

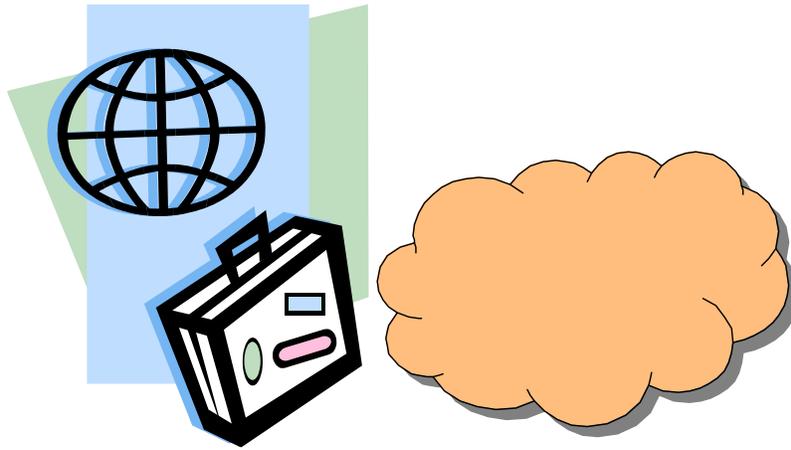


IMPORTANT LINUX COMMANDS YOU SHOULD KNOW

BY MANJUNATH.R

Important Linux Commands You Should Know

(A Pocket Guide For Beginners)



"The only true wisdom is in knowing you know nothing."

– Socrates

Manjunath.R

#16/1, 8th Main Road, Shivanagar, Rajajinagar, Bangalore560010, Karnataka, India

***Email:** manjunath5496@gmail.com

Disclaimer

Despite my best efforts to assure the accuracy of the material in this book, I do not accept and hereby disclaim any liability to any party for any loss, damage, or disruption caused by mistakes or omissions, whether caused by negligence, accident, or any other cause.

For any suggestions or concerns, please write to me: **manjunath5496@gmail.com**

© Copyright 2019 Manjunath.R

This work is licensed under a **Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License**.

(CC BY-NC-SA 4.0)

Under the terms of the cc-4.0-by license, you may:

- Share – copy and distribute the content in any form or media
- Remix, alter, and build upon the content for any non-commercial objective

As long as you comply by the conditions of the license, the licensor cannot revoke these rights.

You have to

- Provide proper recognition;
- Cite the license by including a link to it (<https://creativecommons.org/licenses/by-nc-sa/4.0/>);
and
- Specify whether (and if so, which) changes were made from the original.

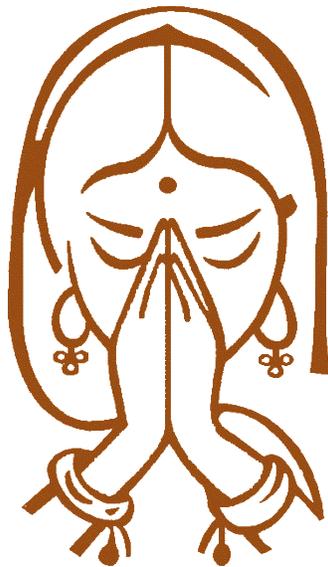
Dedication

I **dedicate this book** to every individual, programmer, teacher, educational institutions and enterprise corporations in every country of the world for their immense contributions towards the process of creating, designing, deploying and supporting software...



Acknowledgements

Without the amazing work of some renowned programmers, their creativity, and their inventiveness in the field of software programming, this book would not have been accomplished. I would like to use this opportunity to thank my dearest friend and well-wisher "**Lawrence**" for his unwavering support during the **COVID crisis** and for giving me access to all the resources I needed to finish this book. I want to express my gratitude to my family for their support and encouragement as I wrote this book, especially to my **mother**, who has been a tremendous source of inspiration in my life. I owe a lot of gratitude to my mother for teaching me how to be perseverant and strong in life. Finally, I want to emphasize how crucial patience is when writing a book or taking on any other project in life.



Foreword

I'm neither the proprietor of a well-known publishing house or a top IT firm with hundreds of in-house programmers who could easily produce anything I needed. I am a self-employed software engineer who is passionate about what I do, and believe me when I say that a lot of work and effort went into compiling this **comprehensive edition**. I'll be overjoyed if it helps even a few others reach their ideal positions in their professions.

Thank You

– Manjunath.R

A Complete Introduction to the Linux Command Line



Introduction

Want to use a Linux-based, open-source Unix-like operating system to increase the productivity and efficiency of your everyday computing? You can get ideas, strategies, and tips on how to do the task in this book. This book's main objective is to make system administration on Linux machines simple by giving you all the information you need. This book takes a task-oriented approach to system administration; as a result, it is structured around the requirements of a system administrator rather than the characteristics of the Linux operating system.

The command-line interface is one of the nearly all well built trademarks of Linux. There exists an ocean of Linux commands. Although, this to the end of time creates a problem: by all of so copious commands accessible to manage, you don't comprehend where and at which point to fly learning them, especially when you are learner. If you are facing this problem, and are peering for a painless method to begin your command line journey in Linux, you've come to the right place, as in this book, we will launch you to a hold of well liked and helpful Linux commands:

- You can copy, move, and remove files and folders
- Identify any network connectivity issues.
- Modify the ownership and permissions of files and folders.
- Manage, add, and create users and groups.
- Use "**su**" and "**sudo**" to securely access the root account.
- Create and modify text files without a **graphical editor**, and a lot more things.

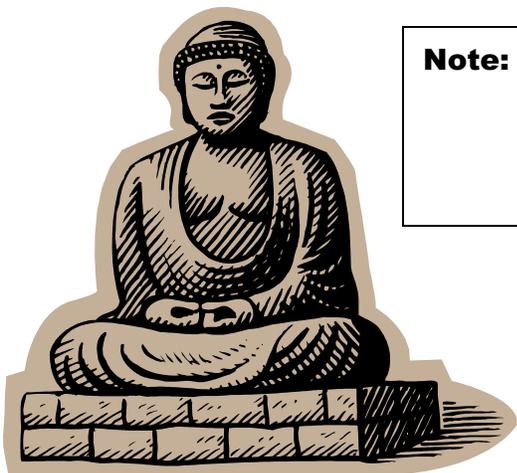
This book is heavily focused on Linux system administration. It will make an effort to explain how to use the command line interface more effectively. How does everything work? What is its scope? What method of use is ideal? However, by giving the reader a strong foundation in the use of the command line, a necessary tool for any significant system administration activity, it will set them up for further exploration of a command-line executable software or utility. There is no quick route to Linux wisdom. The command line is difficult to learn and requires significant effort. It's not that it's so difficult; rather, it's that it's so big. On a typical Linux system, you can use literally a plethora of programs via the command line. Be forewarned: learning the command line is not something you should take lightly. On the other hand, as 90% of cloud infrastructure and hosting services use Linux, mastering the Linux command line is quite advantageous.

Have Fun!

Remember that a command line is an interface that receives lines of text and converts them into instructions for your computer while you go through this book. A **graphical user interface** (GUI) is simply a command-line application abstraction. For instance, a command is carried out every time you click the "X" to close a window. Make sure to install Linux on your computer before diving into the most popular commands.

"All the best people in life seem to like LINUX."

—Steve Wozniak



Note:

- **Linux version used:** CentOS Linux release 7.3.1611 (Core)

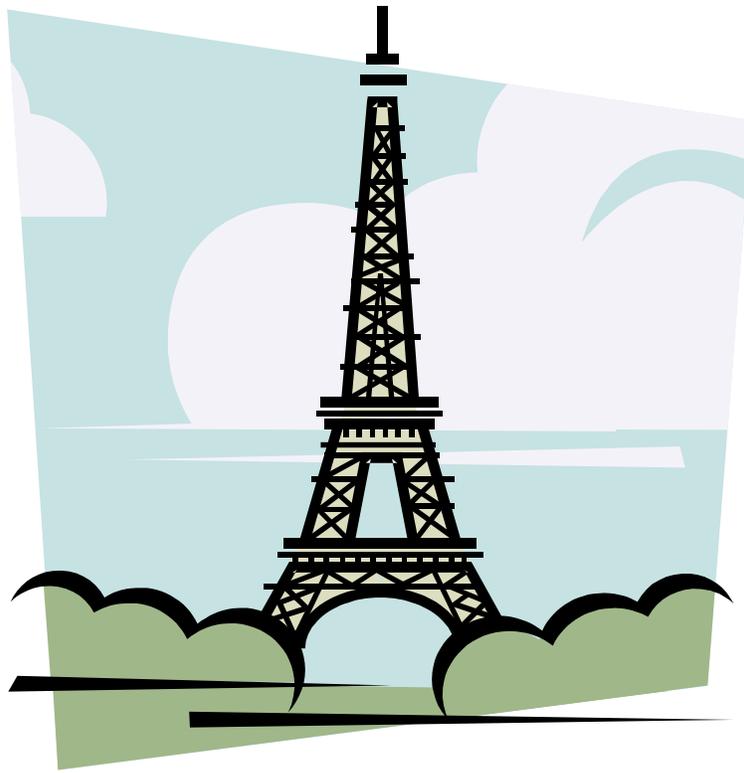
17 Principles of the philosophy of UNIX



- Principle of **Modularity**: A system should be composed of several components that are joined, collaborate well, and have clearly defined functions
- Principle of **Clarity**: Clearness is better than smartness
- Principle of **Composition**: Create software that can communicate with other software
- Principle of **Separation**: Programming mechanisms and rules should be kept distinct. Keep front-end interfaces and back-end engines separate
- Principle of **Simplicity**: Build for simplicity and only add complexity where necessary
- Principle of **Parsimony**: Only write a large program when it is clear by demonstration that nothing else will work
- Principle of **Transparency**: Design with visibility in view to simplify analysis and troubleshooting
- Principle of **Robustness**: Transparency and ease of use produce robustness
- Principle of **Representation**: Create programs easier to understand for any programmer involved in the project so that it can be maintained

When offered the choice, programmers should choose to complicate the data rather than the procedural logic of the software because complex data is simpler for us to understand than complex logic

- Principle of **Least Surprise**: Developers should be encouraged to create user-friendly, intuitive products
- Principle of **Silence**: Allow programmers and other programs to get the data they require from a program's output without having to interpret unnecessary extensive and detailed
- Principle of **Repair**: Programmers should create software that fails in a way that is simple to identify and diagnose
- Principle of **Economy**: Project development costs should be minimized
- Principle of **Generation**: Programmers should develop abstract, high-level programs that produce code rather than writing code by hand to decrease human error and save time
- Principle of **Optimization**: Before you can optimize it, get it working. Software should be developed and tested before being masterfully crafted by developers
- Principle of **Diversity**: Make programs flexible, enabling their use in ways other than those that their creators intended
- Principle of **Extensibility**: Increase the usefulness and lifespan of the developer's written code



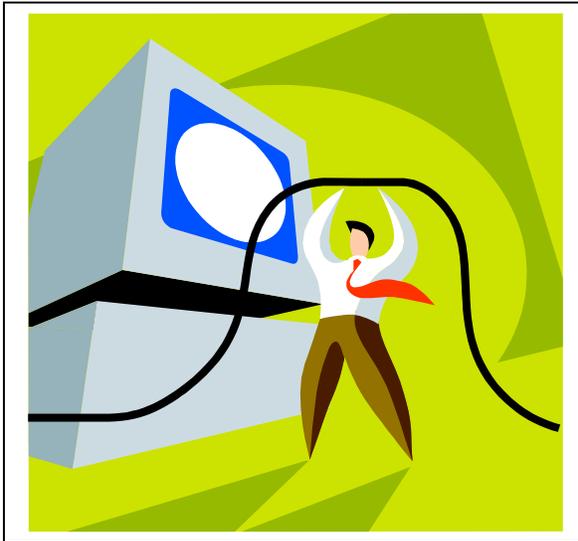
Better skills come with increased learning.

Your main focus as a novice should be on becoming familiar with the ins and outs of operating system architecture as well as discovering shortcuts and time-saving techniques.



CentOs is a wonderful option if you use Windows and want to learn Linux because it is one of the best Linux distributions for beginners. Your first few days using **CentOs** won't be that odd. But you must learn how to utilize Linux's command line interface if you want to experience its full capabilities. You will initially experience some difficulty learning several instructions. Although employing the instructions won't make you a genius, it will assist you in carrying out certain fundamental tasks. Here are the most basic **CentOs** commands for new users to ensure a smooth start. Let's get right into it!





"... being a Linux user is sort of like living in a house inhabited by a large family of carpenters and architects. Every morning when you wake up, the house is a little different. Maybe there is a new turret, or some walls have moved. Or perhaps someone has temporarily removed the floor under your bed."

~ **Unix for Dummies, 2nd Edition**

Linux Commands

Description:

Display system date and time.

Command:

```
date
```

Description:

Display calendar.

Command:

```
cal
```

Description:

Display date, time and calendar.

Command:

```
date & cal
```

Description:

Display August month 2016 year calendar.

Command:

```
cal 8 2016
```

Description:

Used to clear the terminal window.

Command:

```
clear
```

Description:

Exit from the terminal window.

Command:

```
exit
```

Description:

Display free and used system memory.

Command:

```
free
```

Description:

Display free and used system memory in bytes.

Command:

```
free -b
```

Description:

Display free and used system memory in kilobytes.

Command:

```
free -k
```

Description:

Display free and used system memory in megabytes.

Command:


```
free -m
```

Description:

Change user password.

Command:

```
passwd
```

Description:

Power-off the machine.

Command:

```
shutdown
```

Description:

Power-off the machine immediately.

Command:

```
shutdown -h now
```

Description:

Power-off the machine after 10 minutes.

Command:

```
shutdown -h +10
```

Description:

Print current working directory.

Command:

```
echo $PWD
```

Description:

Print previous working directory.

Command:

```
echo $OLDPWD
```

Description:

Executes the 11th command in command history.

Command:

```
!11
```

Description:

Reveals your command history.

Command:

```
history
```

Description:

Power off or reboot the Operating system.

Command:

```
sudo reboot
```

Description:

Display the IP address of the host.

Command:

```
ip address
```

Description:

List the size of files and directories.

Command:

```
ls -s
```

Description:

View mounted file systems.

Command:

mount

Description:

Display the information of disk usage of files and directories.

Command:

du

Description:

Tells you how long the system has been running.

Command:

uptime

Description:

Set current date as 02 Nov 1988.

Command:

```
date --set 1998-11-02
```

Description:

Set current time as 12:11:02 IST.

Command:

```
date --set 12:11:02
```

Description:

View Specific Disk Partition in Linux.

Command:

```
fdisk -l /dev/sda
```

Description:

Lists all files and directories in the present working directory.

Command:

```
ls
```

Description:

Report the process information.

Command:

```
ps
```

Description:

Display disk usage.

Command:

```
df
```

Description:

Display disk usage in gigabytes, megabytes, or kilobytes.

Command:

```
df -H
```

Description:

Delete every file and every directory.

Command:

```
rm -r *
```

Description:

Provides a quick overview of the currently running processes.

Command:

```
top
```

Description:

The system performs an immediate reboot.

Command:

```
reboot
```

Description:

Terminate processes without having to log out or reboot.

Command:

```
kill
```

Description:

Change the current working directory.

Command:

```
cd
```

Description:

Create a new session on the system.

Command:

```
login
```

Description:

List open files.

Command:

```
lsdf
```

Description:

List USB devices.

Command:

```
lsusb
```

Description:

Check the status of the network services.

Command:


```
service network status
```

Description:

Start the network service.

Command:

```
service network start
```

Description:

Stop the network service.

Command:

```
service network stop
```

Description:

Restart the network service.

Command:

```
service network restart
```

Description:

Report information about the users currently on the machine and their processes.

Command:

```
w
```

Description:

Display the current directory.

Command:

```
pwd
```

Description:

Displays CPU architecture information (such as number of CPUs, threads, cores, sockets, and more).

Command:

```
lscpu
```

Description:

Displays the number of processing units available to the current process.

Command:

```
nproc
```

Description:

The system performs an immediate reboot.

Command:

```
init 6
```

Description:

Power-off the machine.

Command:

```
init 0
```

Description:

List files by date.

Command:

```
ls -lrt
```

Description:

Report information about storage devices such as hard disks, flash drives etc.

Command:

```
lsblk
```

Description:

Show exit status of previous command.

Command:

```
echo $?
```

Description:

Lists a few useful info commands.

Command:

```
info
```

Description:

Prints current year's calendar.

Command:

```
cal -y
```

Description:

Check the status of all the services.

Command:

```
service --status-all
```

Description:

Display time in hh:mm:ss.

Command:

```
date +%T
```

Description:

Tells when the user last logged on and off and from where.

Command:

```
last -1 username
```

Description:

Sort files and directories by extension name.

Command:

```
ls -X
```

Description:

Display the manual for the pwd command.

Command:

```
man pwd
```

Description:

Displays information about running processes in the form of a tree.

Command:

```
pstree
```

Description:

Resets your terminal.

Command:

```
reset
```

Description:

Displays What date is it this Friday.

Command:

```
date -d fri
```

Description:

Displays the size of each individual file.

Command:

```
du -a
```

Description:

Display information about the Advanced configuration and power Interface.

Command:

```
acpi
```

Description:

Takes you two folders back.

Command:

```
cd ../../
```

Description:

Takes you to the previous directory.

Command:

```
cd -
```

Description:

Displays a list of shell built-in commands.

Command:

```
help
```

Description:

Lists your last logins.

Command:

```
last yourusername
```

Description:

Create a new directory called myfiles.

Command:

```
mkdir myfiles
```

Description:

Remove the directory myfiles.

Command:


```
rmdir myfiles
```

Description:

Disable password for a specific user "root1".

Command:

```
passwd -d root1
```

Description:

Switch to user "root1".

Command:

```
sudo su root1
```

Description:

Exit from the terminal window.

Command:

```
logout
```

Description:

Creates a user "root1".

Command:

```
useradd "root1"
```

Description:

Assign password to user "root1".

Command:

```
passwd "root1"
```

Description:

Repeats the last command.

Command:

```
!!
```

Description:

Display Who you are logged in as.

Command:

```
whoami
```

Description:

Display the login name of the current user.

Command:

```
logname
```

Description:

Report the name of the kernel.

Command:

```
uname
```

Description:

Print the kernel version.

Command:

```
uname -v
```

Description:

Print the operating system.

Command:

```
uname -o
```

Description:

Report the machine hardware name.

Command:

```
uname -m
```

Description:

Print version information and exit.

Command:

```
uname --version
```

Description:

Print the kernel release.

Command:

```
uname -r
```

Description:

Report the network node hostname.

Command:

```
uname -n
```

Description:

Display all port connections (both TCP and UDP).

Command:

```
netstat -a
```

Description:

Display only TCP (Transmission Control Protocol) port connections.

Command:

```
netstat -at
```

Description:

Display only UDP (User Datagram Protocol) port connections.

Command:

```
netstat -au
```

Description:

Display all active listening ports.

Command:

```
netstat -I
```

Description:

Display all active listening TCP ports.

Command:

```
netstat -It
```

Description:

Display all active listening UDP ports.

Command:

```
netstat -lu
```

Description:

Reveal all the information about the current user (user id, username, group id, group name etc.).

Command:

```
id
```

Description:

Reveal all the information about the user "root1" (user id, username, group id, group name etc.).

Command:

```
id root1
```

Description:

Print the machine's architecture.

Command:

```
arch
```

Description:

Display the list of available fonts.

Command:

```
fc-list
```

Description:

Create two directories (myfiles, files).

Command:

```
mkdir myfiles files
```

Description:

install apache (CentOS).

Command:

```
yum install httpd
```

Description:

install apache (Ubuntu).

Command:

```
apt install httpd
```

Description:

upgrade apache (CentOS).

Command:

```
yum update httpd
```

Description:

upgrade apache (Ubuntu).

Command:


```
apt update httpd
```

Description:

uninstall apache (CentOS).

Command:

```
yum remove httpd
```

Description:

uninstall apache (Ubuntu).

Command:

```
apt remove httpd
```

Description:

Display usage summary for the command (date).

Command:

```
date --help
```

Description:

List active connections to/from system.

Command:

```
ss -tup
```

Description:

List internet services on a system.

Command:

```
ss -tupl
```

Description:

Display all active UNIX listening ports.

Command:

```
netstat -lx
```

Description:

Display all the active interfaces details.

Command:

```
ifconfig
```

Description:

Display information of all network interfaces.

Command:

```
ifconfig -a
```

Description:

Compare the contents of two files (1.txt, 2.txt).

Command:

```
diff 1.txt 2.txt
```

Description:

Tells you how many lines, words, and characters there are in a file (1.txt).

Command:

```
wc 1.txt
```

Description:

Compresses file (1.txt), so that it take up much less space.

Command:

```
gzip 1.txt
```

Description:

Uncompresses file (1.txt) compressed by gzip.

Command:

```
gunzip 1.txt
```

Description:

Examine the contents of the file (1.txt).

Command:

```
cat 1.txt
```

Description:

Display calendar.

Command:

```
ncal
```

Description:

Removes the file (1.txt).

Command:

```
rm 1.txt
```

Description:

Rename a file named 1.txt to 0.txt.

Command:

```
mv 1.txt 0.txt
```

Description:

Replace the contents of 0.txt with that of 1.txt.

Command:

```
cp 1.txt 0.txt
```

Description:

Create a empty file (test.txt).

Command:

```
touch test.txt
```

Description:

Print the last 10 lines of a file (1.txt).

Command:

```
tail 1.txt
```

Description:

Print N number of lines from the file (1.txt).

Command:

```
tail -n N 1.txt
```

Description:

Prints the number of words in a file (1.txt).

Command:

```
wc -w 1.txt
```

Description:

Prints the number of characters from a file (1.txt).

Command:

```
wc -m 1.txt
```

Description:

Prints the length of the longest line in a file (1.txt).

Command:

```
wc -L 1.txt
```

Description:

Print information about usb ports, graphics cards, network adapters etc.

Command:

```
lspci
```

Description:

View contents of a file (1.txt).

Command:

```
less 1.txt
```

Description:

Display calendar (last month, current month, and next month).

