The terminal window has a dark background and displays the following text:

```
user@user-virtual-machine:~$ gedit loop.sh
user@user-virtual-machine:~$ ./loop.sh
1 shell script
3 shell script
5 shell script
7 shell script
9 shell script
user@user-virtual-machine:~$ █
```

The terminal window is part of a desktop interface with a menu bar at the top. The URL in the browser bar is `https://alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5`.

ry-Harris

Bash Sleep - Javatpoint

labs.alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow.html?laun

S File Edit View VM Tabs Help || +

Open +

*loop.sh

```
1 #!/bin/bash
2 str="All you touch is gold"
3 for s in $str
4 do
5 echo $s
6 done
7
8
9
10
11
12
13
14
15
16
17
```

A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window. The terminal window has a dark background and displays the following text:

```
user@user-virtual-machine:~$ gedit loop.sh
user@user-virtual-machine:~$ ./loop.sh
All
you
touch
is
gold
user@user-virtual-machine:~$
```

rry-Harris

Bash Sleep - Javatpoint

File Edit View VM Tabs Help

*loop.sh

```
1 #!/bin/bash
2 str="All you touch is gold"
3 for s in $str
4 do
5 echo $s
6 done
7 for i in {1..10..1}
8 do
9 echo "$i * 5 = $((i*5))"
10 done|
```

A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window. The terminal window has a dark background and displays the following text:

```
User@user-virtual-machine:~$ gedit loop.sh
User@user-virtual-machine:~$ ./loop.sh
All
you
touch
is
gold
1 * 5 = 5
2 * 5 = 10
3 * 5 = 15
4 * 5 = 20
5 * 5 = 25
6 * 5 = 30
7 * 5 = 35
8 * 5 = 40
9 * 5 = 45
10 * 5 = 50
user@user-virtual-machine:~$
```

factorial(for)

A screenshot of a web browser window titled "Java Sleep - JavaPoint". The address bar shows the URL: "labs.alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow.html?launchid=...". The browser interface includes a menu bar with File, Edit, View, VM, Tabs, Help, and a toolbar with various icons. A search bar is present above the main content area. The main content area displays a Bash script named "factorial.sh". The script is as follows:

```
1 #!/bin/bash
2 read -p "enter a number" num
3 fact=1
4 for((i=2;i<=num;i++))
5 {
6 fact=$((fact*i))
7 }
8 echo $fact
```

A screenshot of a terminal window titled "Bash Sleep - Javatpoint". The URL in the address bar is "labs.alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/Session". The terminal content is as follows:

```
user@user-virtual-machine:~$ gedit factorial.sh
user@user-virtual-machine:~$ ./factorial.sh
enter a number5
120
user@user-virtual-machine:~$
```

prime(for)

A screenshot of a terminal window titled "Bash Sleep - Javatpoint". The window shows a Bash script named "prime.sh" with the following code:

```
1 echo -e "enter the number: \n"
2 read n
3 for (( i=2; i<=$n/2; i++ ))
4 do
5 a=$(( n%i ))
6 if [ $a -eq 0 ]
7 then
8 echo "$n is not a prime"
9 exit 0
10 fi
11 done
12 echo "$n is a prime"
```

The screenshot shows a Linux desktop interface with a terminal window open. The terminal window title is "Bash Sleep - Javatpoint". The terminal content is as follows:

```
user@user-virtual-machine:~$ gedit prime.sh
user@user-virtual-machine:~$ ./prime.sh
enter the number:
13
13 is a prime
user@user-virtual-machine:~$ ./prime.sh
enter the number:
77
77 is not a prime
user@user-virtual-machine:~$
```

while

It is used when we do not know exactly the number of times loops to be executed, we know the condition on which it has to be looped.

The screenshot shows a Linux desktop interface with a terminal window open. The terminal window has a title bar "Bash Sleep - Javatpoint" and a URL "labs.alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow". The window contains the following Bash script:

```
1 #!/bin/bash
2 i=1
3 while [ $i -le 10 ]
4 do
5 echo $i
6 ((i++))
7 done
8
9
10
11
12
```

Terry-Harris

Bash Sleep - Javatpoint

Activities

File Edit View VM Tabs Help

user@user-virtual-machine:~\$ gedit until.sh

user@user-virtual-machine:~\$./until.sh

1
2
3
4
5
6
7
8
9
10 user@user-virtual-machine:~\$

This screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is 'Bash Sleep - Javatpoint' and its URL is 'labs.alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/SessionWi...'. The terminal content shows a user running a script named 'until.sh'. The script contains ten lines of code, numbered 1 through 10. Line 10 is the final command executed, which is the script itself. The terminal prompt is 'user@user-virtual-machine:~\$'.