Command:

```
cal -3
```

Description:

Compare the contents of three files (1.txt, 2.txt, 3.txt) line by line.

Command:

```
diff3 1.txt 2.txt 3.txt
```

Description:

Compare two files (1.txt, 2.txt) line-by-line.

Command:

```
comm 1.txt 2.txt
```

Description:

Perform byte-by-byte comparison of two files (1.txt, 2.txt).

Command:

```
cmp 1.txt 2.txt
```

Description:

Prints the CRC checksum and byte count for the file "myfiles.txt".

Command:


```
cksum myfiles.txt
```

Description:

Append contents of files (1.txt, 2.txt) into one file (0.txt).

Command:

```
cat 1.txt 2.txt > 0.txt
```

Description:

Append contents of files (1.txt, 2.txt, 3.txt) into one file (0.txt).

Command:

```
sed r 1.txt 2.txt 3.txt > 0.txt
```

Description:

Append contents of files (1.txt, 2.txt, 3.txt) into one file (0.txt).

Command:

```
sed h 1.txt 2.txt 3.txt > 0.txt
```

Description:

Append contents of files (1.txt, 2.txt, 3.txt) into one file (0.txt).

Command:

```
sed -n p 1.txt 2.txt 3.txt > 0.txt
```

Shortcuts:

ctrl+c	Halts the current command	
ctrl+z	Stops the current command	
ctrl+d	Logout the current session	
ctrl+w	Erases one word in the current line	
ctrl+u	Erases the whole line	
ctrl+r	Type to bring up a recent command	

Description:

Writes contents of a file (0.txt) to output, and prepends each line with line number.

Command:

```
nl 0.txt
```

Description:

Create a empty file (test1.txt) inside a directory (test).

Command:

```
mkdir test  
cd test  
pwd  
touch test1.txt
```

Description:

Gather information about hardware components such as CPU, disks, memory, USB controllers etc.

Command:

```
sudo lshw
```

Description:

Gather information about file system partitions.

Command:

```
sudo fdisk -l
```

Description:

Displays the line (good morning) in which the string (good) is found in the file (1.txt).

Command:

```
grep good 1.txt
```

Description:

Append contents of files (1.txt, 2.txt, 3.txt) into one file (0.txt) using for loop.

Command:

```
for i in {1..3}; do cat "$i.txt" >> 0.txt; done
```

Description:

Search for files (test.txt, test1.txt, test2.txt, test.php, test.html) in a directory as well as its sub-directories.

Command:

```
find test*
```

Description:

Displays status related to a file (1.txt).

Command:

```
stat 1.txt
```

```
###
| Command | Description |
|:-----|:-----:|
| vi | Open vi editor |
| i | Go to Insert mode |
| | |
| a =20; b =64; | |
| print (a + b); | |
| Hit Escape to return to Normal mode. |
| :w hello.py | Save text |
| :q | Quit |
| python hello.py |Print the output:84 |
```

Description:

Download the file (file.txt) from url "http://website.com/files/file.txt".

Command:

```
wget http://website.com/files/file.txt
```

Description:

Display host's numeric ID in hexadecimal format.

Command:

```
hostid
```

Description:

Display file type of the file (myfiles.txt).

Command:

```
file myfiles.txt
```

Description:

Create a file (myfiles.txt) containing a text (Hello World).

Command:

```
echo 'Hello World' > myfiles.txt
```

Description:

Create a file (myfile.txt) containing a text (Hello World).

Command:

```
printf 'Hello World' > myfile.txt
```

Description:

Display IP address of the hostname.

Command:

```
hostname -i
```

Description:

Add a new line of text to an existing file (1.txt).

Command:

```
echo "Hello world!" >> 1.txt  
echo "this is 2nd line text" >> 1.txt  
echo "last line!" >> 1.txt
```

Description:

Displays a single line description about a command (cal).

Command:

```
whatis cal
```

```
###
| Command | Description |
|:-----|:-----:|
| vi | Open vi editor |
| i | Go to Insert mode |
| Type some text. | |
| Hit Escape to return to Normal mode. | |
| :w test.txt | Save text |
| :q | Quit |
| :q! |Quit without saving |

###
| Command | Description |
|:-----|:-----:|
| vi | Open vi editor |
| i | Go to Insert mode |
| $name = "Paul"; | |
| print "$name"; | |
| Hit Escape to return to Normal mode. | |
| :w hello.pl | Save text |
| :q | Quit |
| perl hello.pl |Print the output: Paul |

###
| Command | Description |
|:-----|:-----:|
| vi | Open vi editor |
| i | Go to Insert mode |
| echo "What is your name?" | |
| read PERSON | |
| echo "Hello, $PERSON" | |
| Hit Escape to return to Normal mode. | |
| :w hello.sh | Save text |
| :q | Quit |
| sh hello.sh | Output: |
| | What is your name? |
| | If you enter: Zara Ali |
| | Hello, Zara Ali |
```

Description:

Check the network connectivity between host (your connection) and server (Google server).

Command:

```
ping google.com
```

Description:

Find the location of source/binary file of a command (cal).

Command:

```
whereis cal
```

There are 2 ways to use the command:

- Numeric mode
- Symbolic mode

```
# Overwrite existing file
$ echo "Albert Einstein" > 1.txt
# Append a second line
$ echo "Alan Turing" >> 1.txt
```

Numeric mode	Permission Type	Symbolic mode
0	No Permission	---
1	Execute	--x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r--
5	Read + Execute	r-x
6	Read + Write	rw-
7	Read + Write + Execute	rwX

```
[manju@localhost ~]$ ps -ef | grep sshd
root      988      1  0 06:14 ?        00:00:00 /usr/sbin/sshd
manju    3501    3461  0 06:24 pts/0    00:00:00 grep --color=auto sshd
```

```
# Check if the SSH server (sshd) is running
```



```
cd /etc && ls
```

```
# Execute ls after cd /etc
```

```
rm myfiles.txt && echo success || echo failed
```

Print 'success' if myfiles.txt is removed and print 'failed' if it is not removed

```
[manju@localhost ~]$ echo This is the $SHELL shell
```

```
This is the /bin/bash shell
```

```
[manju@localhost ~]$ echo This is $SHELL on computer $HOSTNAME
```

```
This is /bin/bash on computer localhost.localdomain
```

```
[manju@localhost ~]$ echo The user ID of $USER is $UID
```

```
The user ID of manju is 1000
```

```
[manju@localhost ~]$ echo My home directory is $HOME
```

```
My home directory is /home/manju
```

```
[manju@localhost ~]$ bash -c 'echo $SHELL $HOME $USER'
```

```
/bin/bash /home/manju manju
```

```
[manju@localhost ~]$ env -i bash -c 'echo $SHELL $HOME $USER'
```

```
/bin/bash
```

```
env LANG=C bash -c 'ls test[a-z].txt'
```

```
testa.txt testb.txt testc.txt
```

```
env LANG=en_US.UTF-8 bash -c 'ls test[a-z].txt'
```

```
testa.txt testA.txt testb.txt testc.txt testC.txt
```

```
[manju@localhost ~]$ prefix=John
```

```
[manju@localhost ~]$ echo Hello ${prefix}Dalton and ${prefix}Humphrys
```

```
Hello JohnDalton and JohnHumphrys
```

```
echo $(a=5;echo $a)
```

5

```
echo 'a=5;echo $a'
```

a=5;echo \$a

```
[manju@localhost ~]$ touch myfiles.txt
```

```
[manju@localhost ~]$ cat myfiles.txt
```

```
Hello World
```

```
[manju@localhost ~]$ !to
```

```
touch myfiles.txt
```

```
[manju@localhost ~]$ echo $HISTSIZE
```

```
1000
```

The number of commands that are stored in memory in a history list while your bash session is ongoing

```
[manju@localhost ~]$ echo $HISTFILE
```

```
/home/manju/.bash_history
```

Holds the name and location of your **Bash history file**

```
echo $HISTFILESIZE
```

```
1000
```



How many commands can be stored in the `.bash_history` file

```
[manju@localhost ~]$ ls *file1.txt
```

```
file1.txt
```

```
[manju@localhost ~]$ ls f*file1.txt
```

```
file1.txt
```

```
[manju@localhost ~]$ ls f*1.txt
```

```
file1.txt
```

```
[manju@localhost ~]$ ls file?.txt
```

```
file1.txt file2.txt file3.txt
```

```
[manju@localhost ~]$ ls fil?1.txt
```

```
file1.txt
```

```
[manju@localhost ~]$ ls fil???.txt
```

```
file1.txt file2.txt file3.txt
```

```
[manju@localhost ~]$ ls file???.txt
```

```
file23.txt file34.txt
```

```
[manju@localhost ~]$ ls test[5A].txt
```

```
testA.txt
```

```
[manju@localhost ~]$ ls test[A5].txt
```

```
testA.txt
```

```
[manju@localhost ~]$ ls file[!5]*.txt
```

```
file123.txt file1.txt file23.txt file2.txt file34.txt file3.txt
```

```
[manju@localhost ~]$ ls file[!5]?.txt
```

```
file23.txt file34.txt
```

```
[manju@localhost ~]$ ls [a-z]ile?.txt
```

```
file1.txt file2.txt file3.txt
```

```
[manju@localhost ~]$ ls [A-Z]ile?.txt
```

```
file1.txt file2.txt file3.txt
```

```
[manju@localhost ~]$ echo \*
```

```
*
```

```
[manju@localhost ~]$ echo '*'
```

```
*
```

```
[manju@localhost ~]$ echo "*"
```

```
*
```

```
[manju@localhost ~]$ ls [a-z]*[0-9].txt
```

```
file123.txt file1.txt file23.txt file2.txt file34.txt file3.txt
```

List all `.txt` files starting with a letter and ending in a number

```
ls ?????
```

```
# List all files that have exactly five characters
```

```
ls [fF]*[3A].txt
```

```
# List all .txt files that start with f or F and end with 3 or A
```

```
ls f[iR]*[0-9].txt
```

```
# List all .txt files that start with f have i or R as second character and end in a number
```

```
ls [!f]*.txt
```

```
# List all .txt files that do not start with the letter "f"
```

```
[manju@localhost ~]$ echo Einstein2 | sed 's/2/36/'
```

```
Einstein36
```

```
[manju@localhost ~]$ echo Einstein36 | sed 's/Einstein/Hilbert/'
```

```
Hilbert36
```

```
[manju@localhost ~]$ echo Hawking6 Lucy8 | sed 's/Hawking/Lucy/'
```

```
Lucy6 Lucy8
```

```
[manju@localhost ~]$ echo Lucy3 Lucy6 | sed 's/Lucy/Hawking/g'
```

```
Hawking3 Hawking6
```

```
[manju@localhost ~]$ who | cut -d' ' -f1 | sort
```

```
manju
```

```
manju
```

Display a sorted list
of logged on users

```
[manju@localhost ~]$ who | cut -d' ' -f1 | sort | uniq
```

```
manju
```

Display a sorted
list of logged on
users - but every
user only once

```
[manju@localhost ~]$ grep bash /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
```

```
manju:x:1000:1000:su,root,yopp,hhhh:/home/manju:/bin/bash
```

Display a list of all bash user accounts on this computer

```
grep bash /etc/passwd | cut -d: -f1 | sort > bu.txt
```

```
# Place a sorted list of all bash users in bu.txt
```

```
who | cut -d' ' -f1 | sort > users.txt
```

```
# Place a sorted list of all logged on users in users.txt
```

```
ls /etc | grep conf
```

```
# List of all filenames in /etc that contain the string "conf" in their filename
```

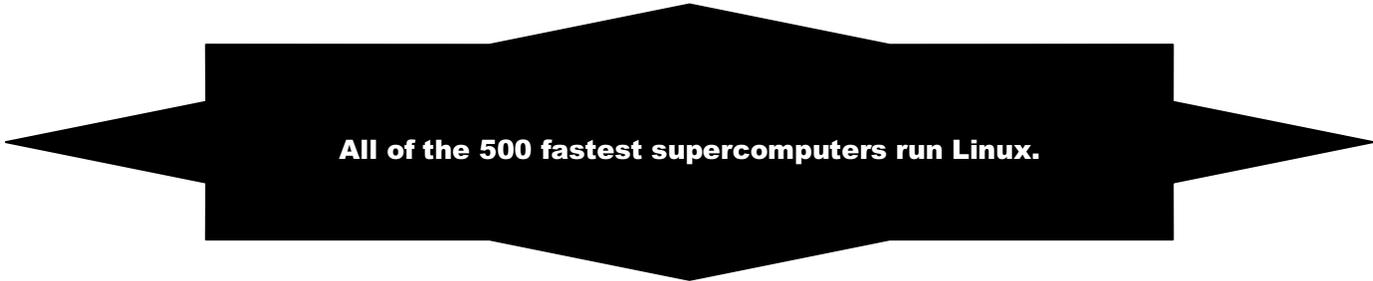
```
ls /etc | grep -i conf | sort
```

Display a sorted list of all files in /etc
that contain the case insensitive string
"conf" in their filename

90% of the public cloud workload is run on Linux distros.

The first ever Linux kernel just occupied only 65 KB.

<pre>import subprocess subprocess.call('linux command')</pre>	<pre>import os os.system('linux command')</pre>
---	---



<pre>import os os.system('ls')</pre>	<p>List all the files and directories in the current directory</p>
<pre>import subprocess subprocess.call('ls')</pre>	

Execution of the linux command "ls" using the python program

Command:

```
last reboot
```

Description:

Show system reboot history

Command:

```
dmesg
```

Description:

Displays the messages from the kernel ring buffer (a data structure that records messages related to the operation of the kernel)

Command:

```
cat /proc/cpuinfo
```

Description:

Display CPU information

Command:

```
cat /proc/meminfo
```

Description:

Display memory information

Command:

```
lspci -tv
```

Description:

Display PCI (Peripheral Component Interconnect) devices

Command:

```
lsusb -tv
```

Description:

Display USB devices

Command:

```
free -h
```

Description:

Display free and used memory (-h for human readable, -m for MB, -g for GB)

Command:

```
mpstat 1
```

Description:

Display processor related statistics

Command:

```
vmstat 1
```

Description:

Display virtual memory statistics

Command:

```
iostat 1
```

Description:

Display Input / Output statistics

Command:

```
watch df -h
```

Description:

Execute "df -h" command, showing periodic updates

Command:

```
ps -ef
```

Description:

Display all the currently running processes on the system

Command:

```
ip a
```

Description:

Display all network interfaces and IP address

Command:

```
dig wikipedia.org
```

Description:

Display DNS information for domain (wikipedia.org)

Command:

```
host wikipedia.org
```

Description:

Display the IP address details of the specified domain (wikipedia.org)

Command:

```
netstat -nutlp
```

Description:

Display listening Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) ports and corresponding programs

Command:

```
rpm -qa
```

Description:

List all installed packages

Command:

```
yum list installed
```

Description:

List all installed packages (CentOS)

Command:

```
yum info httpd
```

Description:

Display description and summary information about package "httpd" (CentOS)

Command:

```
du -ah
```

Description:

Display disk usage for all files and directories in human readable format

Command:

```
du -sh
```

Description:

Display total disk usage off the current directory

Command:

```
cd /etc
```

Description:

Change to the /etc directory

Command:

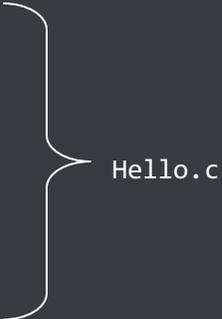
```
ps -A
```

Description:

List the status of all the processes along with process id and PID

Command:

```
#include <stdio.h>
int main()
{
    printf("Hello world\n");
    return 0;
}
```



Hello.c

```
gcc Hello.c
```

Description:

Compile the C program saved in Hello.c file

Command:

```
#include <iostream>
int main()
{
std::cout << "Hello world!";
    return 0;
}

g++ Hello.cpp
```



Hello.cpp

Description:

Compile the C++ program saved in Hello.cpp file

Command:

```
tty
```

Description:

Displays the file name of the terminal connected to standard input

Command:

```
public class MyClass {  
    public static void main(String [] args) {  
        System.out.println("Hello, World!");  
    }  
}  
  
javac MyClass.java
```

**Description:**

Compile the Java program saved in MyClass.java file using javac compiler

Command:

```
od -b myfiles.txt
```

Description:

Displays the contents of myfiles.txt file in octal format

Command:

```
od -c myfiles.txt
```

Description:

Displays the contents of `myfiles.txt` file in character format

Command:

```
od -An -c myfiles.txt
```

Description:

Displays the contents of `myfiles.txt` file in character format but with no offset information

Command:

```
csplit myfiles.txt 13 62 101
```

Description:

If the file `myfiles.txt` has 123 lines, the `csplit` command would create four files: the `xx00` file would contain lines 1–12, the `xx01` file would contain lines 13–61, the `xx02` file would contain lines 62–100, the `xx03` file would contain lines 101–123

Command:

```
md5sum myfiles.txt
```

Description:

Prints a 32-character (128-bit) checksum of myfiles.txt file using the MD5 algorithm

Command:

```
more myfiles.txt
```

Description:

Displays the content of myfiles.txt file

Command:

```
sha1sum myfiles.txt
```

Description:

Prints SHA1 (160-bit) checksum of myfiles.txt file

SHA 1 → Secure Hash Algorithm 1

Command:

```
shred myfile.txt
```

Description:

Overwrites the myfile.txt file repeatedly – in order to make it harder for even very expensive hardware probing to recover the data

Command:

```
cat myfile.txt
```

```
01. Einstein  
02. Newton  
03. Maxwell  
04. Tesla  
05. Edison
```

```
tac myfile.txt
```

```
05. Edison  
04. Tesla  
03. Maxwell  
02. Newton  
01. Einstein
```

Description:

Print the lines of myfile.txt in reverse (from last line to first)

Command:

```
uniq myfiles.txt
```

Description:

Delete repeated lines in the file (myfiles.txt).

Command:

```
chkconfig --list
```

Description:

Displays a list of system services and whether they are started (on) or stopped (off) in run levels 0-6

Command:

```
halt -p
```

Description:

Power-off the system

Command:

```
xdg-open myfiles.txt
```

Description:

Open a file (myfiles.txt).

Command:

```
lastlog
```

Description:

Prints the details of the last login (login-name, port and last login time)

Command:

```
lastlog -t 1
```

Description:

Displays the login information (1 day ago)

Command:

```
lastlog -u manju
```

Description:

Display lastlog information for a particular user (manju)

Command:

```
cat /etc/passwd
```

```
more /etc/passwd
```

```
less /etc/passwd
```

```
getent passwd
```

Description:

List all users on Linux

Command:

```
tail -5 /etc/passwd
```

```
head -5 /etc/passwd
```

Description:

List last 5 users on Linux

List first 5 users on Linux

Command:

```
wall "The system will be shutdown in 10 minutes."
```

Description:

The message (The system will be shutdown in 10 minutes.) **will be broadcasted to all users that are currently logged in**

Command:

```
chage -l manju
```

Description:

List the password and its related details for a user (manju)

Command:

```
chage -M 10 manju
```

Description:

Set Password Expiry Date for an user (manju)

Command:

```
chage -E "2020-07-30" manju
```

Description:

Set the Account Expiry Date for an User (manju)

Command:

```
chage -I 10 manju
```

Description:

Force the user (manju) account to be locked after 10 inactivity days

```
cat /etc/hostname
```

```
→ localhost.localdomain
```

```
hostname
```

```
→ localhost.localdomain
```

Display the hostname of the system

```
nmtui
```

```
# Configure a network interface IPv4 address
```

```
yum check-update
```

```
# Check whether any updates are available for your installed packages
```

```
yum search httpd
```

```
# Find any packages containing the specified keyword "httpd"
```

```
ls /etc
```

```
# List the contents of /etc
```

```
ls /bin /sbin
```

```
# List the contents of /bin and /sbin
```

```
ls -al ~
```

```
# List all the files (including hidden files) in the home directory
```

```
ls -lh /boot
```

```
# List the files in /boot in a human readable format
```

```
mkdir ~/mydir
```

```
# Create a directory "mydir" under home directory
```

```
cd /etc ; mkdir ~/mydir
```

```
# Change to the /etc directory and create a directory "mydir" under home directory.
```

```
rm -i file.txt
```

```
rm: remove regular empty file `file.txt'?
```

```
If we type "yes"
```

```
file.txt is removed
```

```
If we type "no"
```

```
file.txt is not removed
```

```
rename .txt .backup *.txt
```

```
# Renames all .txt files replacing.txt with .backup
```

```
ls
```

```
file.txt  cod.txt  conf.txt
```

```
rename file FILE *
```

```
ls
```

```
FILE.txt  cod.txt  conf.txt
```

```
file /bin/cat /etc/passwd /usr/bin/passwd
```

```
Display the type of file of /bin/cat, /etc/passwd and /usr/bin/passwd
```

Command:

```
ftp 192.168.42.77
```

Description:

Connect to an FTP server at remote server IP address "192.168.42.77"

Command:

```
arp -a
```

Description:

Lists all the peers connected at various interfaces along with their MAC
Addresses and IP addresses

Command:

```
dnsdomainname
```

Description:

Display the system's DNS domain name

Command:

```
domainname
```

Description:

Display the name of the domain your machine belongs to

Command:

```
echo 'Hello World!' | base64
```

Output: SGVsbG8gV29ybGQhCg==

Description:

Encode text (Hello World!) to base64

Command:

```
echo 'SGVsbG8gV29ybGQhCg==' | base64 -d
```

Output: Hello World!

Description:

Decode (SGVsbG8gV29ybGQhCg==) to text (Hello World!)

Command:

```
fc-cache -f -v
```

Description:

Build font information cache files

Command:

```
cat 1.txt  
  
Einstein  
Newton  
Albert  
  
fmt 1.txt  
  
Einstein Newton Albert
```

Description:

Formats text in a single line

```
df -h | sort -rnk 5 | head -3 | \
awk '{ print "Partition " $6 "\t: " $5 " full!" }'
```

```
Partition /boot : 51% full!
Partition /      : 29% full!
Partition /run   : 2% full!
```

```
awk 'BEGIN { FS=":" } { print $1 "\t" $5 }' /etc/passwd
```

```
# Display all the users on your system
```

```
ls *.xml
```

```
1.xml 2.xml
```

```
ls *.xml > list.txt
```

```
[manju@localhost ~]$ cat list.txt
```

```
1.xml
```

```
2.xml
```

```
for i in `cat list.txt`; do cp "$i" "$i".md ; done
```

```
[manju@localhost ~]$ ls
```

```
12.txt  2.xml.md  Documents  file34.txt  Music      Pictures  tree.cpio
13.txt  3.txt     Downloads  file3.txt   mydi       Public    users.txt
145.txt all       echo       FILE.backup mydir      SHOW      Videos
1.txt   allfiles.txt file       file.md    mydir1     Templates
1.xml  bu.txt    file123.txt first.bash myfiles.txt test
1.xml.md Desktop  file1.txt  first.txt  myFILES.txt.xz testA.txt
2.txt  DICT     file23.txt fool.txt   newdir     testB.txt
2.xml  dir      file2.txt  list.txt  nohup.out  text
```

```
[manju@localhost ~]$ df -h /  
  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/sda3      18G   5.2G  13G   29%  /
```

Check the actual
used space on the
current root device

```
less /proc/modules
```

```
# Display information about what kernel-modules are loaded on your system
```

```
[manju@localhost ~]$ free -tm
```

	total	used	free	shared	buff/cache	available
Mem:	999936	511156	73480	8572	415300	284236
Swap:	2097148	0	2097148			

Display the **memory usage** including
totals in megabytes

```
[manju@localhost ~]$ date --date="3 months 1 day ago"
```

```
Mon Jul 18 23:17:47 PDT 2022
```

Print the date **3 months and 1 day ago** from the current date

```
[manju@localhost ~]$ date -d "3 days"
```

```
Fri Apr 22 23:20:01 PDT 2022
```

Print the date **3 days in the future** from now

```
[manju@localhost ~]$ cat myfiles.txt
```

```
Hello World
```

```
[manju@localhost ~]$ cat myfiles.txt | tr 'H' 'A' > myfilesB.txt
```

```
[manju@localhost ~]$ cat myfilesB.txt
```

```
Aello World
```



```
[manju@localhost ~]$ fgrep 'He' myfiles.txt
```

```
Hello World
```

Look for the string "He" in the file "myfiles.txt"

Command:

```
lsattr
```

Description:

List the files in the current directory

Command:

```
cp {*.txt,*.md} ~
```

Description:

Copy the files ending with .txt or .md to the user's home directory

```
[manju@localhost ~]$ cat myfiles.txt
```

```
Hello World
```

```
[manju@localhost ~]$ grep --color -i Hello myfiles.txt
```

```
Hello World
```

```
ls file*
```

```
# List all files in the current directory starting with "file"
```

```
ls *file
```

```
# List all files in the current directory ending with "file"
```

```
cat phy.txt
```

```
Albert Einstein was a German-born theoretical physicist, widely acknowledged to be one of the greatest physicists of all time. Einstein is known for developing the theory of relativity, but he also made important contributions to the development of the theory of quantum mechanics.
```

```
fmt -w 1 phy.txt
```

```
Albert  
Einstein  
was  
a  
German-born  
theoretical  
physicist,  
widely  
acknowledged  
to  
be  
one  
of  
the  
greatest  
physicists  
of  
all  
time.  
Einstein  
is  
known  
for  
developing
```

the
theory
of
relativity,
but
he
also
made
important
contributions
to
the
development
of
the
theory
of
quantum
mechanics.

```
cat phy.txt
```

```
Albert Einstein was a German-born theoretical physicist, widely acknowledged to  
be one of the greatest physicists of all time. Einstein is known for developing  
the theory of relativity, but he also made important contributions to the  
development of the theory of quantum mechanics.
```

```
fold -w 20 phy.txt
```

```
Albert Einstein was
```

a German-born theoretical physicist, widely acknowledged to be one of the greatest physicists of all time. Einstein is known for developing the theory of relativity, but he also made important contributions to the development of the theory of quantum mechanics.

Command:

```
tracert google.com
```

Description:

Prints the route that a packet takes to reach the Google (172.217.26.206) host from the local machine

Command:

```
cat 1.txt
```

```
Einstein  
Newton  
Albert  
  
gzip 1.txt  
  
zcat 1.txt.gz  
  
Einstein  
Newton  
Albert
```

Description:

View the contents of zipped file

Command:

```
zdiff 1.txt.gz 2.txt.gz
```

Description:

Compare the contents of two zipped files (1.txt.gz, 2.txt.gz)

Command:

```
ss | less
```

Description:

List all connections

Command:

```
ss -aA tcp
```

Description:

Filter out TCP (Transmission Control Protocol) connections

Command:

```
ss -aA udp
```

Description:

Filter out UDP (User Datagram Protocol) connections

Command:

```
ss -lnt
```

Description:

Display only listening sockets

Command:

```
ss -ltp
```

Description:

Print process name and PID

Command:

```
ss -s
```

Description:

Print summary statistics

Command:

```
ss -t16
```

Description:

Display only IPv6 connections

Command:

```
ss -t1 -f inet
```

Description:

Display only IPv4 socket connections

Command:

```
ss -t4 state established
```

Description:

Display all IPv4 TCP sockets that are in connected state

Command:

```
pmap 3244
```

Description:

View the memory map of a process with Process ID (3244)

Command:

```
apropos -r 'remove file'
```

Description:

Find command that removes file

Command:

```
apropos editor
```

Description:

Display information about the editing programs that are available on a system

Command:

```
apropos pstree
```

Description:

Provide information about the pstree command (which displays the names of the processes currently on the system in the form of a tree diagram)

The **apropos command** is useful when you know what you want to do, but you have no idea what command you should be using to do it. If you were wondering how to locate files, for example, the commands

```
apropos find
```

and

```
apropos locate
```

would have a lot of suggestions to offer.

```
basename /etc/passwd
```

```
Output: passwd
```

```
basename /usr/local/apache2/conf/httpd.conf
```

```
Output: httpd.conf
```

```
echo a b c d e f | xargs
```

```
Output: a b c d e f
```

```
echo a b c d e f | xargs -n 3
```

```
Output:
```

```
a b c  
d e f
```

} display only 3 items per line

Command:

```
env
```

Description:

Print out a list of all environment variables

Command:

```
printenv HOME
```

Description:

Print HOME variable value

```
cat score.txt
```

```
Albert-30
```

```
John-50
```

```
William-80
```

```
Stephen-20
```

```
Justin-40
```

```
cut -d- -f2 score.txt
```

```
30
```

```
50
```

```
80
```

```
20
```

```
40
```

```
cut -d- -f1 score.txt
```

```
Albert
```

```
John
```

```
William
```

```
Stephen
```

```
Justin
```

```
cat 1.txt
```

```
Hello World
```

```
cat 2.txt
```

```
Computer Program
```

```
paste 1.txt 2.txt
```

```
Hello World  Computer Program
```

```
cat 1.txt
```

```
Hello World
```

```
cat 2.txt
```

```
Computer Program
```

```
join 1.txt 2.txt
```

```
Hello World  Computer Program
```

Command:

```
rev 1.txt
```

Description:

Reverse lines of a file (1.txt)

```
cat 3.txt
```

```
22
```

```
33
```

```
11
```

```
77
```

```
55
```

```
sort 3.txt
```

```
11
```

```
22
```

```
33
```

```
55
```

```
77
```

} sorts numeric values in 3.txt file and displays sorted output

```
cat 1.txt
```

```
Hello World
```

```
cat 1.txt | tr "[a-z]" "[A-Z]"
```

```
HELLO WORLD
```

} convert from lower case to upper case

```
cat 5.txt
```

```
zz
```

```
zz
```

```
yy
```

```
yy
```

```
yy
```

```
xx
```

```
uniq 5.txt
```

```
zz
```

```
yy
```

```
xx
```

} removes duplicate lines and displays unique lines

```
cat 6.txt
```

```
Einstein
```

```
Newton
```

```
Tesla
```

```
nl 6.txt
```

```
1 Einstein
```

```
2 Newton
```

```
3 Tesla
```

} numbers the lines in a file (6.txt)

Command:

```
ls -l *.txt
```

Description:

Lists the files with .txt extension

The thing with Linux is that the developers themselves are actually customers too: that has always been an important part of Linux.

Linus Torvalds


```
ls /proc/bus/
```

```
# List the contents of the /proc/bus/ directory
```

```
[manju@localhost ~]$ dmesg | grep "irq 1[45]"
```

Find irq's allocated at boot time

```
[ 2.269581] ata1: PATA max UDMA/33 cmd 0x1f0 ctl 0x3f6 bmdma 0x1060 irq 14
```

```
[ 2.269585] ata2: PATA max UDMA/33 cmd 0x170 ctl 0x376 bmdma 0x1068 irq 15
```

```
cat /proc/ioports
```

```
# List system's IO ports
```

```
echo Albert > 1.txt ; echo Einstein > 2.txt
```

```
cat 1.txt
```

```
Albert
```

```
cat 2.txt
```

```
Einstein
```

```
[manju@localhost home]$ echo $-
```

```
himBH
```

```
[manju@localhost ~]$ s=01234567890abcdefgh; echo ${s:7}
```

```
7890abcdefgh
```

```
[manju@localhost ~]$ cd /home/manju; echo $PWD
```

```
/home/manju
```

```
[manju@localhost ~]$ cd ../; pwd
```

```
/home
```

```
[manju@localhost home]$ w | cut -d " " -f 1 - | grep -v USER | sort -u
```

manju



Users currently connected

```
[manju@localhost ~]$ echo "\\\"
```

\

```
[manju@localhost ~]$ echo Al{ber,an,er}t
```

Albert Alant Alert

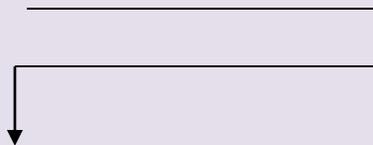
```
[manju@localhost ~]$ echo ${Albert:=Einstein}
```

Einstein

```
[manju@localhost ~]$ echo ${5*5}
```

25

```
[manju@localhost ~]$ ls
```



12.txt	allfiles.txt	echo	file3.txt	mydi	Pictures	text
13.txt	bu.txt	file	FILE.backup	mydir	Public	tree.cpio
145.txt	Desktop	file123.txt	file.md	mydir1	SHOW	users.txt
1.txt	DICT	file1.txt	first.bash	myfiles.txt	Templates	Videos
2.txt	dir	file23.txt	first.txt	myFILEs.txt.xz	test	
3.txt	Documents	file2.txt	fool.txt	newdir	testA.txt	
all	Downloads	file34.txt	Music	nohup.out	testB.txt	

```
ls -ldh * | grep -v total | \  
awk '{ print "Size is " $5 " bytes for " $9 }'
```

```
Size is 135K bytes for 12.txt           Size is 0 bytes for file34.txt  
Size is 13M bytes for 13.txt           Size is 0 bytes for file3.txt  
Size is 0 bytes for 145.txt            Size is 0 bytes for FILE.backup  
Size is 7 bytes for 1.txt               Size is 3 bytes for file.md  
Size is 9 bytes for 2.txt               Size is 13 bytes for first.bash  
Size is 8 bytes for 3.txt               Size is 13 bytes for first.txt  
Size is 20 bytes for all                Size is 66 bytes for foo1.txt  
Size is 13M bytes for allfiles.txt      Size is 6 bytes for Music  
Size is 11 bytes for bu.txt             Size is 31 bytes for mydi  
Size is 6 bytes for Desktop             Size is 6 bytes for newdir  
Size is 0 bytes for DICT                Size is 148 bytes for nohup.out  
Size is 6 bytes for dir                 Size is 6 bytes for Pictures  
Size is 6 bytes for Documents           Size is 6 bytes for Public  
Size is 6 bytes for Downloads           Size is 0 bytes for SHOW  
Size is 0 bytes for echo                 Size is 6 bytes for Templates  
Size is 0 bytes for file                 Size is 6 bytes for test  
Size is 0 bytes for file123.txt         Size is 0 bytes for testA.txt  
Size is 0 bytes for file1.txt           Size is 0 bytes for testB.txt  
Size is 0 bytes for file23.txt          Size is 25 bytes for text  
Size is 0 bytes for file2.txt           Size is 512 bytes for tree.cpio  
Size is 45 bytes for mydir               Size is 12 bytes for users.txt  
Size is 47 bytes for mydir1             Size is 6 bytes for Videos  
Size is 12 bytes for myfiles.txt  
Size is 68 bytes for myFILES.txt.xz
```

Linux	Unix
Free to use (open source)	Licensed Operating System (closed source)
Linux is just the kernel	Unix is a complete package of Operating System
Bash (Bourne Again SHell) is default shell for Linux	Bourne Shell is default shell for Unix
Portable and is booted from a USB Stick	Unportable
Source code is accessible to the general public	Source code is not accessible to anyone
Uses Graphical User Interface with an optional Command Line Interface	Uses Command Line Interface

Command:

```
echo $SHELL
```

Description:

Print the Default shell of user

Command:

```
echo $0
```

Description:

Display the name of the currently running process (**\$0 is the name of the running process**).
 If you use it inside of a shell then it will return the name of the shell. If you use it inside of a script, it will return the name of the script

Command:

```
echo *
```

Description:

Print all files and folders – similar to ls command

Command:

```
ps -p $$
```

Output:

PID	TTY	TIME	CMD
3352	pts/0	00:00:00	bash

Description:

Print the process ID of the current shell (\$\$ is the process ID of the current shell)

```
sudo du -a Documents/ | sort -n -r | head -n 5  
# List 5 biggest files from directory "Documents"
```

Command:

```
cat /etc/shells
```

Description:

List shells

Command:

```
echo m*
```

Description:

Display the files in the current folder that start with the letter "m".

Command:

```
last
```

Description:

List last logins of users and what happened such as "shutdown" or "crash" etc.

Command:

Command:

```
echo ~
```

Description:

Print your home folder path

```
bzip2 -k phy.txt
```

Description:

Compresses but does not delete the original file

phy.txt → phy.txt.bz2

Command:

```
bzip2 -d phy.txt.bz2
```

Description:

Decompresses the compressed file (phy.txt.bz2)

phy.txt.bz2 → phy.txt

Command:

```
bzcat phy.txt.bz2
```

Description:

Display the contents of compressed file (phy.txt.bz2)

Command:

```
bunzip2 phy.txt.bz2
```

Description:

Decompresses the compressed file (phy.txt.bz2)

Command:

```
crontab -l
```

Description:

Display current logged-in user's crontab entries

```
cat /dev/null > phy.txt
```



```
cp /dev/null phy.txt
```

```
echo "" > phy.txt
```

```
echo > phy.txt
```

Description:

Empty the content of a file (phy.txt)

Command:

```
nohup ping google.com &
```

Description:

Ping google.com and send the process to the background

Command:

```
nohup ping google.com > log.txt &
```

Description:

Save the ping logs to log.txt

```
pgrep -a ping
```

Output:

```
3858 ping google.com
```

```
4200 ping google.com
```

```
4236 ping google.com
```

```
kill 3858
```

```
pgrep -a ping
```

Output:

```
4200 ping google.com
```

```
4236 ping google.com
```

Command:

```
ls -la /home
```

Description:

Display the contents of /home

Command:

```
sudo shutdown 2
```

Description:

Power-off the machine after 2 minutes

Command:

```
shutdown -c
```

Description:

Cancel the shutdown process

Command:

```
pr 36.txt
```

Description:

Display the contents of the file (36.txt) one page after the other

Command:

```
stty -a
```

Description:

Display all current terminal settings

Command:

```
ls -1
```

Description:

List files one per line

Command:

```
yes John
```

Description:

Outputs a string (John) repeatedly until killed

Command:

```
vdir
```

Description:

List files and directories in the current directory (one per line) with details

Command:

```
who -b
```

Description:

Print when the system was booted

```
# Open phy.txt with nano
```

```
nano phy.txt
```

```
# Open phy.txt with vim
```

```
vim phy.txt
```

Command:

```
ls -al *.txt
```

Description:

Display all .txt files, including its individual permission.

Command:

```
uname -i
```

Description:

Display the platform of hardware

Command:

```
uname -p
```

Description:

Display the type of processor

Command:

```
cat /proc/interrupts
```

Description:

Display the interrupts

```
w --ip-addr
```

```
# Displays information regarding the users currently on the machine, login time, IDLE time,
TTY and CPU time
```

Output:

```
11:12:10 up 1:29, 2 users, load average: 0.02, 0.04, 0.10
USER      TTY      FROM    LOGIN@  IDLE   JCPU   PCPU WHAT
manju     :0       :0      02:43   ?xdm?  3:30   0.65s gdm-session-worker [pa
manju     pts/0    :0      11:01   2.00s  0.10s  0.01s w --ip-addr
```

```
w -short
```

```
# Omits CPU time and login information
```

Output:

```
11:11:46 up 1:28, 2 users, load average: 0.02, 0.04, 0.11
USER      TTY      FROM    IDLE   WHAT
manju     :0       :0      ?xdm?  gdm-session-worker [pam/gdm-password]
manju     pts/0    :0      2.00s  w --short
```

Command:

```
findmnt
```

Description:

Display a list of currently mounted file systems

Command:

```
ip addr show
```

Description:

List IP addresses and network interfaces

Command:

```
netstat -pnltn
```

Description:

List active (listening) ports

Command:

```
journalctl
```

Description:

Display systemd, kernel and journal logs

Command:


```
sudo systemctl status network
```

Description:

Display the status of network service

Command:

```
sudo systemctl start network
```

Description:

Start the network service

Command:

```
sudo systemctl stop network
```

Description:

Stop the network service

Command:

```
sestatus -b
```

Description:

Display the current state of Booleans

Command:

```
getenforce
```

Description:

Reports whether SELinux is enforcing, permissive or disabled

Security-Enhanced Linux (SELinux) is a security architecture for Linux systems that allows administrators to have more control over who can access the system

```
setenforce 0
```

```
getenforce
```

```
Output:
```

```
Permissive
```

```
setenforce 1
```

```
getenforce
```

```
Output:
```

```
Enforcing
```

- **Enforcing** - SELinux security policy is enforced.
- **Permissive** - SELinux prints warnings instead of enforcing.
- **Disabled** - No SELinux policy is loaded.

```
[manju@localhost ~]$ let a="36 + 5" ; echo $a
```

```
41
```

```
[manju@localhost ~]$ let a="20 + 50/10" ; echo $a
```

```
25
```

```
[manju@localhost ~]$ let a="20 - 50/10" ; echo $a
```

```
15
```

```
[manju@localhost ~]$ let a="20 * 50/10" ; echo $a
```

```
100
```

```
[manju@localhost ~]$ grep ^PASS /etc/login.defs
```

```
PASS_MAX_DAYS    99999
```

```
PASS_MIN_DAYS    0
```

```
PASS_MIN_LEN     5
```

```
PASS_WARN_AGE    7
```

```
[manju@localhost ~]$ grep PASS /etc/login.defs
```

```
# PASS_MAX_DAYS    Maximum number of days a password may be used.
```

```
# PASS_MIN_DAYS    Minimum number of days allowed between password changes.
```

```
# PASS_MIN_LEN     Minimum acceptable password length.
```

```
# PASS_WARN_AGE    Number of days warning given before a password expires.
```

```
PASS_MAX_DAYS    99999
```

```
PASS_MIN_DAYS    0
```

```
PASS_MIN_LEN     5
```

```
PASS_WARN_AGE    7
```

Command:

```
cut -d: -f1 /etc/passwd | column
```

Description:

List all local user accounts in column

Command:

```
mkdir ~/mydir1 ; touch ~/mydir1/myfiles1.txt
```

Description:

Create a directory "mydir1" and create a file "myfiles1.txt" in it

Command:

```
echo hi > file.md ; chmod 744 file.md
```

Description:

Create a file "file.md" and give only read access to others

```
[manju@localhost ~]$ ls -l $(which sudo)
---s--x--x. 1 root root 130776 Nov  5 2016 /bin/sudo
```

Command:

```
sestatus
```

Description:

Display the current status of the SELinux that is running on your system

Command:

```
ps -aef
```

Description:

Display full listing of processes on your system

Command:

```
sar
```

Description:

Display System Activity Report

Command:

```
ulimit
```

Description:

Report the resource limit of the current user

Output:

Unlimited

The current user can consume all the resources the current system supports

2 types of resource limitation:

- **Hard resource limit:** The physical limit that the user can reach.
- **Soft resource limit:** The limit that is manageable by the user (**its value can go up to the hard limit**)

Command:

```
ulimit -a
```

Description:

Report all the resource limits for the current user

Command:

```
ulimit -s
```

Description:

Check the maximum stack size of the current user

Command:

```
ulimit -e
```

Description:

Check out the max scheduling priority of the current user

Command:

```
ulimit -u
```

Description:

Display the maximum number of user processes

Command:

```
ulimit -v
```

Description:

Check out the size of virtual memory

Command:

```
ulimit -n
```

Description:

Check out how many file descriptors a process can have

Command:

```
man limits.conf
```

Description:

Display the in-depth information on the `limits.conf` configuration file

Command:

```
sar -V
```

Description:

Display the sar version

Command:

```
sar -u 2 5
```

Description:

Report CPU details total 5 times with the interval of 2 seconds

Command:

```
sar -n DEV 1 3 | egrep -v lo
```

Description:

Report about network interface, network speed, IPV4, TCPV4, ICMPV4 network traffic and errors

Command:

```
sar -v 1 3
```

Description:

Report details about the process, kernel thread, i-node, and the file tables

Command:

```
sar -S 1 3
```

Description:

Report statistics about swapping

Command:

```
sar -b 1 3
```

Description:

Report details about I/O operations like transaction per second, read per second, write per second

Command:

```
sudo systemctl status firewalld
```

Description:

Display the status of the firewalld

Command:

```
sudo systemctl start firewalld
```

Description:

Start the firewalld service

firewalld is a firewall management tool for Linux operating systems

Command:

```
firewall-config
```

Description:

Start the graphical firewall configuration tool

firewall-cmd

Command:

```
firewall-cmd --list-all-zones
```

Description:

List all zones

Command:

```
firewall-cmd --get-default-zone
```

Description:

Check the currently set default zone

Command:

```
firewall-cmd --list-services
```

Description:

Display currently allowed service on your system

Command:

```
firewall-cmd --list-ports
```

Description:

List the ports that are open on your system

Command:

```
firewall-cmd --zone=work --list-services
```

Description:

List services that are allowed for the public zone

Command:

```
mtr --report google.com
```

Description:

Provides information about the route that Internet traffic takes between the local system and a remote host (google.com)

Command:

```
sudo yum install samba
```

Description:

install Samba (CentOS)

Samba is client/server technology that implements network resource sharing across operating systems. With Samba, files and printers can be shared across Windows, Mac and Linux/UNIX clients.