A screenshot of a web-based terminal session titled "Bash Sleep - Javatpoint". The terminal window has a dark theme and displays a Bash script. The script starts with a shebang line, initializes a variable 'a' to 5, and then enters a 'while' loop. Inside the loop, it prints the value of 'a' using 'echo', increments 'a' by 1 using 'expr \$a+1', and then exits the loop with 'done'. The script ends with a final line number '13'.

```
1 #!/bin/bash
2 a=5
3 while [ $a -lt 5 ]
4 do
5 echo $a
6 a=`expr $a+1`
7 done
8
9
10 |
11
12
13
```

factorial(while)

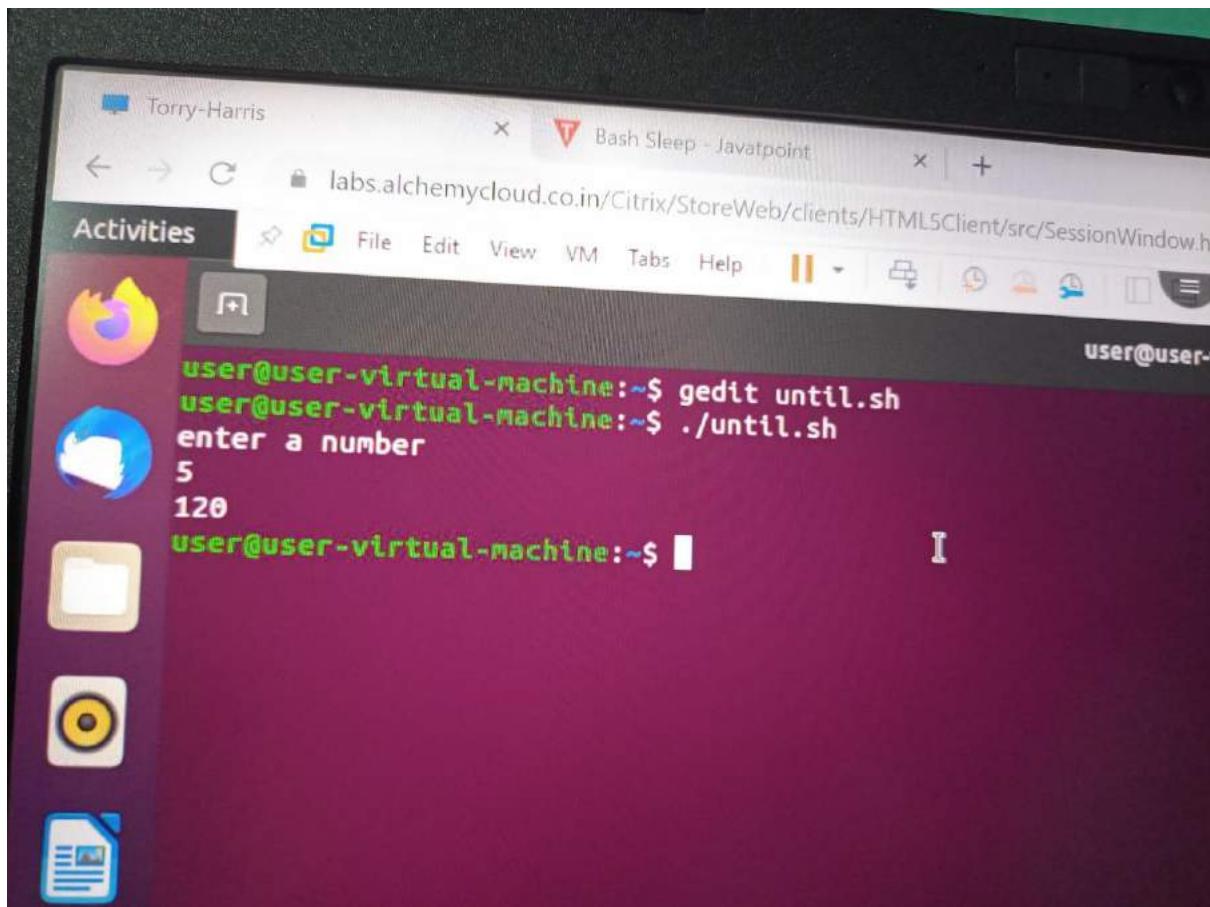
Torry-Harris

Bash Sleep - Javatpoint

File Edit View VM Tabs Help

*until.sh

```
1 #!/bin/bash
2 echo "enter a number"
3 read num
4 fact=1
5 while [ $num -gt 1 ]
6 do
7 fact=$(( fact * num ))
8 num=$(( num-1 ))
9 done
10 echo $fact
11
12
13
14
15
16
```



reverse(while)

A screenshot of a terminal window titled "reverse.sh". The window has a dark theme with light-colored text. The menu bar includes "File", "Edit", "View", "VM", "Tabs", and "Help". Below the menu is a toolbar with various icons. The main area contains the following shell script:

```
1
2 read -p "enter the number to be reversed" num
3 while [[ $num -gt 0 ]]
4 do
5 r=$((num%10))
6 num=$((num/10))
7 sum=$((sum*10+r))
8 done
9 echo $sum
```