Command:

```
sudo firewall-cmd --add-service samba --permanent
```

Description:

Add Samba service to firewalld

Command:

```
zip q.zip q.txt
```

Description:

Create a zip file (q.zip)

Command:


```
unzip q.zip
```

Description:

Unzip a zip file (q.zip)

```
zipcloak q.zip
```

```
-----
```

```
# zipcloak prompts you for a password, and then ask you to confirm it:
```

```
    Enter password:
```

```
    Verify password:
```

```
...if the passwords match, it encrypts q.zip file
```

```
-----
```

```
unzip q.zip
```

```
# When you try to unzip the q.zip file, it prompts you for the password before  
allowing you to extract the file (q.txt) it contains
```

Command:

```
zgrep -l "Einstein" *
```

Description:

Display the names of the files with the word (Einstein) present in it

Command:

```
zipsplit -n 1048576 q.zip
```

Description:

Split q.zip file to create a sequence of zipfiles (q1.zip, q2.zip.....) – each no larger than **1048576 bytes** (one megabyte)

You could concatenate (q1.zip, q2.zip....) into a new file, w.zip, with the command:

```
cat q*.zip > w.zip
```

Command:

```
mtr google.com
```

Description:

Test the route and connection quality of traffic to the destination host **google.com**

Command:

```
route
```

Description:

Display IP routing table of a Linux system

Command:

```
nmcli dev status
```

Description:

View all your network devices

Command:

```
nmcli con show
```

Description:

Check network connections on your system

Command:

```
ss -ta
```

Description:

List all TCP ports (**sockets**) that are open on a server

Command:

```
ss -to
```

Description:

Display all active TCP connections together with their timers

Command:

```
type -a alias
```

Description:

Check Bash Aliases in Linux

Difference between %Y and %y is %Y will print 4 digits while %y will print the last 2 digits of the year.

```
echo "We are in the year = $(date +%Y)"
```

```
echo "We are in the year = $(date +%y)"
```

Difference between %B and %b is, %B will print full month name while %b will print abbreviated month name.

```
echo "We are in the month = $(date +%b)"
```

```
echo "We are in the month = $(date +%B)"
```

```
# Difference between %A and %a is, %A will print full Weekday name while %a will print abbreviated weekday name.
```

```
echo "Current Day of the week = $(date +%A)"
```

```
echo "Current Day of the week = $(date +%a)"
```

```
echo "Date using %D = $(date +%D)"
```

```
echo "Date using %F = $(date +%F)"
```

```
Date using %D = 08/15/21
```

```
Date using %F = 2021-08-15
```

```
echo "current time in 24 hour format = $(date +%T)"
```

```
current time in 24 hour format = 01:27:46
```

```
echo "current time in 12 hour format = $(date +%r)"
```

```
current time in 12 hour format = 01:27:47 AM
```

```
# Print yesterday's date and time.
```

```
echo "Yesterday = $(date -d "Yesterday")"
```

```
# Print Tomorrow date and time.
```

```
echo "tomorrow = $(date -d "tomorrow")"

# Find what is the date and time before 10 days from now.
echo "Before 10 days = $(date -d "tomorrow -10 days")"

# Find last month and next month
echo "Last month = $(date -d "last month" "%B")"
echo "Next month = $(date -d "next month" "%B")"

# Find last year and next year
echo "Last Year = $(date -d "last year" "+%Y")"
echo "Next Year = $(date -d "next year" "+%Y")"
```

Command:

```
ls -lai /
```

Description:

Get the number of inodes of files in a directory (**root directory**)

Command:

```
sudo du --inode /
```

Description:

Get the total number of inodes in the root directory

Command:

```
ss -o state established '( sport = :http or sport = :https )'
```

Description:

Get the list of all clients connected to HTTP (Port 80) or HTTPS (Port 443)

Command:

```
ss -tn src :80 or src :443
```

Description:

List the numerical port numbers

Command:

```
sudo yum install putty
```

Description:

Install PuTTY on CentOS

Command:

```
sudo watch netstat -tulpn
```

Description:

Watch TCP and UDP Open Ports in Real-Time

Command:

```
sudo watch ss -tulpn
```

Description:

Watch TCP and UDP Open Ports in Real-Time

Command:

```
timeout 5s ping google.com
```

Description:

Timeout a ping command after 5 seconds

Command:

```
yum install curl
```

Description:

Install curl on CentOS

Command:

```
ss -ua
```

Description:

List all UDP Connections

Command:

```
ss -lu
```

Description:

List all Listening UDP Connections

Command:

```
ss -p
```

Description:

Display the Process IDs related to socket connections

Command:

```
ss -4
```

Description:

Display IPv4 and IPv6 Socket Connections

Command:

```
ss -6
```

Description:

Display IPv6 connections

Command:

```
ss -at '( dport = :22 or sport = :22 )'
```

Description:

Filter Connections by Port Number

"The only way to learn a new programming language is by writing programs in it."

–Dennis Ritchie

```
[manju@localhost ~]$ echo {a..z}
```

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

```
[manju@localhost ~]$ echo {z..a}
```

```
z y x w v u t s r q p o n m l k j i h g f e d c b a
```

```
[manju@localhost ~]$ echo {05..12}
```

```
05 06 07 08 09 10 11 12
```

```
[manju@localhost ~]$ echo {12..5}
```

```
12 11 10 9 8 7 6 5
```

```
[manju@localhost ~]$ echo {005..10}
```

```
005 006 007 008 009 010
```

```
mkdir 20{09..11}-{01..12}
```

```
# Create directories to group files by month and year
```

```
[manju@localhost ~]$ echo {12..05}
```

```
12 11 10 09 08 07 06 05
```

```
[manju@localhost ~]$ echo {x..z}{1..3}
```

```
x1 x2 x3 y1 y2 y3 z1 z2 z3
```

```
[manju@localhost ~]$ echo {0..10..2}
```

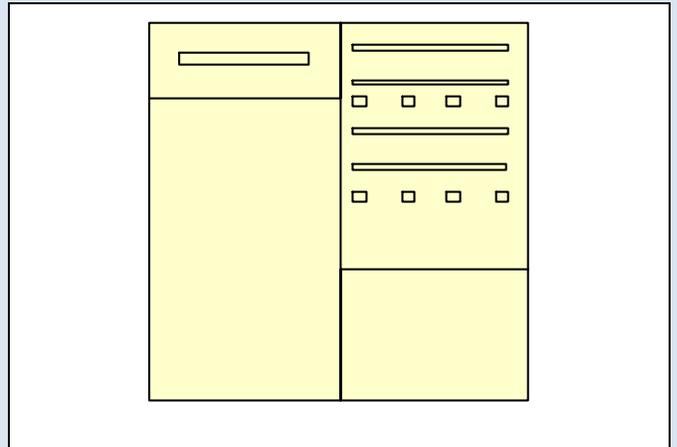
```
0 2 4 6 8 10
```

```
[manju@localhost ~]$ for i in {a..z..5}; do echo -n $i; done
```

```
afkpuz
```

```
[manju@localhost ~]$ ls *.txt; echo $_  
12.txt 1.txt 2.txt abc.txt my.txt phy.txt  
13.txt 24.txt 3.txt marks.txt names.txt mphy.txt
```

```
[manju@localhost ~]$ cut -d, -f2,1 <<<'Albert,Bob,John'  
Albert,Bob  
[manju@localhost ~]$ cut -d, -f2,2 <<<'Albert,Bob,John'  
Bob  
[manju@localhost ~]$ cut -d, -f2,3 <<<'Albert,Bob,John'  
Bob,John
```



```
[manju@localhost ~]$ x="W X Y Z"; echo "$x"  
W X Y Z  
[manju@localhost ~]$ x="W X Y Z"; echo $x  
W X Y Z
```

echo \$x and echo "\$x" yield different results

Quoting a variable preserves whitespace

```
[manju@localhost ~]$ let x=20+7; echo "The value of \"x\" is $x."
```

The value of "x" is 27.

```
[manju@localhost ~]$ x=100; let "x += 1"; echo "x = $x"
```

x = 101

```
[manju@localhost ~]$ x="a+b+c"; IFS=+; echo $x
```

a b c

The "+" sign will be interpreted as a separator

```
[manju@localhost ~]$ x="a-b-c"; IFS=-; echo $x
```

a b c

The "-" sign will be interpreted as a separator

```
[manju@localhost ~]$ x="a,b,c"; IFS=,; echo $x
```

a b c

The "comma" will be interpreted as a separator

```
free | grep Mem | awk '{ print $4 }'
```

Display the unused RAM memory

```
du -ach
```

Display (disk) file usage

```
readelf -h /bin/bash
```

Display information and statistics about a designated elf binary

```
[manju@localhost ~]$ expr 5 \* 2 + 3
```

13 # 10 + 3

```
[manju@localhost ~]$ expr 5 \* \( 2 + 3 \)
```

25 # 5 * 5

```
[manju@localhost ~]$ echo -e "\033[4mAlbert Einstein.\033[0m"
```

Albert Einstein.

```
[manju@localhost ~]$ echo -e "\033[1mAlbert Einstein.\033[0m"
```

Albert Einstein.

```
[manju@localhost ~]$ echo -e '\E[34;47mAlbert Einstein'; tput sgr0
```

Albert Einstein

```
[manju@localhost ~]$ echo -e '\E[33;44m"Albert Einstein"; tput sgr0
```

Albert Einstein

```
[manju@localhost ~]$ echo -e '\E[1;33;44m"Albert Einstein"; tput sgr0
```

Albert Einstein

```
[manju@localhost ~]$ x=2; y=3; echo $((2*$x + 3*$y))
```

13

```
[manju@localhost ~]$ x=2; y=3; echo $((2*x + 3*y))
```

13

```
[manju@localhost ~]$ let x=2+3 y=3+2; echo $x $y
```

5 5

Command:

```
sdiff phy.txt score.txt
```

Description:

Show Difference between Two Files (**phy.txt** and **score.txt**)

Command:

```
history -c
```

Description:

Delete or clear all the entries from bash history

Command:

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

Description:

The ping test will stop after sending 5 packets

```
# count number of lines in each .txt file
```

```
ls *.txt | xargs wc -l
```

```
# count number of words in each .txt file
```

```
ls *.txt | xargs wc -w
```

```
# count number of characters in each .txt file
```

```
ls *.txt | xargs wc -c
```

```
# count lines, words and characters in each .txt file
```

```
ls *.txt | xargs wc
```

Command:

```
lslogins -u
```

Description:

Displays user accounts

Command:

```
systemctl list-units --type=service
```

Description:

List all loaded services on your system (whether active; running, exited or failed)

Command:

```
systemctl --type=service
```

Description:

List all loaded services on your system (whether active; running, exited or failed)

Command:

```
systemctl list-units --type=service --state=active
```

Description:

List all loaded but active services

Command:

```
systemctl --type=service --state=active
```

Description:

List all loaded but active services

Command:

```
systemctl list-units --type=service --state=running
```

Description:

List all running services (i.e., all loaded and actively running services)

Command:

```
systemctl --type=service --state=running
```

Description:

List all running services (i.e., all loaded and actively running services)

#scan a single port

```
nc -v -w 2 z 192.168.56.1 22
```

scan multiple ports

```
nc -v -w 2 z 192.168.56.1 22 80
```

scan range of ports

```
nc -v -w 2 z 192.168.56.1 20-25
```

Command:

```
cat /etc/resolv.conf
```

Description:

Find out your DNS Server IP address

Command:

```
less /etc/resolv.conf
```

Description:

Find out your DNS Server IP address

Command:

```
findmnt --poll --mountpoint /mnt/test
```

Description:

Monitor mount, unmount, remount and move actions on a directory (i.e., on `/mnt/test`)

Command:

```
uptime -p
```

Description:

Check Linux Server Uptime

Command:

```
uptime -s
```

Description:

Check Linux Server Starting Time

Command:

```
uptime -h
```

Description:

Display uptime's version information

Command:

```
grep -o -i Justin score.txt | wc -l
```

Description:

Count the number of times "**Justin**" appears in the file (**score.txt**)

Command:

```
crontab -r
```

Description:

Delete all crontab jobs

```
ADD=$(( 1 + 2 ))
```

```
echo $ADD
```

3

```
MUL=$(( $ADD * 5 ))
```

```
echo $MUL
```

15

```
SUB=$(( $MUL - 5 ))
```

```
echo $SUB
```

10

```
DIV=$(( $SUB / 2 ))
```

```
echo $DIV
```

5

```
MOD=$(( $DIV % 2 ))
```

```
echo $MOD
```

1

Command:

```
expr length "This is myw3schools.com"
```

Description:

Find the length of a string (**This is myw3schools.com**)

```
echo '3+5' | bc
```

8

```
awk 'BEGIN { a = 6; b = 2; print "(a + b) = ", (a + b) }'
```

(a + b) = 8

Command:

```
factor 10
```

Description:

Decompose an integer (**10**) into prime factors

Command:

```
ps -e
```

Description:

Display every active process on a Linux system

Command:

```
ps -x
```

Description:

Display User Running Processes

Command:

```
ps -fU manju
```

Description:

Display a user's processes by user name (**manju**)

Command:

```
ps -fu 1000
```

Description:

Display a user's processes by real user ID (**RUID**)

Command:

```
ps -U root -u root
```

Description:

Display every process running with root user privileges (real and effective ID)

```
echo -e "The following users are logged on the system:\n\n $(who) "
```

```
manju      :0          Aug 15 03:31 (:0)
manju      pts/1        Aug 15 03:32 (:0)
```

Command:

```
sh <(curl https://nixos.org/nix/install) --daemon
```

Description:

Install Nix Package Manager in Linux

Command:

```
locale
```

Description:

View System Locale in Linux

Command:

```
locale -a
```

Description:

Display a list of all available locales

```
cat score.txt
```

```
Justin-40
```

```
cat score.txt | tr [:lower:] [:upper:]
```

```
JUSTIN-40
```

```
cat score.txt | tr [a-z] [A-Z] >output.txt  
cat output.txt
```

```
JUSTIN-40
```

```
cat domainnames.txt
```

```
www. google. com  
www. fb. com  
www. mactech. com
```

```
cat domainnames.txt | tr -d ' '
```

```
www.google.com  
www.fb.com  
www.mactech.com
```

Remove the
spaces in the
domain names

```
cat domainnames.txt
```

```
www.google....com  
www.fb.com  
www.mactech.Com
```

```
cat domainnames.txt | tr -s ' '
```

```
www.google.com  
www.fb.com  
www.mactech.Com
```

```
echo "My UID is $UID"
```

```
My UID is 0
```

```
echo "My UID is $UID" | tr " " "\n"
```

```
My  
UID  
is  
0
```

A space into a ":" character

```
echo "myw3schools.com =>Linux-Books,Src,Tutorials" | tr " " ":"
```

```
myw3schools.com:=>Linux-Books,Src,Tutorials
```

Command:

```
!sud
```

Description:

Re-execute previously used command

Command:

```
!sudo
```

Description:

Re-execute previously used command

Command:

```
cut -d: -f1 < /etc/passwd | sort | xargs
```

Description:

Generate a compact list of all Linux user accounts on the system

Command:

```
zcat phy.txt.gz myfiles.txt.gz
```

Description:

View multiple compressed files (**phy.txt.gz** and **myfiles.txt.gz**)

Command:

```
find . -type f -name "*.php"
```

Description:

Find all php files in a directory

```
mkdir /tmp/DOCUMENTS ←  
  
# Create a directory 'DOCUMENTS' under "/tmp" directory
```

Command:


```
find . -type f -perm 0777 -print
```

Description:

Find all the files whose permissions are *777*

Command:

```
find / -type f ! -perm 777
```

Description:

Find all the files without permission *777*

Command:

```
find / -perm /g=s
```

Description:

Find all SGID set files

Command:

```
find / -perm /u=r
```

Description:

Find all Read-Only files

Command:

```
find / -perm /a=x
```

Description:

Find all Executable files

```
[manju@localhost ~]$ echo "ALBERT" | awk '{print tolower($0)}'
```

```
albert
```

Convert text from upper case to lower case

Command:

```
find . -type f -name "phy.txt" -exec rm -f {} \;
```

Description:

Find and remove **phy.txt** File

Command:

```
[manju@localhost ~]$ echo "Phone number: 55602369" | tr -cd [:digit:]  
55602369
```

Get the digits from string

```
find . -type f -name "*.txt" -exec rm -f {} \;
```

Description:

To find and remove multiple **.txt** files

Command:

```
find . -type f -name "*.mp3" -exec rm -f {} \;
```

Description:

To find and remove multiple **.mp3** files

Command:

```
find /tmp -type d -empty
```

Description:

Find all Empty Directories

Command:

```
find /tmp -type f -name ".*"
```

Description:

File all Hidden Files

```
[manju@localhost ~]$ echo "Phone number: 55602369" | tr -d [:digit:]
```

Phone number:

Remove all digits from string

Command:

```
find / -mtime 50
```

Description:

Find Last 50 Days Modified Files

Command:

```
find / -atime 50
```

Description:

Find Last 50 Days Accessed Files

Command:

```
find / -cmin -60
```

Description:

Find Changed Files in Last 1 Hour

Command:

```
find / -mmin -60
```

Description:

Find Modified Files in Last 1 Hour

Command:

```
find / -amin -60
```

Description:

Find Accessed Files in Last 1 Hour

Command:

Command:

```
type cat
```

Description:

Identifies whether the "cat" command is a shell built-in command, subroutine, alias, or keyword.

```
find / -size 50M
```

Description:

Find all 50MB files

Command:

```
find / -type f -size +100M -exec rm -f {} \;
```

Description:

Find and Delete 100MB Files

Command:

```
find / -type f -name *.mp3 -size +10M -exec rm {} \;
```

Description:

Find all .mp3 files with more than 10MB and delete them

```
ls -l --color
```

```
# List the files in current directory (with colorized output)
```

```
info df
```

```
# Loads the "df"info page
```

```
ls /usr/include
```

```
# List the Header files for compiling C programs
```

```
ls /usr/local
```

```
# List the Locally installed files
```

```
ls /usr/bin/d*
```

```
# List all files whose names begin with the letter "d" in the /usr/bin directory
```

```
[manju@localhost ~]$ ls .b*
```

```
.bash_history .bash_logout .bash_profile .bashrc
```

```
[manju@localhost ~]$ ls [a-h]*
```

```
all          DICT  file1      file2      file34.txt  file.md    fool.txt
allfiles.txt echo   file123.txt file23.txt  file3.txt   first.bash
bu.txt       file  file1.txt  file2.txt  FILE.backup first.txt
```

```
[manju@localhost ~]$ touch hello.cpp; touch hello.f99
```

```
[manju@localhost ~]$ ls *.*[9p]?
```

```
hello.cpp  hello.f99
```



```
ls /usr
```

```
# List the /usr directory
```

```
ls ~ /usr
```

```
# List the user's home directory and the /usr directory
```

```
[manju@localhost ~]$ echo f*
```

Display any file beginning with "f"

```
file file1 file123.txt file1.txt file2 file23.txt file2.txt file34.txt file3.txt  
file.md first.bash first.txt fool.txt
```

```
[manju@localhost ~]$ echo f*.txt
```

Display any file beginning with "f" followed by
any characters and ending with ".txt"

```
file123.txt file1.txt file23.txt file2.txt file34.txt file3.txt first.txt fool.txt
```

```
sudo vim myfiles.txt
```

```
# Open a file "myfiles.txt" using Vim editor
```

```
[manju@localhost ~]$ for ((i=0;i<8;i++)); do echo $((i)); done
```

```
0
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

Command:

```
cat /proc/sys/fs/file-max
```

Description:

Find Linux Open File Limit

Command:

```
ulimit -Hn
```

Description:

Check Hard Limit in Linux

Command:

```
ulimit -Sn
```

Description:

Check Soft Limits in Linux

Command:

```
timedatectl status
```

Description:

Display the current time and date on your system

Command:

```
timedatectl list-timezones
```

Description:

View all available timezones

Command:

```
timedatectl list-timezones | egrep -o "Asia/B.*"  
timedatectl list-timezones | egrep -o "Europe/L.*"  
timedatectl list-timezones | egrep -o "America/N.*"
```

Description:

Find the local timezone according to your location

Command:

```
timedatectl set-timezone "Asia/Kolkata"
```

Description:

Set your local timezone in Linux

Command:

```
swapon --summary
```

Description:

View a summary of swap space usage by device

Command:

```
cat /proc/swaps
```

Description:

Check swap usage information

```
# start recording of Linux terminal
```

```
script history_log.txt
```

```
Script started, file is history_log.txt
```

```
exit
```

```
Script done, file is history_log.txt
```

Command:

```
dir -shl
```

Description:

List files and their allocated sizes in blocks

Command:

```
less /proc/sys/dev/cdrom/info
```

Description:

Display information about CD-ROM

```
while true; do date >> date.txt ; sleep 5 ; done &  
  
cat date.txt
```

```
Mon Aug 16 03:05:36 PDT 2021  
Mon Aug 16 03:05:41 PDT 2021  
Mon Aug 16 03:05:46 PDT 2021  
Mon Aug 16 03:05:51 PDT 2021
```

"Don't write better error messages, write code that doesn't need them."

– Jason C. McDonald

```
[manju@localhost ~]$ echo hello > 1.txt
```

```
[manju@localhost ~]$ echo world > 2.txt
```

```
[manju@localhost ~]$ echo program > 3.txt
```

```
[manju@localhost ~]$ cat 1.txt
```

```
hello
```

```
[manju@localhost ~]$ cat 2.txt
```

```
world
```

```
[manju@localhost ~]$ cat 3.txt
```

```
program
```

```
[manju@localhost ~]$ cat 1.txt 2.txt 3.txt
```

```
hello
```

```
world
```

```
program
```

```
[manju@localhost ~]$ cat 1.txt 2.txt 3.txt >all
```

```
[manju@localhost ~]$ cat all
```

```
hello
```

```
world
```

```
program
```

```
strings /usr/bin/passwd
```

```
# Display the readable character strings from the /usr/bin/passwd
```

```
ls -lrS /etc
```

```
# List the biggest file in /etc
```

```
cat /etc/passwd >> myfiles.txt
```

```
# Create a file named myfiles.txt that contains the contents of myfiles.txt followed by the contents of /etc/passwd
```

```
[manju@localhost ~]$ ls /etc/*.conf
```

```
/etc/asound.conf           /etc/kdump.conf           /etc/radvd.conf
/etc/autofs.conf           /etc/krb5.conf            /etc/request-key.conf
/etc/autofs_ldap_auth.conf /etc/ksmtuned.conf        /etc/resolv.conf
/etc/brltty.conf           /etc/ld.so.conf           /etc/rsyncd.conf
/etc/cgconfig.conf         /etc/libaudit.conf        /etc/rsyslog.conf
/etc/cgrules.conf          /etc/libuser.conf         /etc/sestatus.conf
/etc/cgsnapshot_blacklist.conf /etc/locale.conf         /etc/sos.conf
/etc/chrony.conf           /etc/logrotate.conf       /etc/sudo.conf
/etc/dleyna-server-service.conf /etc/man_db.conf         /etc/sudo-ldap.conf
/etc/dnsmasq.conf          /etc/mke2fs.conf          /etc/sysctl.conf
/etc/dracut.conf           /etc/mtools.conf          /etc/tcsd.conf
/etc/e2fsck.conf           /etc/nfsmount.conf        /etc/updatedb.conf
/etc/fprintd.conf          /etc/nsswitch.conf        /etc/usb_modeswitch.conf
/etc/fuse.conf             /etc/ntp.conf             /etc/vconsole.conf
/etc/GeoIP.conf            /etc/numad.conf           /etc/wvdial.conf
/etc/host.conf             /etc/oddjobd.conf         /etc/yum.conf
/etc/idmapd.conf           /etc/pbm2ppa.conf
/etc/ipsec.conf            /etc/pnm2ppa.conf
```

Display configuration files located in /etc

```
ls /dev/sd*
```

```
/dev/sda /dev/sda1 /dev/sda2 /dev/sda3
```

Display SATA device files


```
echo \$USER
```

```
# $USER
```

```
echo -e "2+2\t=4" ; echo -e "12+12\t=24"
```

```
2+2 =4  
12+12 =24
```

```
echo Hello && echo World
```

```
Hello
```

```
World
```

```
echo Hello ; echo World
```

```
Hello
```

```
World
```

```
echo Hello || echo Hi ; echo World
```

```
Hello
```

```
World
```

```
rm myfiles.txt && echo It worked! || echo It failed!
```

```
It worked!
```

```
rm files.txt && echo It worked! || echo It failed!
```

```
rm: cannot remove 'files.txt': No such file or directory
```

```
It failed!
```

```
pwd ; pwd
```

```
/home/manju
```

```
/home/manju
```

```
Execute the pwd command twice
```

```
a=$(pwd)
echo "Current working directory is : $a"
```

`/home/manju`

Command:

```
echo *.jpeg
```

Description:

Print all **.jpeg** files

Command:

```
echo 'linux' | fold -w1
```

Description:

Break down a word (**linux**) into individual

```
l  
i  
n  
u  
x
```

Command:

```
find . -user root
```

Description:

Output the files with respect of the user (**root**) owned files in the current directory

Command:

```
strace pwd
```

Description:

Trace a command (**pwd**) execution

Command:

```
top -u manju
```

Description:

Display specific User (**manju**) process details

3 characteristics of big data:

- **Volume** — How much data is there?
- **Variety** — How diverse is different types of data?
- **Velocity** — At what speed is new data generated?

```
[manju@localhost ~]$ netstat -plunt
```

```
# print all listening ports
```

```
[manju@localhost ~]$ netstat -plunt | grep 8080
```

```
# check if server is listening on port 8080 or not
```

```
[manju@localhost ~]$ netstat -s
```

```
# list statistics of all ports
```

```
[manju@localhost ~]$ cat myfiles.txt
```

```
Hello World
```

```
[manju@localhost ~]$ cat myfiles.txt | tr ' ' '\n'
```

```
Hello
```

```
World
```

```
find . -name "*.txt"
```

```
# Find files that end in .txt in the current directory and all subdirectories
```

```
find /etc > 12.txt
```

```
# Find all files in /etc and place the list in 12.txt
```

```
find . -newer file1.txt
```

```
# Find files that is newer than file1.txt
```

```
[manju@localhost ~]$ date +%A %d-%m-%Y'
```

```
Tuesday 19-04-2022
```

```
[manju@localhost ~]$ date -d '2022-04-01 + 2000000000 seconds'
```

```
Thu Aug 16 03:33:20 PDT 2085
```

```
find /etc -type f -name '*.txt' | wc -l
```

```
# Print the number of .txt files in /etc and all its subdirectories
```

```
[manju@localhost ~]$ cat myfiles.txt
```

```
Hello World
```

```
[manju@localhost ~]$ grep -E 'o*' myfiles.txt
```

```
Hello World
```

```
[manju@localhost ~]$ grep -E 'o+' myfiles.txt
```

```
Hello World
```



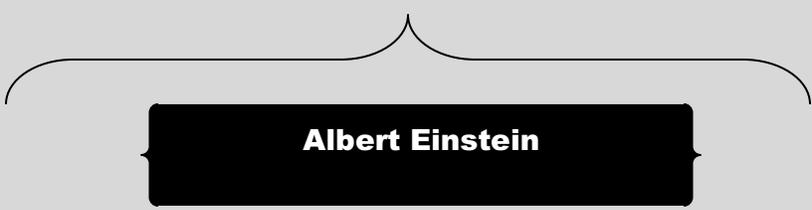
```
[manju@localhost ~]$ cat myfiles.txt
```

```
Hello World
```

```
[manju@localhost ~]$ cat myfiles.txt | sed 's/1\{2\}/A/'
```

```
HeAo World
```

```
echo Albert `echo -n Einstein`
```



Albert Einstein

```
[manju@localhost ~]$ test 50 -gt 15 ; echo $?
```

```
0  
True: 50 is greater than 15
```

```
[manju@localhost ~]$ test 5 -gt 15 ; echo $?
```

```
1  
False: 5 is not greater than 15
```

```
[manju@localhost ~]$ test 5 -lt 15 ; echo $?
```

```
0  
True: 5 is lesser than 15
```

```
[manju@localhost ~]$ test 50 -gt 15 && echo true || echo false
```

```
true
```

```
[manju@localhost ~]$ test 5 -gt 15 && echo true || echo false
```

```
false
```

```
[manju@localhost ~]$ [ 50 -gt 15 ] && echo true || echo false
```

```
true
```

```
[manju@localhost ~]$ [ 5 -gt 15 ] && echo true || echo false
```

```
false
```

```
[manju@localhost ~]$ [ 100 -gt 10 -a 100 -lt 150 ] && echo true || echo false
```

```
true
```

```
[manju@localhost ~]$ [ 100 -gt 10 -a 100 -lt 15 ] && echo true || echo false
```

```
false
```

```
[manju@localhost ~]$ a=2; b=a; eval c=\$$b; echo $c
```

```
2
```

```
[manju@localhost ~]$ date
```

```
Tue Apr 19 02:55:39 PDT 2022
```

```
[manju@localhost ~]$ date --date="1 week ago"
```

```
Tue Apr 12 02:55:05 PDT 2022
```


Command:

```
uname -or
```

Description:

Find Out Linux Kernel Version

Command:

```
uname -a
```

Description:

Print linux system information

Command:

```
cat /proc/version
```

Description:

Display some of your system information including the Linux kernel version

Command:

```
cat /etc/centos-release
```

Description:

Find Out Linux Distribution Name and Release Version

Command:

```
fuser .
```

Description:

Displays the PIDs of processes currently accessing your current working directory

Command:

```
fuser -v -m .bashrc
```

Description:

Determine which processes are accessing your **~.bashrc file**

Command:

```
sudo fuser --list-signals
```

Description:

Displays all the possible signals that can be used with the fuser tool

Command:

```
sudo fuser -k -HUP /boot
```

Description:

Sends the HUP signal to all processes that have your **/boot directory** open

Command:

```
ls -al
```

Description:

List all the files with the file permissions, the number of links to that file, the owner of the file, the group of the file, the file size in bytes, the file's last modified datetime and the file name

Command:

```
echo "shutdown -h now" | at -m 23:55
```

Description:

Shutdown the system at 23:55 today

Command:

```
echo "updatedb" | at -m 23.55
```

Everyone can now read the file

```
chmod a+r myfiles.txt
```

Everyone can now read and write the file

```
chmod a+rw myfiles.txt
```

Others (not the owner, not in the same group of the file) cannot read, write or execute the file

```
chmod o-rwx myfiles.txt
```

Description:

Creates and updates the database of file names used by locate

Run **updatedb** at 23:55 today



Command:

```
echo $(ls -al)
```

Description:

Execute command "ls -al" and print the result to the standard output

Command:

```
top -b -o +%MEM | head -n 22
```

Description:

Display the top 15 processes sorted by memory use in descending order

Command:

```
top -b -o +%MEM | head -n 22 > report.txt
```

Description:

Redirect the output to a file (**report.txt**) for later inspection

Command:

```
ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem | head
```

Description:

Check Top Processes sorted by RAM or CPU Usage in Linux

Command:

```
find . -type f \( -name "*.sh" -o -name "*.txt" \)
```

Description:

Find all files in the current directory with **.sh** and **.txt file** extensions

Command:

```
find . -type f \( -name "*.sh" -o -name "*.txt" -o -name "*.c" \)
```

Description:

Find all files in the current directory with **.sh**, **.c** and **.txt file** extensions

Description:

Find files edited more than 3 days ago.

Command:

```
find . -type f -mtime +3
```

Description:

Find files edited in the last 24 hours.

Command:

```
find . -type f -mtime -1
```

Description:

Find files that have more than 100 characters (bytes) in them.

Command:

```
find . -type f -size +100c
```

Description:

Find files bigger than 100 KB but smaller than 1 MB.

Command:

```
find . -type f -size +100k -size -1M
```

Description:

Deletes all the files edited in the last 24 hours.

Command:

```
find . -type f -mtime -1 -delete
```

Description:

List all files including hidden files.

Command:


```
ls -a
```

Description:

List Files and Directories with "/" Character at the End.

Command:

```
ls -F
```

Description:

List Files in Reverse Order.

Command:

```
ls -r
```

Description:

Sort Files by File Size.

Command:

```
ls -lS
```

Description:

List Files with an inode number.

Command:

```
ls -i
```

Description:

Check the version of the ls command.

Command:

```
ls --version
```

Description:

List files under directory /tmp.

Command:

```
ls -l /tmp
```

Description:

Display UID and GID of files and directories.

Command:

```
ls -n
```

Description:

Find all 30 MB files.

Command:

```
find / -size 30M
```

Description:

Find files with sizes between 100 - 200MB.

Command:

```
find / -size +100M -size -200M
```

Description:

List directories larger than 20 KB.

Command:

```
find / -type d -size +20k
```

Description:

Find empty files and directories.

Command:

```
find ./ -type f -size 0
```

Description:

List files modified within the last 17 hours.

Command:

```
find . -mtime -17 -type f
```

Description:

*** List directories modified within the last 10 days.***

Command:

```
find . -mtime -10 -type d
```

Description:

List all files modified between 6 and 15 days ago in the home directory.

Command:

```
find /home -type f -mtime +6 -mtime -15
```

Description:

Display files with permission 777.

Command:

```
find -perm 777
```

Description:

List files owned by a user (manju).

Command:

```
find /home -user manju
```

Description:

Find all text files owned by user "manju".

Command:

```
find /home -user manju -iname "*.txt"
```

Description:

Find and list files and directories together with their permissions.

Command:

```
find -name "*.conf" | ls -l
```

Description:

List directories only.

Command:

```
ls -d */
```

Description:

List multiple files on a single line.

Command:

```
ls --format=comma
```

Description:

View the process of a specific user "manju".

Command:

```
ps -u manju
```

Description:

Execute a previous command starting with a specific letter "c".

Command:

```
!c
```

Description:

Display BIOS information (You need elevated permissions to run this).

Command:

```
dmidecode -t 0
```

Description:

Display CPU information (You need elevated permissions to run this).

Command:

```
dmidecode -t 4
```

Description:

View all the system logs.

Command:

```
gnome-system-log
```

Description:

Identify SSH Client Version.

Command:

```
ssh -V
```

Description:

Display total connect time of users.

Command:

```
ac -d
```

Description:

Display connect time for all the users.

Command:

```
ac -p
```

Description:

Diaplay connect time report for a specific user "manju".

Command:

```
ac -d manju
```

Description:

Display the modules compiled inside Apache.

Command:

```
httpd -l
```

Description:

*** View Processes Owned by Current User.***

Command:

```
ps U $USER
```

Description:

Display the information about the filesystem Type.

Command:

```
df -Tha
```

Description:

Display Active Connections with Process ID and Program Name.

Command:

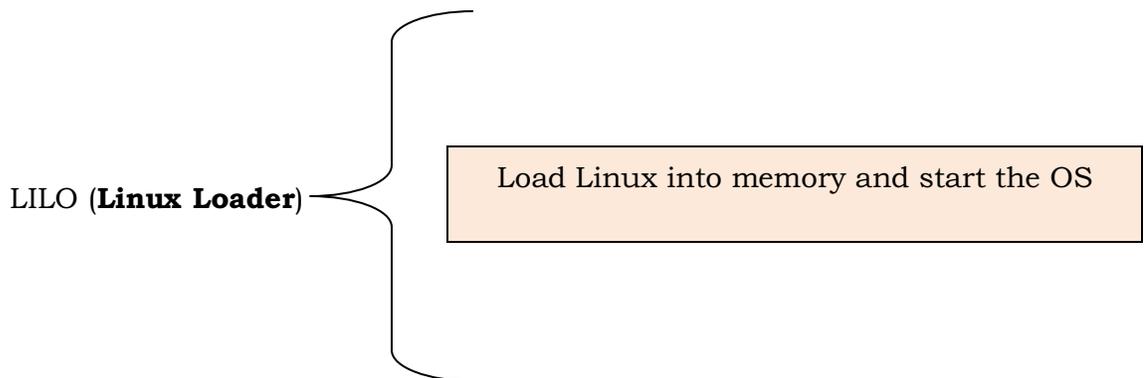
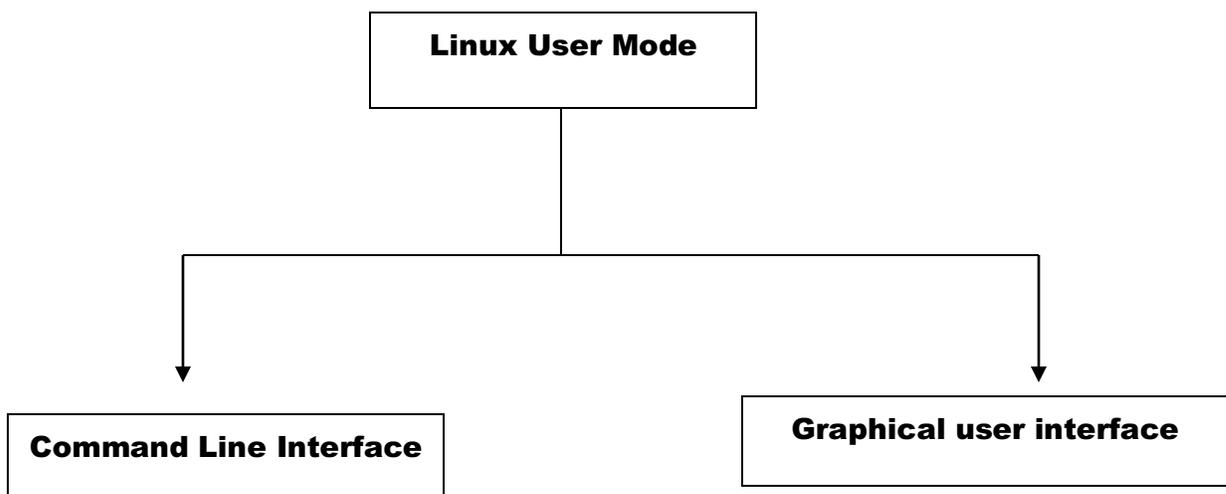
```
netstat -tap
```

Description:

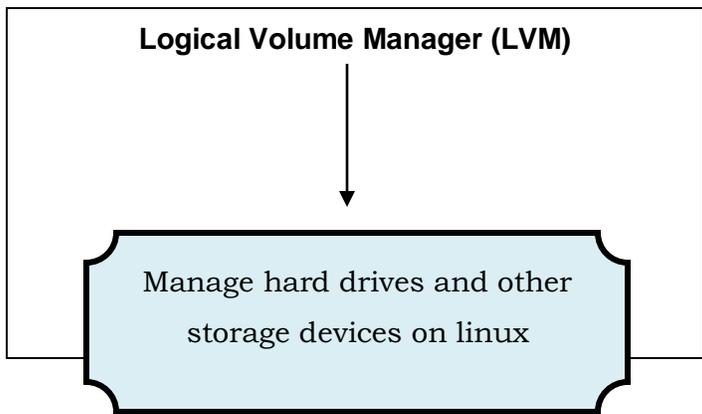
Display RAW network statistics.

Command:

```
netstat --statistics --raw
```



The maximum length for a filename under Linux is 255 bytes.



```
[manju@localhost ~]$ PS1="Please enter a command: "
```

```
Please enter a command: date
```

```
Thu Apr 21 20:51:19 PDT 2022
```

```
Please enter a command: cal
```

```
April 2022
```

```
Su Mo Tu We Th Fr Sa
```

```
1 2
```

```
3 4 5 6 7 8 9
```

```
10 11 12 13 14 15 16
```

```
17 18 19 20 21 22 23
```

```
24 25 26 27 28 29 30
```

```
Please enter a command:
```

```
ps -aux | grep 'httpd'  
# Check for the httpd process
```

```
[manju@localhost ~]$ ls /var/spool
```

```
/var/spool holds spooled files such as those  
generated for printing jobs and network transfers
```

```
abrt abrt-upload anacron at cron cups lpd mail plymouth postfix
```

```
[manju@localhost ~]$ ls /usr/share/man
```

```
/usr/share/man holds the online Man files
```

```
ca en hu ko man1x man3p man5 man7 man9 pl ro tr zh_TW
```

```
cs es id man0p man2 man3x man5x man7x man9x pt ru uk
```

```
da fr it man1 man2x man4 man6 man8 mann pt_BR sk zh
```

```
de hr ja man1p man3 man4x man6x man8x overrides pt_PT sv zh_CN
```

```
[manju@localhost ~]$ ls /etc/gdm
```

List the contents of GDM configuration directory

```
custom.conf  Init  PostLogin  PostSession  PreSession  Xsession
```

```
ls /etc/gconf
```

```
# List the GConf configuration files
```

```
ls /usr/share/gnome
```

```
# List the files used by GNOME applications
```

```
[manju@localhost ~]$ ls /etc/sysconfig
```

List the **system configuration** files

```
atd          firewallld    libvirt-guests  qemu-ga        samba
authconfig   grub          man-db          radvd          saslauthd
autofs       init          modules         raid-check     selinux
cbq          ip6tables-config netconsole      rdisc          smartmontools
cgred        iptables-config network         readonly-root  sshd
console      irqbalance   network-scripts rpcbind        sysstat
cpupower     kdump        nfs             rpc-rquotad    sysstat.ioconf
crond        kernel       ntpd            rsyncd         virtlockd
ebtables-config ksm          ntpdate        rsyslog        virtlogd
fcoe         libvirtd     pluto          run-parts      wpa_supplicant
```

```
ls /etc/rc.d
```

```
# List the system startup and shutdown files
```

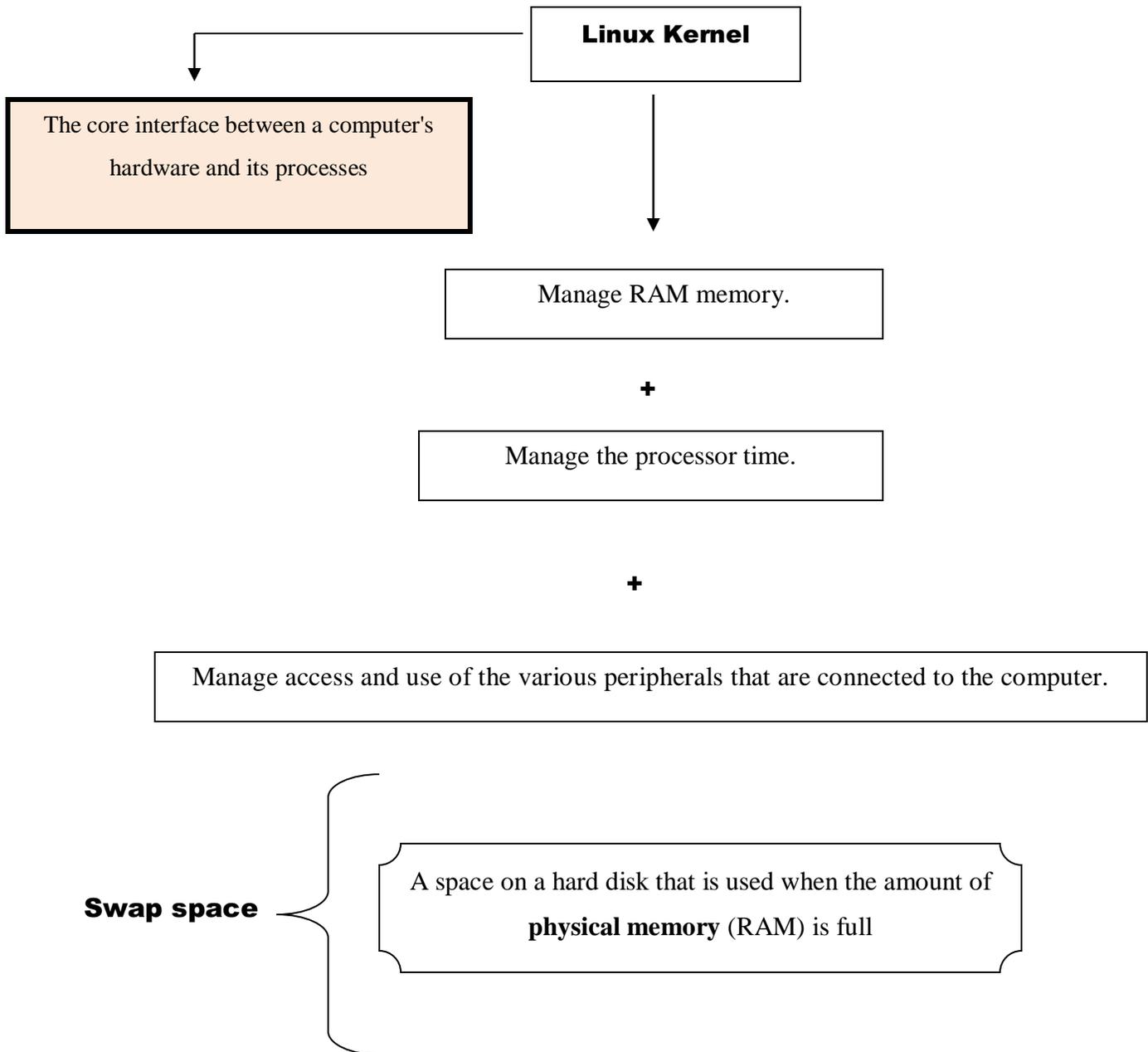
```
[manju@localhost ~]$ ls /etc/init.d
```

/etc/init.d holds network scripts to start up
network connections