Terry-Harris

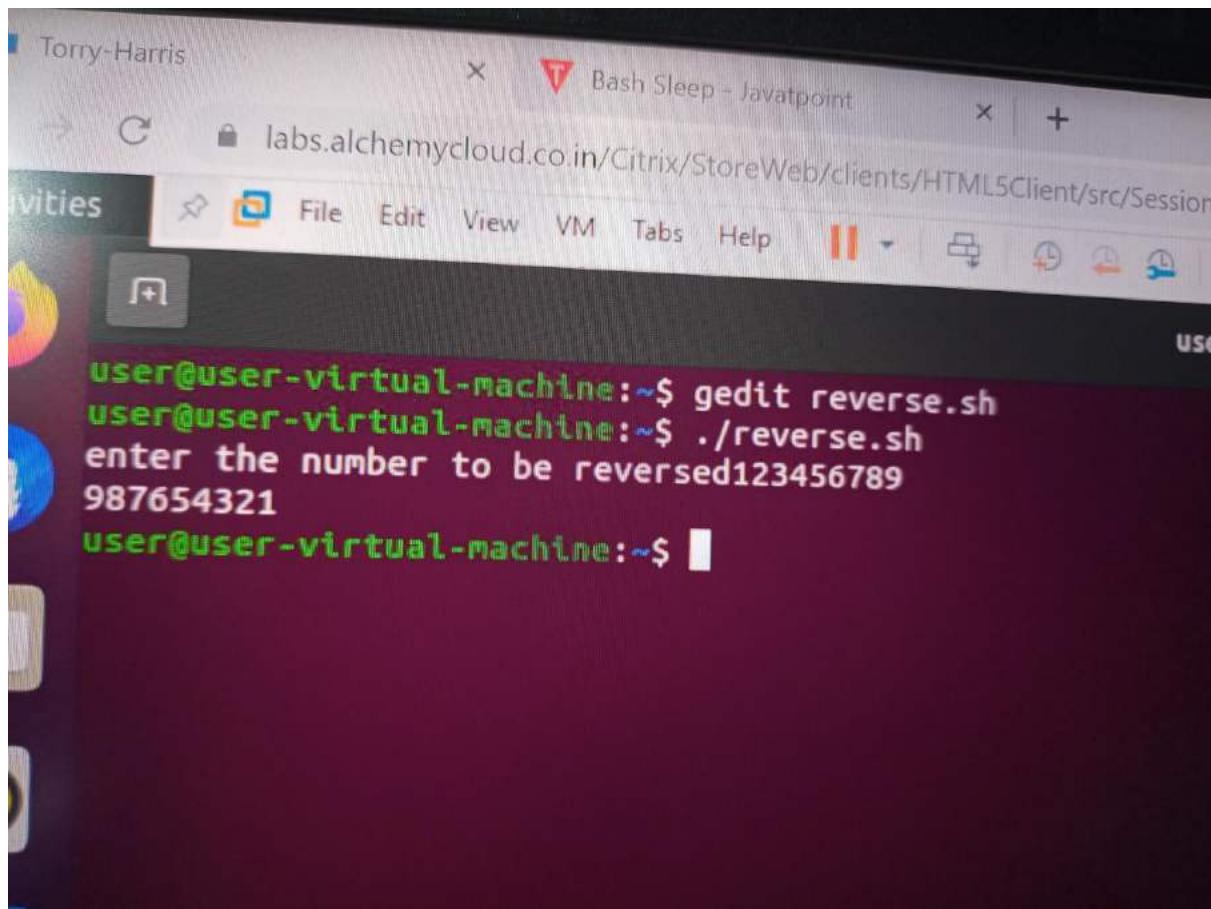
Bash Sleep - Javatpoint

labs.alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/Session

File Edit View VM Tabs Help

Activities

user@user-virtual-machine:~\$ gedit reverse.sh
user@user-virtual-machine:~\$./reverse.sh
enter the number to be reversed123456789
987654321
user@user-virtual-machine:~\$ █



until

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Bash Sleep - Javatpoint". The URL in the browser bar is "labs.alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/SessionWindow". The terminal window contains the following Bash script:

```
1#!/bin/bash
2i=1
3until [ $i -gt 10 ]
4do
5echo $i
6(( i++ ))
7done
8
9
10
11
12
13
14
```

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window is titled "Bash Sleep - Javatpoint" and has the URL "labs.alchemycloud.co.in/Citrix/StoreWeb/clients/HTML5Client/src/Sess" in the address bar. The terminal window contains the following text:

```
user@user-virtual-machine:~$ gedit until.sh
user@user-virtual-machine:~$ ./until.sh
1
2
3
4
5
6
7
8
9
10
user@user-virtual-machine:~$
```

factorial(until)

A screenshot of a Linux desktop environment, likely Ubuntu, showing a terminal window titled "Bash Sleep - Javatpoint". The terminal displays a Bash script for calculating the factorial of a number. The script uses a loop to multiply the current value of \$fact by \$num until \$num is greater than 5. It then prints the final value of \$fact.

```
1 #!/bin/bash
2 num=1
3 fact=1
4 until [ $num -gt 5 ]
5 do
6 fact=$((fact*num))
7 ((num++))
8 done
9 echo $fact
10
11
12
```