```
functions  netconsole  network  README
```

Important features of Linux Operating System

- Free and Open Source
- Portable and More secure
- Robust and Adaptable




```
[manju@localhost ~]$ cd /etc
```

```
[manju@localhost etc]$ pwd
```

```
/etc
```

```
[manju@localhost etc]$ cat /etc/hosts
```

/etc/hosts contains hostnames with their ip address

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1        localhost localhost.localdomain localhost6 localhost6.localdomain6
```

```
chmod u+w myfiles.txt
```

```
# Add user write privileges
```

```
chmod u-w myfiles.txt
```

```
# Remove user write privileges
```

```
chmod g+w myfiles.txt
```

```
# Add group write privileges
```

```
chmod g=r myfiles.txt
```

```
# Allow only the group read privileges
```

```
chmod o+x myfiles.txt
```

```
# Add execute privileges for others
```

```
chmod a+x myfiles.txt
```

```
# Add execute privileges for everyone
```

```
chmod a=rx myfiles.txt
```

```
# Allow read and execute only to everyone
```

```
chmod go-r myfiles.txt
```

```
# Remove group and others read privileges
```

```
ps -L 3315
```

```
# List all threads for a particular process (with process ID 3315)
```

```
ps aux --sort pmem
```

```
# Check the memory status
```

```
awk '/Hello/' myfiles.txt
```

```
# Find "Hello" in myfiles.txt
```

```
awk -F: '{ print $1 }' /etc/passwd | sort
```

```
# Display a sorted list of the login names of all users
```

```
awk 'END { print NR }' myfiles.txt
```

```
# Counts lines in myfiles.txt
```

```
[manju@localhost ~]$ awk 'BEGIN { for (i = 1; i <= 7; i++) print int(101 * rand()) }'
```

```
24
```

```
29
```

```
85
```

```
15
```

```
59
```

```
19
```

```
81
```

Prints **seven random numbers** from zero to 100

```
ls -lg *.txt | awk '{ x += $5 } ; END {print "total bytes:" x }'
```

```
# Prints the total number of bytes used by all .txt files
```

Random-access memory	Virtual memory
The internal memory of the CPU for storing data, program and program result.	A storage area that holds the files on your hard drive for retrieval when a computer runs out of RAM

Process States in Linux:

- **Ready:** a new process is created and is ready to run.
- **Running:** The process is being executed.
- **Wait:** The process is waiting for input from the user.
- **Completed:** The process has completed the execution.
- **Zombie:** The process is terminated but information regarding the process still exists and is available in the process table.

Cron	Anacron
A service that enables us to run scheduled jobs in Linux/Unix systems every minute.	A service that only enables us to run scheduled jobs in Linux/Unix systems on daily basis.

Command:

```
cat /etc/crontab
```

Description:

View system defined cron jobs

Command:

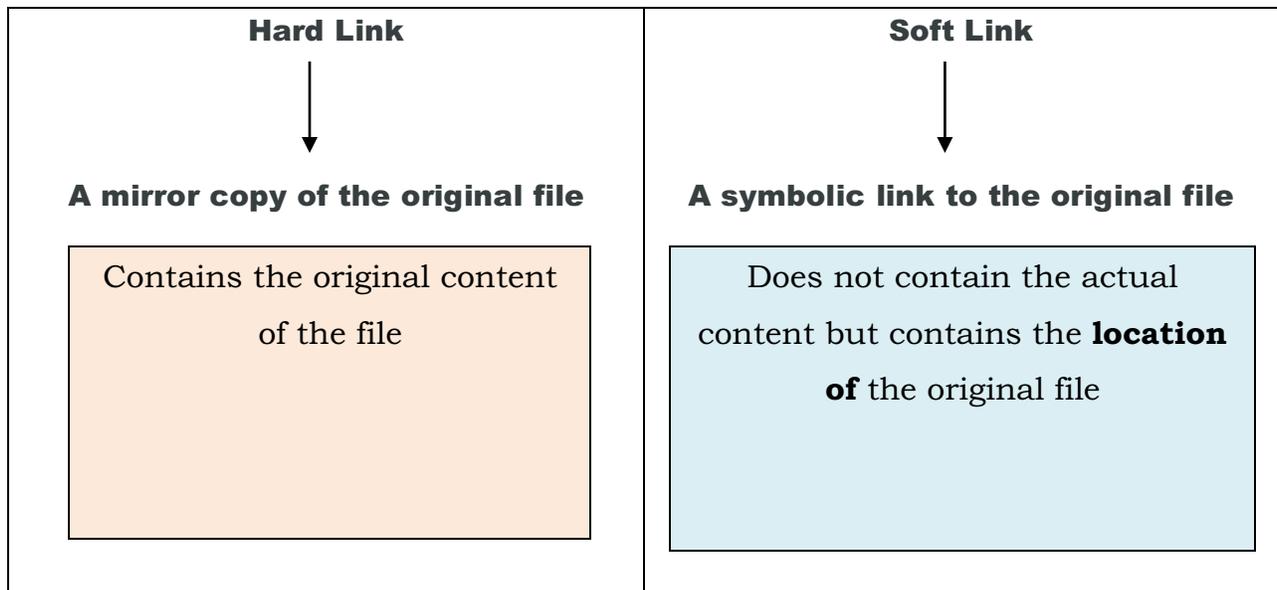
```
netstat --listen
```

Description:

Check which ports are in listening in Linux Server

Network Interface Card teaming is the process of combining multiple network cards together for performance, load balancing and to increase uptime.

Service	Default Port
DNS	53
SMTP	25
FTP	20 (Data transfer), 21 (Connection established)
SSH	22
DHCP	67/UDP (dhcp server), 68/UDP (dhcp client)
squid	3128



```
ls /bin
```

```
# List the binaries and other executable programs
```

```
ls /boot
```

```
# List the files needed to boot the operating system
```

```
ls /dev
```

```
# List the device files - typically controlled by the operating system and the system administrators
```

```
ls /etc
```

```
# List the System configuration files
```

```
ls /lib
```

```
# List the System Libraries
```

```
ls /lib64
```

```
# List the System Libraries (64 bit)
```

```
ls /proc
```

```
# List the information about running processes
```

```
ls /sbin
```

```
# List the System administration binaries
```

```
ls /var/log
```

```
# List the Log files
```

```
mkdir mydir{1,2,3,4,5}
```

Create 5 new directories:

- mydir1
- mydir2
- mydir3
- mydir4
- mydir5

```
[manju@localhost ~]$ ls -l myfiles.txt
-rw-r--r--. 1 manju nath 12 Apr 19 20:22 myfiles.txt
```

Display the permissions for the file **"myfiles.txt"**

```
find . -mtime +1 -mtime -3
```

```
# Display files that are more than 1 day old - but less than 3 days old in the current directory
```

```
find . -name "s*" -ls
```

```
# Find files that start with the letter "s" and perform the command "ls" on them
```

```
find . -size +3M
```

```
# Find files that is larger than 3 megabytes
```

```
[manju@localhost ~]$ cat myfile.txt
```

```
ffff
```

```
b
```

```
eee
```

```
cc
```

```
[manju@localhost ~]$ cat myfile.txt | sort
```

```
b
```

```
cc
```

```
eee
```

```
ffff
```

```
[manju@localhost ~]$ touch file1; touch file2
```

```
[manju@localhost ~]$ ls file{1,2}
```

```
file1 file2
```

```
[manju@localhost ~]$ NUMLOGINS=$(who | grep $USER | wc -l)
```

```
[manju@localhost ~]$ echo You have $NUMLOGINS login sessions
```

```
You have 2 login sessions
```

Command:

```
chmod go-rwx myfiles.txt
```

Description:

Remove read write and execute permissions on the file **"myfiles.txt"** for the group and others

Command:

```
chmod a+rw myfiles.txt
```

Description:

Give read and write permissions on the file "myfiles.txt" to all

Command:

```
!-3
```

Description:

Repeats the third most recent command

```
[manju@localhost ~]$ echo $OSTYPE
```

```
linux-gnu
```



The current operating system you are using

Command:

```
df -i /dev/sda1
```

Description:

Check Inodes on File system

Command:

```
ls -il myfiles.txt
```

Description:

Find Inode number of File (myfiles.txt)

Command:

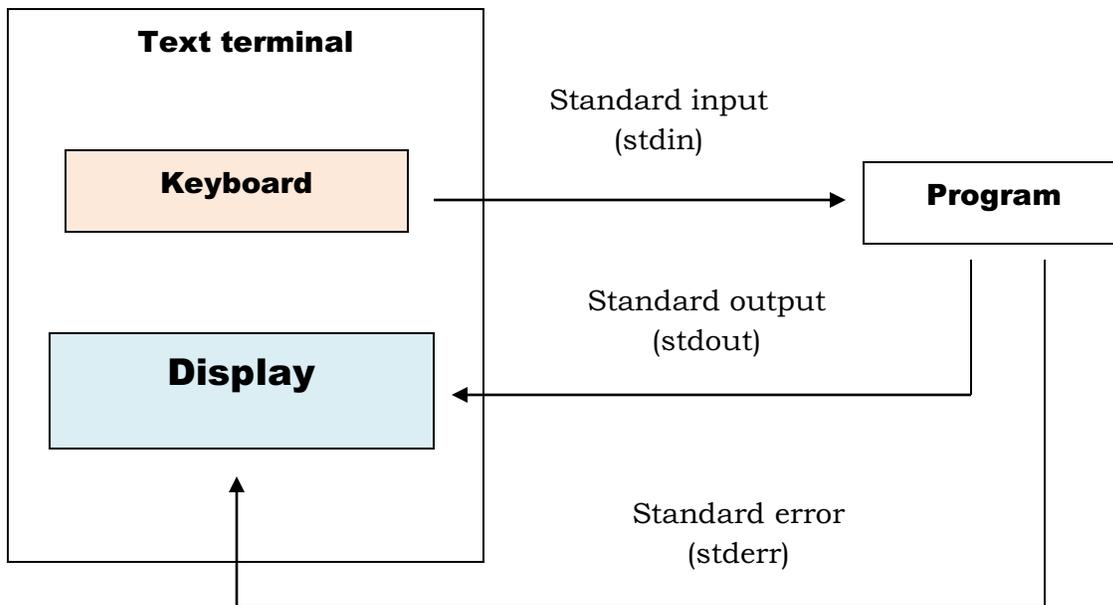
```
getfacl myfiles.txt
```

Description:

Check ACL (Access control list) configured on a file (myfiles.txt)

SSH (**Secure Shell or Secure Socket Shell**) is a network protocol that gives users and system administrators a secure way to access a computer over an unsecured network.

3 standard streams in Linux:



Command:

```
du -sh /var/log/*
```

Description:

Check information of disk usage of files and directories on a machine.

Command:

```
ldd /bin/cp
```

Description:

Display dependencies of the "cp" command.

Command:

```
ldd -v /bin/cp
```

Description:

Display dependencies of the "cp" command with details.

Command:

```
ldd -u /bin/cp
```

Description:

Display unused direct dependencies of the "cp" command.

```
[manju@localhost ~]$ date; cal
```

```
Thu Apr 21 19:44:12 PDT 2022
```

```
April 2022
```

```
Su Mo Tu We Th Fr Sa
```

```
1 2
```

```
3 4 5 6 7 8 9
```

```
10 11 12 13 14 15 16
```

```
17 18 19 20 21 22 23
```

```
24 25 26 27 28 29 30
```

date command is executed followed by a **cal command**

-gt	Greater than
-lt	Lesser than
-ge	Greater than or equal to
-le	Lesser than or equal to
-eq	Equal to
-ne	Not equal to

```
[manju@localhost ~]$ date && cal
```

```
Thu Apr 21 19:44:21 PDT 2022
```

```
April 2022
```

```
Su Mo Tu We Th Fr Sa
```

```
1 2
```

```
3 4 5 6 7 8 9
```

```
10 11 12 13 14 15 16
```

```
17 18 19 20 21 22 23
```

```
24 25 26 27 28 29 30
```

cal command is executed only if the **date command** is successfully executed

```
[manju@localhost ~]$ ls *.c
```

```
hello.c vim.c
```

```
[manju@localhost ~]$ ls *.[co]
```

```
hello.c hello.o vim.c
```

```
[manju@localhost ~]$ a=`ls *.c`; echo $a
```

```
hello.c main.c vim.c
```

```
[manju@localhost ~]$ test 50 -ge 15 && echo true || echo false
```

```
true
```

```
[manju@localhost ~]$ test 50 -ge 50 && echo true || echo false
```

```
true
```

```
[manju@localhost ~]$ test 20 -le 50 && echo true || echo false
```

```
true
```

```
[manju@localhost ~]$ test 20 -le 20 && echo true || echo false
```

```
true
```

```
[manju@localhost ~]$ test 30 -eq 30 && echo true || echo false
```

```
true
```

```
[manju@localhost ~]$ test 320 -eq 30 && echo true || echo false
```

```
false
```

```
[manju@localhost ~]$ test 30 -ne 30 && echo true || echo false
```

```
false
```

```
[manju@localhost ~]$ test 320 -ne 30 && echo true || echo false
```

```
true
```

Command:

```
cat /proc/net/dev
```

Description:

Display network adapters and statistics

Command:

```
cat /proc/mounts
```

Description:

Display the mounted file system

Command:

```
telinit 0
```

Description:

Shutdown the system

```
cd /home
```

```
# Takes you to the home directory
```

```
cd ..
```

```
# Takes you one folder back
```

```
ls * [0-9] *
```

Display the files and folders that contain numbers 0 to 9

```
iconv -l
```

```
# Display the lists of known ciphers
```

```
ls -lSr | more
```

```
# Display the size of the files and directories ordered by size
```

```
du -sk * | sort -rn
```

```
# Display the size of the files and directories ordered by size
```

Command:

```
ls -lh
```

Description:

Display permissions

Command:

```
yum list
```

Description:

List all packages installed on the system

Command:

```
yum clean packages
```

Description:

Clean all the saved packages

Command:

```
yum clean headers
```

Description:

Clean package headers

Command:

```
yum clean all
```

Description:

Clean all cached information

Command:

```
yum clean metadata
```

Description:

Clean Metadata

```
ip link show
```

```
# Display the link status of all interfaces
```

```
ps -eafw
```

```
# Display Linux tasks
```

```
lsof -p $$
```

```
# Display a list of files opened by processes
```

Command:

```
find /var -atime -90
```

Description:

Find files in the /var directory that have not been accessed in the last 90 days

Command:

```
find / -name core -exec rm {} \;
```

Description:

Search for core files in the entire directory tree and delete them as found without prompting for confirmation

Command:

```
who -r
```

Description:

Check current run level of a Linux server

Bash script:

```
for i in *linux*; do rm $i; done
```

Description:

Delete all the files in the current directory that contains the word "linux"

Command:

```
awk '{print}' myfiles.txt
```

Description:

Display the content of file (myfiles.txt)

```
# Wait for 5 seconds
```

```
sleep 5s
```

```
# Wait for 5 minutes
```

```
sleep 5m
```

```
# Wait for 5 hours
```

```
sleep 5h
```

```
# Wait for 5 days
```

```
sleep 5d
```

Sleep commands used to introduce
wait time in scripts

Command:

```
In myfiles.txt hardF1
```

Description:

Create hard-link to myfiles.txt

Command:

```
cat hardF1
```

Description:

Check content of the hard link - hardF1

Command:

```
In myfiles.txt softF1
```

Description:

Create Soft-link to myfiles.txt

Command:

```
cat softF1
```

Description:

Check content of the soft link - softF1

Foreground processes	Background processes
Require a user to start them or to interact with them.	Run independently of a user.

Command:

```
ps -p 13
```

Description:

Display information about the process with process ID – 13

Command:

```
ulimit -f 100
```

Description:

Set the file size limit to 51,200 bytes

Command:

```
lsmod
```

Description:

Find out what kernel modules are currently loaded

Absolute path	Relative path
The path of a file or directory from the root directory.	The path of a file or directory from the present working directory.

Command:

```
sudo yum install php
```

Description:

Install php version 7.2

Command:

```
php -r 'echo "Hello World\r\n";'
```

Description:

Run a PHP statement from the command line without creating a file

Command:

```
php -a
```

Description:

Start a PHP interactive shell

Command:

```
du -h -d 1 /
```

Description:

Display disk usage of all top-level directories

Command:

```
yum install man
```

Description:

Install man package in Centos

Command:

```
man -f ls
```

Description:

Display man Pages and Print Short Description of the `ls` command

```
man -a ls
```

```
# Display all man Pages of the ls command
```

```
man -k ls
```

```
# Allows users to search the short command descriptions and manual page names for ls command
```

```
man -w ls
```

```
# Displays the location of the manual page of the ls command
```

```
cat /etc/redhat-release
```

```
# Display Linux distribution name and version
```

```
ls ~
```

```
# Display the contents of the home directory
```

```
ls ../
```

```
# Display the contents of the parent directory
```

Command:

```
ps -U root -u root
```

Description:

Display all process running under the root user account

Command:

```
cal -1
```

Description:

Display current month calendar

Command:

```
cal -j
```

Description:

Print the calendar in day numbers

Command:

```
su
```

Description:

Used to switch from one account to another

Command:

```
nmcli connection show
```

Description:

Display what are the network connection connected in our system

Command:

```
ps aux | grep 'telnet'
```

Description:

Searches for the id of the process 'telnet'

```
ps r
```

```
# List only running processes on Linux
```

```
ps T
```

```
# List all processes on this current terminal
```

```
ps -f
```

```
# List processes along with the parent process ID associated with the current Terminal
```

```
ps -x
```

```
# View all processes owned by you
```

```
ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem
```

```
# Display the processes using highest memory
```

```
sudo yum list --installed | more
```

```
# Lists installed packages on CentOS
```

Command:

```
sudo rpm -qa
```

```
sudo rpm -qa | more
```

Description:

Get a list of all installed packages with rpm command

Command:

```
sudo rpm -q nginx
```

Description:

Check whether nginx package installed or not

Command:

```
sudo rpm -q bash
```

Description:

Check whether bash package installed or not

Command:

```
sudo yum history
```

Description:

List all installed packages with yum on CentOS history command

Command:

```
sudo yum history info 2
```

Description:

Examine history entries in detail using transaction ID [\[2\]](#)

Command:

```
file /etc/passwd
```

Description:

Displays the file type of a given file

```
[root@localhost manju]# file /etc/passwd
/etc/passwd: ASCII text
```

Command:

```
wc /etc/passwd
```

Output:

```
46 91 2373 /etc/passwd
```



The `/etc/passwd` file has 46 lines, 91 words and 2373 letters present in it

Command:

```
grep root: /etc/passwd
```

Description:

Display all lines from `/etc/passwd` containing the string "**root**"

Command:

```
grep -n root /etc/passwd
```

Description:

Display all lines from `/etc/passwd` containing the string "**root**" with line numbers

Command:

```
grep -c false /etc/passwd
```

Description:

Display the number of accounts that have `/bin/false` as their shell

Command:

```
grep ^root: /etc/passwd
```

Description:

Display all lines from `/etc/passwd` starting with the string "**root**" followed by colon

Command:

```
last | head
```

Description:

Displays information about the users who logged in and out of the system

(Display the top 10 lines only)

```
lastb
```

```
# Display the last unsuccessful login attempts
```

```
du /etc/passwd
```

```
# Display the disk usage of a /etc/passwd file
```

```
killall proc
```

```
# Kill all the process named proc
```

```
wget https://repo.mysql.com/mysql80-community-release-el8-1.noarch.rpm
```

```
# Download the RPM file to install
```

```
sudo yum localinstall mysql80-community-release-el8-1.noarch.rpm
```

```
# Install the RPM file
```

```
sudo yum localinstall https://repo.mysql.com/mysql80-community-release-el7-1.noarch.rpm

# Install the RPM package via URL

curl --version

# Display curl Version

curl -O http://website.com/myfiles.tar.gz

# Download the file (myfiles.tar.gz) from url "http://website.com/myfiles.tar.gz"

# Saved as myfiles.tar.gz

curl -o files.tar.gz http://website.com/myfiles.tar.gz

# Download the file (myfiles.tar.gz) from url "http://website.com/myfiles.tar.gz"

# Saved as files.tar.gz
```

```
echo 'https://repo.mysql.com/mysql80-community-release-el8-1.noarch.rpm' > urls.txt

xargs -n 1 curl -O < urls.txt

# Download files from a list of URLs in "urls.txt" file

exit 110

# Exit from the terminal window

sudo -l

# know which commands are permitted and not permitted on the current host
```

Command:

```
echo -e "\thello\nworld"
```



```
world    hello
```

```
history | grep cd | head -12
```

Searches history of first 12 commands which have cd word match

Disadvantages of Open Source Operating System:

- Difficulty to use
- Compatibility Issues

Command:

```
rpm -qa | grep ftp
```

Description:

Check all installed packages of ftp

Command:

```
find /home -mtime +120
```

Description:

Find files in the /home directory which were modified more than 120 days ago

Samba enables Linux / UNIX machines to communicate with Windows machines in a network.

- The **/etc directory** contains configuration files in Linux.
- **The Network File System (NFS)** is a mechanism for storing files on a network.
- **"init"** is the first process in linux which is started by the kernel and its process id is 1.

```
egrep "Hello|Einstein" file.txt
```

Returns line with Hello or Einstein in the **file.txt**

```
date "+%s"
```

Prints the date in seconds

```
cat file.txt | uniq
```

Display duplicate record only once

Command:

```
cd ../../..
```

Takes you three folders back

Command:

```
ps -ef | grep xlogo
```

Description:

List all the processes on the system containing the string 'xlogo'

```
echo -n "abc";echo "def"
```

abcdef

```
echo "abc";echo "def"
```

abc

def

```
ls -ltr /etc
```

List the files in **/etc** in order of last modification

Command:

```
ls -Rlh /var | grep [0-9]M
```

Description:

List the files in `/var` larger than 1 megabyte but less than 1 gigabyte

Command:

```
ls -lhS
```

Description:

List files by size

```
cat /etc/passwd /etc/group
```

```
# Display the contents of multiple files (/etc/passwd and /etc/group)
```

```
find /tmp -name *.txt -exec rm -f {} \;
```

```
# Searches for all files in the /tmp directory named *.txt and deletes them
```

```
echo "use" "of" "Linux"
```



```
use of Linux
```



```
watch -n 5 tail -n 3 /etc/passwd
```

```
# Display the end of the /etc/passwd file every 5 seconds
```

```
watch -n 1 'ls -l | wc -l'
```

```
# Monitor the number of files in a folder
```

```
watch -t -n 1 date
```

```
# Display the clock
```

```
find / -name "*.txt"
```

```
# Search all files with .txt extension
```

```
find . -name "*file*"
```

```
# Search all files containing "file" in the name
```

```
find /home -name "*file*"
```

```
# Search all files in /home containing "file" in the name
```

```
grep -nre "hello computer" ./*
```

```
# Search for files containing the string "hello computer" in the current directory
```

```
(echo In Linux; exit 0) && echo OK || echo exit
```

In Linux

OK

```
(echo In Linux; exit 4) && echo OK || echo exit
```

In Linux

exit

Command:

`free -t -m`

Display free memory size in MB

Command:

`gnome-system-monitor`

Description:

Displays what programs are running and how much processor time, memory and disk space are being used

`lsblk -m`

Display device permissions and ownership

`lsblk -S`

Display **SCSI** (Small Computer Systems Interface) devices

`lsblk -n`

List **devices** without the header

Command:

```
ls -l ~
```

Description:

Check the file and folder permissions

Command:

```
ls ./Documents
```

Description:

Display the list of files that reside in the Documents folder

```
ls -R
```

```
# List out all the contents of subdirectories
```

```
compgen -c
```

```
# Displays the list of all commands which we can use in the command-line interface
```

- `pwd -L` → Prints a symbolic path
- `pwd -P` → Prints the actual full path

```
hostnamectl
```

```
# Display system information including operating system, kernel and release version
```

Command:

```
find . -type f
```

Find files

Command:

```
find . -type d
```

Description:

Find directories

```
find . -iname "*.jpg"
```

Find files by case-insensitive extension (ex: .jpg, .JPG, .jpG)

```
find . -type f -perm 777
```

Find files by octal permission

```
cd logs; ls -lt | head; du -sh ; df -h
```

Concatenating all of the above tasks in a single line using the ";" operator

```
{ echo "Albert Einstein"; pwd; uptime; date; }
```



```
Albert Einstein
/home/manju
00:26:53 up 28 min, 2 users, load average: 0.00, 0.01, 0.05
Tue Mar 29 00:26:53 PDT 2022
```

```
cal; { date; uptime; }; pwd
```

```
March 2022
```

```
Su Mo Tu We Th Fr Sa
```

```
1 2 3 4 5
```

```
6 7 8 9 10 11 12
```

```
13 14 15 16 17 18 19
```

```
20 21 22 23 24 25 26
```

```
27 28 29 30 31
```

```
Tue Mar 29 00:52:38 PDT 2022
```

```
00:52:38 up 54 min, 2 users, load average: 0.00, 0.01, 0.05
```

```
/home/manju
```

```
shutdown -r
```

```
# Kicks off a reboot
```

```
shutdown +0
```

```
# Shuts down the system immediately
```

```
shutdown -r +5
```

```
# Begins a reboot of the system in five minutes
```

Command:

kill 12838

Terminate the process with process ID 12838

Command:

ss -t -r state established

Description:

List all the established ports

shutdown -Fr now

Force the file system check during
reboot

ss -t -r state listening

List all sockets in listening state

mtr google.com

Diagnose Network Issues

sudo tcpdump --list-interfaces

List all network interfaces

```
ls -al --time-style=+%D | grep `date +%D`
```



List today's files only

```
mpstat -P 0
```

```
# Print processor statistics and helps to monitor CPU utilization on the system
```

```
chmod 777 myfiles.txt
```

```
# Assign (read, write and execute) permission to everyone
```

```
chmod 766 myfiles.txt
```

```
# Assign full permission to the owner and read and write permission to group and others
```

```
chmod -x myfiles.txt
```

```
# Remove the execution permission of myfiles.txt file
```

```
history 30
```

```
# List the last 30 commands we have entered on the system
```

```
find ~ -empty
```

```
# Find all empty files in home directory
```

```
gzip -l *.gz
```

```
# Display compression ratio of the compressed file
```

Command:

```
ps -efH | more
```

View current running processes in a tree structure

Command:

```
df -T
```

Description:

Display what type of file system

```
mkdir ~/temp
```

Creates a directory called temp under home directory

```
ls *.py
```

```
# List all Python files
```

```
chsh -l
```

```
# Display the list of all shells
```

```
ipcs -a
```

```
# Display details about message queue, semaphore and shared memory
```


ipcs -q



Lists only message queues for which the current process has read access

```
ipcs -s
```

```
# List the accessible semaphores
```

```
ipcs -m
```

```
# List all the Shared Memory
```

```
quotastats
```

```
# Display the report of quota system statistics gathered from the kernel
```

```
rpcinfo
```

```
# Display all of the RPC (Remote Procedure Call) services of the local host
```

```
slabtop
```

```
# Display kernel slab cache information in real-time
```

```
tload
```

```
# Display a graph of the current system load average to the specified tty
```

```
cat /proc/devices
```

```
# Display the device drivers configured for the currently running kernel
```

```
cat /proc/dma
```

```
# Display the DMA channels currently used
```

```
cat /proc/filesystems
```

```
# Display the file systems configured into the kernel
```

```
cat /proc/kmsg
```

```
# Display the messages generated by the kernel
```

```
cat /proc/loadavg
```

```
# Display the system load average
```

```
ls /proc/net
```

```
# List the network protocols
```

```
ls /etc/udev
```

```
# List the contents of udev configuration directory
```

```
cat /proc/stat
```

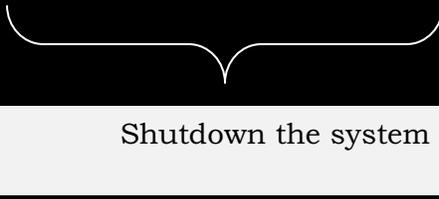
```
# Display the system operating statistics
```

```
cat /proc/uptime
```

```
# Display the time the system has been up
```

Command:

```
poweroff -i -f
```



Shutdown the system

```
[2 = 2 ] ; echo $?
```

```
# 0 (logically TRUE)
```

```
[ 2 = 6 ] ; echo $?
```

```
# 1 (logically FALSE)
```

```
type echo
```

```
# echo is a shell builtin
```

```
find /usr -print
```

```
# Find and print all files under "/usr"
```

```
systemctl list-units --type=target
```

```
# List all target unit configuration
```

```
systemctl list-units --type=service
```

```
# List all service unit configuration
```

```
systemctl list-sockets
```

```
# List all socket units in memory
```

```
systemctl list-timers
```

```
# List all timer units in memory
```

```
systemctl list-dependencies --all
```

```
# List dependency of all unit services
```

```
systemctl poweroff
```

```
# Shut down the system
```

```
systemctl reboot
```

```
# Shut down and reboot the system
```

```
systemctl suspend
```

```
# Suspend the system
```

```
netstat -ln --tcp
```

```
# Find listening TCP ports (numeric)
```

```
systemctl hibernate
```

```
# Hibernate the system
```

```
loginctl user-status
```

```
# Display terse runtime status information of the user of the caller's session
```

```
loginctl session-status
```

```
# Display terse runtime status information of the caller's session
```

```
ip route show
```

```
# Display all the routing table in numerical addresses
```

```
ip neigh
```

```
# Display the current content of the ARP (Address Resolution Protocol) cache tables
```

```
netstat -l --inet
```

```
# Find listening ports
```

Command:

atq

Lists the user's pending jobs

```
lsof | grep deleted
```

```
# Print all deleted files which are claiming disk space
```

```
echo $$
```

```
# Display the Process ID of the current process
```

```
echo $!
```

```
# Display the Process ID of most recently started background job
```

```
date --date="yesterday"
```

```
# Display yesterday's date
```

```
date --date="10 days ago"
```

```
# Display date 10 days ago
```

```
ls / | wc -w
```

```
# List the number of directories in the root directory
```

```
sudo sfdisk -l -uM
```

```
# Display the size of each partition in MB
```

```
sudo parted -l
```

```
# Lists out the partition details
```

```
df -h | grep ^/dev
```

```
# Filter out real hard disk partitions/file systems
```

```
sudo blkid
```

```
# Displays information about available block devices
```

```
ls / > info.txt
```

```
cat info.txt
```

```
bin
```

```
boot
```

```
dev
```

```
etc
```

```
home
```

```
lib
```

```
lib64
```

```
media
```

```
mnt
```

```
opt
```

```
proc
```

```
root
```

```
run
```

```
sbin
```

```
srv
```

```
sys
```

```
tmp
```

```
usr
```

```
var
```

```
export NAME="Albert Einstein"
```

```
echo $NAME
```

```
Albert Einstein
```

```
TZ=US/Pacific date
```

```
# Display the current date/time in US/Pacific time zone
```

```
ls -l /etc/shadow
```

```
# Display the user password stored in an encrypted form and the password expiry data
```

```
sudo journalctl --since yesterday
```

```
# Display all the logs since yesterday
```

```
sudo journalctl --since "2019-12-10 13:00:00"
```

```
# Display all the logs since 2019-12-10 13:00:00
```

```
journalctl -disk-usage
```

```
# Display the total size of the journal logs
```

Command:

```
ls -m
```

Prints out directories and files separated by a comma

```
ls -Q
```

Add quotation marks to all directories and files

```
ss -f unix
```

List Unix Sockets

```
echo *.desktop
```

Lists all of the .desktop files in the current directory

```
ss --raw
```

List Raw Sockets

```
tracert www.google.com
```

Traces a path to a network host (www.google.com) discovering MTU along the path

```
echo -e "123\b4"
```

124

3 is over-written by 4

```
echo -e "123\r456"
```

456

123 is overwritten by 456

```
echo D*
```

Lists all of the files and directories in the current directory whose name starts with letter D

```
echo $'I\'m a Linux Learner.'
```



```
I'm a Linux Learner.
```

```
echo $USER
```

```
# Print the name of the currently logged in user
```

```
echo -e "\033[0;32mGREEN"
```

```
GREEN
```

```
echo -e "\033[0;31mRED"
```

```
RED
```

```
echo -e 'Hello, \vWorld!'
```



```
Hello,
```

```
World!
```

```
echo "This is the list of directories and files on this system: $(ls)"
```

```
This is the list of directories and files on this system: Desktop
```

```
Documents
```

```
Downloads
```

```
Music
```

```
Pictures
```

```
Public
```

```
Templates
```

```
Videos
```




```
echo *s

# Print all files and folders that end by letter "s"

echo [[:upper:]]*

# Print all files and folders that start by upper case character

echo $((2 + 3))

→ 5

echo $((($((2**2)) * 3))

→ 12

echo Four divided by two equals $((4/2))

→ Four divided by two equals 2

echo Capital-{A,B,C}-Letter

→ Capital-A-Letter Capital-B-Letter Capital-C-Letter

echo {1..5}

→ 1 2 3 4 5

echo {A..Z}

→ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

echo x{P{1,2},Q{3,4}}y

→ xP1y xP2y xQ3y xQ4y
```

```
echo The total price is $500.00
```

→ The total price is 00.00

```
echo "$USER $((3*2)) $(cal)"
```

```
manju 6      March 2022
```

```
Su Mo Tu We Th Fr Sa
```

```
    1  2  3  4  5
```

```
  6  7  8  9 10 11 12
```

```
13 14 15 16 17 18 19
```

```
20 21 22 23 24 25 26
```

```
27 28 29 30 31
```

```
echo -e "\aMy Laptop shut \"down\"."
```

→ My Laptop shut "down".

```
echo -e "C:\\\\WIK2N\\\\LINUX_OS.EXE"
```

→ C:\WIK2N\LINUX_OS.EXE

```
echo $(cal)
```

```
March 2022 Su Mo Tu We Th Fr Sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
```

```
echo "$(cal)"
```

```
March 2022
```

```
Su Mo Tu We Th Fr Sa
```

```
    1  2  3  4  5
```

```
  6  7  8  9 10 11 12
```

```
13 14 15 16 17 18 19
```

```
20 21 22 23 24 25 26
```

```
27 28 29 30 31
```

```
echo The total price is \500.00
```

→ The total price is \$500.00

```
sudo lsof -i -P -n | grep LISTEN
```

Check ports in use

```
sudo netstat -tulpn | grep LISTEN
```

```
sudo ss -tulw
```

```
# Check what ports are open
```

```
netstat -ap | grep ssh
```

```
# Find out on which port a program is running
```

```
[root@localhost manju]# ipcs -m -l
```

```
----- Shared Memory Limits -----
```

```
max number of segments = 4096
```

```
max seg size (kbytes) = 18014398509465599
```

```
max total shared memory (kbytes) = 18014398442373116
```

```
min seg size (bytes) = 1
```

Lists the Limits for Inter-process
Communication facility

```
[root@localhost manju]# ipcs -m -p
```

```
----- Shared Memory Creator/Last-op PIDs -----
```

shmid	owner	cpid	lpid
131072	manju	2998	3135
163841	manju	2998	3135
327682	manju	3277	6920
360451	manju	2827	1406

Display the process ids that accessed
Inter-process Communication facility
recently

```
[root@localhost manju]# ipcs -u
```

```
----- Messages Status -----
```

```
allocated queues = 0
```

```
used headers = 0
```

```
used space = 0 bytes
```

```
----- Shared Memory Status -----
```

```
segments allocated 4
```

```
pages allocated 2432
```

```
pages resident 319
```

```
pages swapped 0
```

```
Swap performance: 0 attempts 0 successes
```

```
----- Semaphore Status -----
```

```
used arrays = 0
```

```
allocated semaphores = 0
```

```
dmidecode -t 16
```

```
# Display the maximum RAM supported by the system
```

```
dmidecode -t baseboard
```

```
# Display all the system baseboard related information
```

```
dmidecode -t bios
```

```
# Display the BIOS information
```

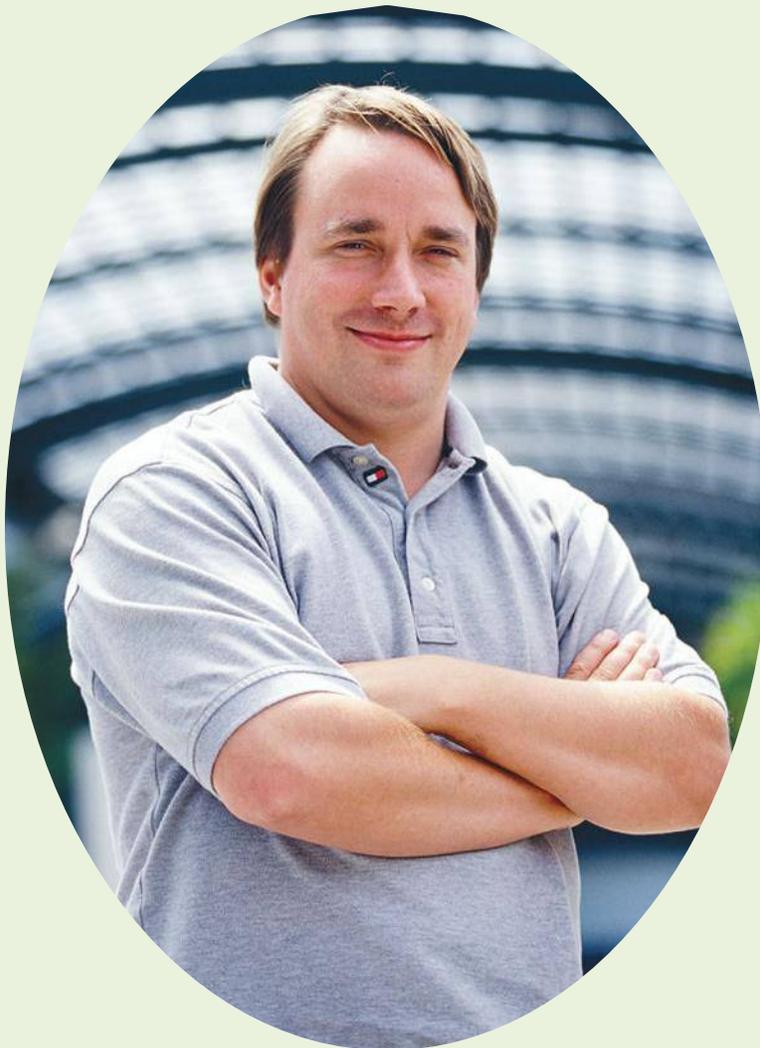
Display the status of current usage
of **Inter-process Communication** facility

Command:

```
dmidecode -t system
```

Description:

Display the information about the manufacturer, model and serial number of the system



The Linux philosophy is **'Laugh in the face of danger'**. Oops. Wrong One. 'Do it yourself'. Yes, that's it.

Linus Torvalds

```
nmcli con show -a
# Display the active network connections

netstat -r
# Display the kernel routing table

yum install nmap
# Install nmap on CentOS

nmap google.com
# Scan a hostname

nmap 193.169.1.1
# Scan a ip address

nmap --iflist
# Display host interfaces and routes

echo [![:digit:]]*
# Print all files and folders that are not beginning with a numeral

echo *[:lower:]123]
# Print all files and folders ending with a lowercase letter or the numeral

echo g*
# Print all files and folders beginning with "g"
```

```
echo b*.txt
```

```
# Print all files and folders beginning with "b" followed by any characters and ending with ".txt"
```

```
echo [abc]*
```

```
# Print all files and folders beginning with either "a", "b" or "c"
```

```
netstat -t
```

```
# Display the download status of active connections
```

```
netstat -x
```

```
# Display Information about all connections, listeners and shared endpoints for Network Direct
```

```
netstat -n
```

```
# Numerical display of addresses and port numbers
```

```
echo $LANG
```

```
# Display the language of a Linux system
```

```
echo "AAA" | grep AAA
```

```
→ AAA
```

```
echo "AAA" | grep BBB
```

```
→
```

```
echo "AAA" | grep -E 'AAA|BBB'
```

```
→ AAA
```

```
echo "BBB" | grep -E 'AAA|BBB'
```

```
→ BBB
```

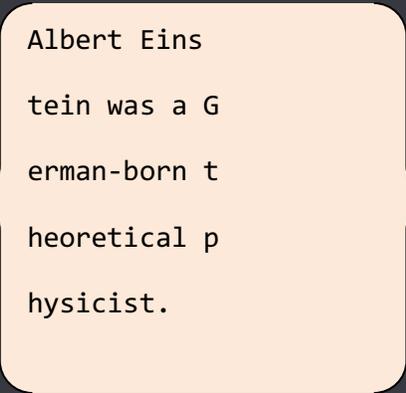
```
echo "albert einstein" | tr a-z A-Z
```

→ ALBERT EINSTEIN

```
echo "albert einstein" | tr [:lower:] E
```

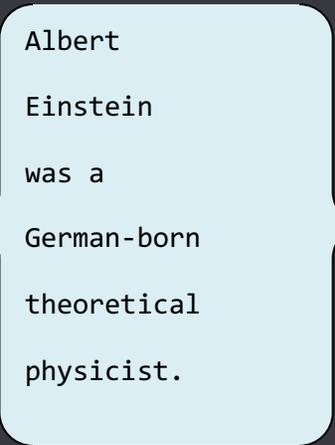
→ EEEEEEE EEEEEEEE

```
echo "Albert Einstein was a German-born theoretical physicist." | fold -w 12
```



```
Albert Eins  
tein was a G  
erman-born t  
heoretical p  
hysicist.
```

```
echo " Albert Einstein was a German-born theoretical physicist." | fold -w 12 -s
```



```
Albert  
Einstein  
was a  
German-born  
theoretical  
physicist.
```

```
printf "English theoretical physicist: %s\n" Hawking
```

→ English theoretical physicist: Hawking


```
ls /usr/bin | pr -3 -w 65 | head
```

Display a directory listing of /usr/bin in a paginated, three-column output format

```
for i in A B C D; do echo $i; done
```

```
A
B
C
D
```

```
for i in {A..D}; do echo $i; done
```

```
A
B
C
D
```

```
for i in file*.txt; do echo $i; done
```

```
file1.txt
file.txt
```

```
echo ${!BASH*}
```

List all the variables in the environment with names that begin with BASH

```
bc <<< "6+6"
```

```
12
```

```
Scientists=("Einstein" "Hawking" "Darwin"); for i
in ${Scientists[*]}; do echo $i; done
```

```
Einstein
Hawking
Darwin
```

```
Scientists=("Einstein" "Hawking" "Darwin"); for i
in "${Scientists[*]}"; do echo $i; done
```

```
Einstein Hawking Darwin
```

```
df -k
```

```
# Check the file system space
```

```
df -h
```

Display disk space in
human-readable format

```
ls -alh
```

```
# List all folders in directory with details
```

```
find /home -name file.txt
```

```
# Check all files in /home directory with the name file.txt
```

```
find /home -iname File.txt
```

```
# Search all files in /home directory irrespective to case sensitive
```

```
find / -ctime +90
```

```
# Search for the files which were modified more than 90 days back
```

```
find / -size 0c
```

```
# Search all empty files
```

```
find / -size +1G
```

```
# Search all files and folders which are more than 1GB
```

```
df -a
```

```
# Display the file system's complete disk usage
```

```
df -i
```

```
# Display used and free inodes
```

```
du -ch *.png
```

```
# Display the size of each png file in the current directory
```

```
du -a /etc/ | sort -n -r | head -n 10
```

```
# List top 10 directories consuming disk space in /etc/
```

```
ac
```

```
# Display the total amount of time users are connected to the system
```

```
ac --individual-totals
```

```
# Display a report on login times for individual users
```

```
cancel
```

```
# Cancels print jobs
```

```
yum install finger
```

```
# Install finger tool (CentOS)
```

```
finger manju
```

```
# Display the details of a user "manju"
```

```
chfn
```

```
# Allows you to modify user's information
```

```
finger -s manju
```

```
# Display idle status and login details of a user "manju"
```

```
groups
```

```
# List all Groups the Current User is a Member of
```

```
id -nG
```

```
# List all Groups the Current User is a Member of
```

```
groupadd mygroup
```

```
# Create a new group named "mygroup"
```

```
groupdel mygroup
```

```
# Delete a group named "mygroup"
```

```
less /etc/group
```

```
# List all Groups
```

```
getent group
```

```
# List all Groups
```

```
usermod -a -G mygroup manju
```

```
# Add an existing user "manju" to a group "mygroup"
```

```
userdel manju
```

```
# Delete a user "manju"
```

```
chgrp mygroup test.txt
```

```
# Change the owning group of the file test.txt to the group named "mygroup"
```

```
sudo depmod -a

# Generates a list of all kernel module dependences and associated map files

dirname /usr/bin

→ /usr

dirname /Desktop/root

→ /Desktop

dmesg > kernel_messages.txt

# Read all messages from kernel ring buffer to a file "kernel_messages.txt"

dmesg | grep -i memory

# Display the kernel messages which relate to memory usage

egrep -c '^Hello|World$' myfiles.txt

# Count the number of lines in myfiles.txt which begin with the word 'Hello' or end with the word 'World'

ex myfiles.txt

# Edits the file myfiles.txt

expand myfiles.txt

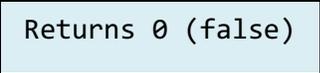
# Expand the file myfiles.txt - changing tabs to spaces - and display on standard output

expand --tabs=10 myfiles.txt > myfiles0.txt

# Convert the tabs in the file myfiles.txt to 10 spaces each, and write the output to myfiles0.txt
```

```
expr 2 = 5
```

```
# 0
```



```
Returns 0 (false)
```

```
fc -l
```

```
# Lists the history of commands
```

```
!l
```

```
# Executes the most recently executed command that begins with the letter "l"
```

```
fc -e - l
```

```
# Executes the most recently executed command that begins with the letter "l"
```

```
fmt myfile.txt
```

```
# Display a reformatted version of the file "myfile.txt "
```

```
fmt < myfile.txt > myfile0.txt
```

```
# Reformat "myfile.txt" and write the output to the file "myfile0.txt"
```

```
finger -p manju
```

```
# Display information about the user "manju"
```

```
fold -w5 myfile.txt > myfile0.txt
```

```
# Wrap the lines of myfile.txt to a width of 5 characters and writes the output to myfile0.txt
```

```
for file in *.txt ; do wc -l $file ; done
```

```
# Performs a word count of all files in the current directory with the .txt extension
```

```
grep manju /etc/passwd

# Search /etc/passwd for user "manju"

groupmod -n group mygroup

# Change the group "mygroup" to "group"

head myfiles.txt

# Display the first 10 lines of "myfiles.txt"

head -15 myfiles.txt

# Display the first 15 lines of "myfiles.txt"

head myfiles.txt myfiles0.txt

# Display the first 10 lines of both myfiles.txt and myfiles0.txt - with a header before each that indicates the file name

head -n 5K myfiles.txt

# Display the first 5,000 lines of "myfiles.txt"

head -n 4 *.txt

# Display the first 4 lines of every file in the working directory whose file name ends with the .txt extension

iostat

# Display operating system storage input and output statistics

last reboot | less

# Display listing of last logged in users and system last reboot time and date
```

```
last -x | less
```

```
# Display last shutdown date and time
```

```
last shutdown
```

```
# Display last shutdown date and time
```

```
ldd /bin/bash
```

```
# Display the shared library dependencies of the program /bin/bash
```

```
less -N myfiles.txt
```

```
# View the file myfiles.txt - displaying a line number at the beginning of each line
```

```
ls *.{html,php,txt}
```

```
# List all files with .html, .php and .txt file extension
```

```
ls /
```

```
# List the contents of root directory
```

```
ls [aeiou]*
```

```
# List only files that begin with a vowel (a, e, i, o and u)
```

```
lsof -i -U
```

```
# List all open Internet, x.25 (HP-UX) and UNIX domain files
```

```
lsof -i 4 -a -p 555
```

```
# List all open IPv4 network files in use by the process whose Process ID is 555
```

```
lsof -i 6
```

List only open IPv6
network files


```
xz myfiles.txt
# Compress the file "myfiles.txt" into "myfiles.txt.xz"

xz -dk myfiles.txt.xz
# Decompress "myfiles.txt.xz" into "myfiles.txt"

mkdir -m a=rwx dir
# Create the directory "dir" and set its file mode so that all users may read, write and execute it

modinfo snd
# Display all available information about the "snd" Linux kernel module

more +3 myfiles.txt
# Display the contents of file "myfiles.txt" beginning at line 3

more +/"Hello" myfiles.txt
# Display the contents of file "myfiles.txt" beginning at the first line containing the string "Hello"

netstat -g
# Display multicast group membership information for both IPv4 and IPv6

netstat -c
# Print netstat information every few second

netstat -natp
# Display statistics about active Internet connections
```

```
netstat -rn
```

```
# Display the routing table for all IP addresses bound to the server
```

```
netstat -an
```

```
# Display information about all active connections to the server
```

```
od -b myfiles.txt
```

```
# Display the contents of "myfiles.txt" in octal format
```

```
od -Ax -c myfiles.txt
```

```
# Display the contents of "myfiles.txt" in ASCII character format - with byte offsets displayed as hexadecimal
```

```
trap -l
```

```
# Display a list of signal names and their corresponding numbers
```

```
trap
```

```
# Display a list of the currently-set signal traps
```

```
yum list openssh
```

```
# Search for a package with a name "OpenSSH"
```

```
yum grouplist
```

```
# List all available Group Packages
```

```
yum repolist
```

```
# List all enabled Yum repositories
```

```
yum repolist all
```

```
# List all Enabled and Disabled Yum Repositories
```

```
paste 1.txt 2.txt
```

```
# Display the contents of 1.txt and 2.txt side-by-side
```

```
ls -a | pr -n -h "Files in $(pwd)" > dc.txt
```

```
cat dc.txt
```

```
2022-04-02 01:10      Files in /home/manju      Page 1
```

```
1  .
2  ..
3  1.txt
4  2.txt
5  .bash_history
6  .bash_logout
7  .bash_profile
8  .bashrc
9  bio.txt
10 .cache
11 .config
12 Data.txt
13 Desktop
14 dir
15 Documents
```

```
printf "Hi, I'm %s.\n" $LOGNAME
```

→ Hi, I'm manju.

```
printf "%.6s" 6 "abcdefg"
```

→ abcdef

```
ps -eLf
```

Get information about threads

```
ps axms
```

Get information about threads

- ps -eo euser,ruser,suser,fuser,f,comm,label
- ps axZ
- ps -eM

Get security information

Command:

```
pstree -h
```

Description:

Display all processes as a tree, with the current process and its ancestors highlighted

```
rm -- 1.txt
```

Delete "1.txt" file in the current directory

```
rm ./1.txt
```

```
rm /home/manju/2.txt
```

Delete "2.txt" file in the directory "/home/manju"

```
ip route list
```

```
# List current routing table
```

```
route -n
```

```
# Display routing table for all IPs bound to the server
```

```
script -c 'echo "Hello, World!"' hello.txt
```

```
Script started, file is hello.txt
```

```
Hello, World!
```

```
Script done, file is hello.txt
```

```
cat hello.txt
```

```
Script started on Sat 02 Apr 2022 03:24:52 AM PDT
```

```
Hello, World!
```

```
Script done on Sat 02 Apr 2022 03:24:52 AM PDT
```

```
sfdisk -s
```

```
# List the sizes of all disks
```

```
ls -d ~/.ssh
```

```
# Check if the .ssh directory exists or not
```

```
sha224sum myfiles.txt
```

```
# Display the SHA224 checksum of the "myfiles.txt" file in the current directory
```

```
sha256sum myfiles.txt
```

```
# Display the SHA256 checksum of the "myfiles.txt" file in the current directory
```

```
sha384sum myfiles.txt
```

```
# Display the SHA384 checksum of the "myfiles.txt" file in the current directory
```

```
sha512sum myfiles.txt
```

```
# Display the SHA512 checksum of the "myfiles.txt" file in the current directory
```

```
shutdown 8:00
```

```
# Schedule the system to shut down at 8 A.M
```

```
shutdown 20:00
```

```
# Schedule the system to shut down at 8 P.M
```

```
shutdown +15 "The system will be shutdown in 15 minutes."
```

```
# Schedule the system to shut down in 15 minutes with the normal message alerting users that the system is shutting down
```

```
shutdown -P now
```

```
# Power off the system immediately
```

```
sleep 10
```

```
# Delay for 10 seconds
```

```
startx -- -depth 16
```

```
# Start an X session at 16 bits color depth
```

```
time cal
# Reports how long it took for the "cal" command to complete

tr "[:lower:]" "[:upper:]" < myfiles.txt
# Translate the contents of "myfiles.txt" to uppercase

tr -cd "[:print:]" < myfiles.txt
# Remove all non-printable characters from "myfiles.txt"
```

```
cat myfiles.txt
```

Hello World

```
xlsfonts
# Lists all fonts available to the default X server and display
```

```
tr -cs "[:alpha:]" "\n" < myfiles.txt
```

Hello
World

xset q

Display the values of all current X Window System preferences

```
uncompress myfiles.txt.xz
# Uncompress the file "myfiles.txt.xz"
```

```
w manju
# Display information for the user named "manju"

write albert
# Write a message to the user "albert"
```

```
yes | rm -i *.txt
```

Remove all files with the extension **.txt** from the current directory

What is Linux and why is it so popular?

Whether you know it or not you are already using Linux (the best-known and most-used open source operating system) every day. From supercomputers to smartphones, the Linux operating system is everywhere. As an operating system, Linux is a family of open source Unix-like software based on the Linux kernel - that sits underneath all of the other software on a computer, receiving requests from those programs and relaying these requests to the computer's hardware. With regard to careers, it is becoming increasingly valuable to have Linux skills rather than just knowing how to use Windows. In general, Linux is harder to manage than Windows, but offers more flexibility and configuration options.

Every desktop computer uses an operating system. The most popular operating systems in use today are: Windows, Mac OS, and LINUX. Linux is the best-known notoriously reliable and highly secure open source portable operating system -- very much like UNIX -- that has become very popular over the last several years -- created as a task done for pleasure by Linus Torvalds - - computer science student at the University of Helsinki in Finland -- in the early 1990s and later developed by more than a thousand people around the world.

Linux is fast, free and easy to use, that sits underneath all the other software on a computer – runs your computer -- handling all interactions between you and the hardware i.e., whether you're typing a letter, calculating a money budget, or managing your food recipes on your computer, the Linux operating system (similar to other Operating Systems, such as Windows XP, Windows 7, Windows 8, and Mac OS X) provides the essential air that your computer breathes.

Linux is the most important technology advancement of the twenty-first century and Licensed under the General Public License (GPL) that Linux uses ensures that the software will always be open to anyone and whose source code is open and available for any user to check, which makes it easier to find and repair vulnerabilities and it power the laptops, development machines and

servers at Google, Facebook, Twitter, NASA, and New York Stock Exchange, just to name a few. Linux has many more features to amaze its users such as: Live CD/USB, Graphical user interface (X Window System) etc.

Why LINUX?

Although Microsoft Windows (which is the most likely the victim of viruses and malware) has made great improvements in reliability in recent years, it is considered less reliable than Linux. Linux is notoriously reliable and secure and it is free from constant battling viruses and malware (which may affect your desktops, laptops, and servers by corrupting files, causing slow downs, crashes, costly repairs and taking over basic functions of your operating system) – and it keeps you free from licensing fees i.e., zero cost of entry ... as in free. You can install Linux on as many reliable computer ecosystems on the planet as you like without paying a cent for software or server licensing. While Microsoft Windows usually costs between \$99.00 and \$199.00 USD for each licensed copy and fear of losing data.

Below are some examples of where Linux is being used today:

- Android phones and tablets
- Servers
- TV, Cameras, DVD players, etc.
- Amazon
- Google
- U.S. Postal service
- New York Stock Exchange



Linux Operating System has primarily three components:

- **Kernel**

Kernel is the core part of Linux Operating System and interacts directly with hardware. It is responsible for all major activities of the Linux operating system.

- **System Library**

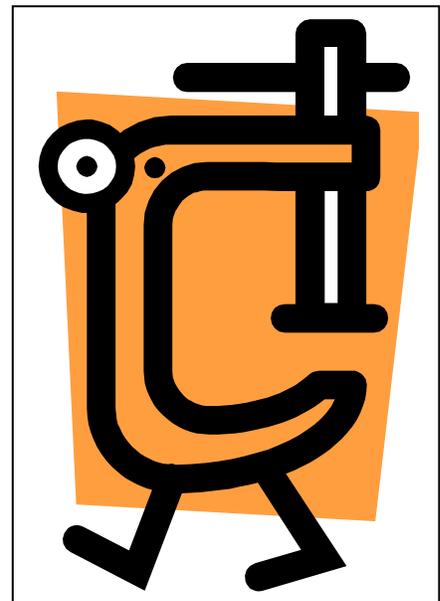
System libraries are special programs using which application programs access Kernel's features.

- **System Utility**

System Utility programs are responsible to do specialized tasks.

Important features of Linux Operating System:

- Portable
- Open Source
- Multi-User
- Multiprogramming
- Hierarchical File System
- Security



Now Linux (successfully being used by several millions of users worldwide) has grown passed the stage where it was almost exclusively an academic system, useful only to a handful of people with a technical background. It provides more than the operating system: there is an entire infrastructure supporting the chain of effort of creating an operating system, of making and testing programs for it, of bringing everything to the users, of supplying maintenance, updates and support and customizations, runs on different platforms including the Intel and Alpha platform. Today, Linux is ready to accept the challenge of a fast-changing world to do various

types of operations, call application programs etc. Since the hiring focus is shifting more and more toward DevOps type skills, a Linux skill set will be the types of things that will make you very deployable.

```
[manju@localhost ~]$ echo al{an,bert,exander}
```

```
alan albert alexander
```

```
[manju@localhost ~]$ mkdir {txt,doc,pdf}files
```

```
[manju@localhost ~]$ ls
```

```
txtfiles docfiles pdffiles
```

```
[manju@localhost ~]$ x=Albert; y="$x won \$100.00"; echo $y
```

```
Albert won $100.00
```

```
[manju@localhost ~]$ x=5; test $x -eq 10; echo $?
```

```
1 exit status of the test command is 1
```

```
[manju@localhost ~]$ x=5; test $x -eq 5; echo $?
```

```
0 exit status of the test command is 0
```

```
cat /etc/profile
```

```
# Display System login initialization file
```

```
cat /etc/bashrc
```

```
# Display System BASH shell configuration file
```

```
cat .bash_profile
```

```
# Display Login initialization file
```

```
[manju@localhost ~]$ date
```

```
Wed Sep 28 08:14:17 PDT 2022
```

```
[manju@localhost ~]$ alias x=date
```

```
[manju@localhost ~]$ x
```

```
Wed Sep 28 08:14:27 PDT 2022
```

```
[manju@localhost ~]$ echo $BASH_VERSION
```

```
4.2.46(1)-release
```

Display the current BASH version number

```
[manju@localhost ~]$ echo $HISTCMD
```

```
290
```

Display the number of the current command in the history list

```
[manju@localhost ~]$ echo $HOSTTYPE
```

```
x86_64
```

Display the type of machine the host runs on

```
[manju@localhost ~]$ echo $OSTYPE
```

```
linux-gnu
```

Display the operating system in use

```
[manju@localhost ~]$ echo $PPID
```

```
3563
```

Display the process ID for shell's parent shell

```
[manju@localhost ~]$ echo $SHLVL
```

```
2
```

Display the current shell level

```
[manju@localhost ~]$ echo $TERM
```

```
xterm-256color
```

Display the terminal type

```
[manju@localhost ~]$ echo $EUID
```

```
1000
```

Display the Effective user ID

```

[manju@localhost ~]$ PS1="\d" ←
Sun Oct 02 # Display the Current date _____

[manju@localhost ~]$ PS1="\h" ←
localhost # Display the Hostname _____

[manju@localhost ~]$ PS1="\s" ←
bash # Display the Shell type currently active _____

[manju@localhost ~]$ PS1="\t" ←
18:42:10 # Display the Time of day in hours, minutes, and seconds _____

[manju@localhost ~]$ PS1="\u" ←
manju # Display the Username _____

[manju@localhost ~]$ PS1="\v" ←
4.2 # Display the Shell version _____

```

PS1="\w"	Display the full pathname of the current working directory
PS1="\W"	Display the name of the current working directory

```

[manju@localhost ~]$ PS1="Hello\n World"

Hello

World

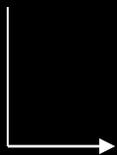
[manju@localhost ~]$ PS1="Hello \\ World"

Hello \ World

```

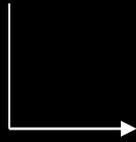
Directory	Function
/	The top-level directory of a Linux system that holds all files, device information, and system information organized into directories
/home	Holds users ' home directories
/bin	Contains every essential command and utility program
/usr	Contains the commands and files that the system uses
/usr/bin	Include utility programs and user-friendly commands
/usr/sbin	Holds commands for system administration
/usr/lib	Contains programming language libraries
/usr/share/doc	Contains documentation for Linux
/usr/share/man	Contains the online " man " files
/var/spool	Contains spooled files, such as those produced for network transfers and printing operations
/sbin	Contains commands for system administration used to boot the system
/var	Holds a variety of files, including mailbox files
/dev	Holds file interfaces for devices like printers and terminals
/etc	Holds all system files, including configuration files

```
find myfiles -name '*.c' -ls
```



Using the **-ls** command, all files in the "**myfiles**" directory with the **.c extension** are searched and displayed

```
find / -user manju -print
```



Finds every file in a user's home directory and every file that user owns in other user directories

```
ls /usr/share/X11
```

```
# List the system X11 configuration and support files
```

```
ls /etc/X11
```

```
# List the configuration files
```

```
ls /etc/gdm
```

```
# Display the contents of GDM configuration directory
```

```
ls /usr/share/gdm
```

```
# Display the contents of GDM configuration directory for default settings and themes
```

```
ls /etc/gconf
```

```
# List the GConf configuration files
```

```
ls /usr/share/gnome
```

```
# List the Files used by GNOME applications
```

```
ls /usr/share/doc/gnome*
```

```
# Display the contents of Documentation for various GNOME packages, including libraries
```

```
ls /usr/share/icons
```

```
# List the Icons used in KDE desktop and applications
```

```
rpm -qa | more
```

```
# Displays a list of all installed packages
```

```
ls /etc/cron.d
```

```
# List the directory with numerous crontab files that is only accessible to the root user
```

```
ls /etc/cron.hourly
```

```
# List the directory for tasks performed hourly
```

```
ls /etc/cron.daily
```

```
# List the directory for tasks performed daily
```

```
ls /etc/cron.weekly
```

```
# List the directory for tasks performed weekly
```

```
ls /etc/cron.monthly
```

```
# List the directory for tasks performed monthly
```

```
ls /etc/mtab
```

```
# List the currently mounted file systems
```

```
ls /etc/services
```

```
# List the services run on the system and the ports they use
```



```
ls /etc/cups
# List the CUPS printer configuration files
```

```
ls /proc/net
# List the Directory for network devices
```

```
free -s 3
# Display the current usage status of Memory continuously after regular interval
```

```
ls -lhR /var | grep \- | grep [1-9]*M
# List "/var" files larger than 1 MB but less than 1 GB
```

```
whereis -b ls
# Search only the binary file related to a command "ls"
```

```
whereis -m ls
# Searches only for man pages related to a command "ls"
```

```
whereis -s ls
# Searches only for source files related to a command "ls"
```

```
[manju@localhost ~]$ echo "Alan" "Mathison" "Turing"
```

```
Alan Mathison Turing
```

```
watch -t -n 1 date

# Display the date

[manju@localhost ~]$ echo "Albert" > 1.txt && cat 1.txt

Albert
```

```
du -sh * --time

# Check each file's size and the date and time it was last edited

dmidecode -s system-serial-number

# Display the serial number of Linux server
```

```
ls -aril

# Display all the files with sequence number

yum search mod_

# Display all the modules
```

```
du -sch *

# Display the sum of size of all files and folders in present directory

dmidecode | grep -A3 '^System Information'

# Display the server hardware name and model

dmesg | grep -i firmware

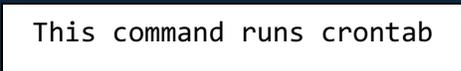
# Display all firmware error
```

```
cat /proc/cpuinfo | grep processor | wc -l  
  
# Display the number of cores  
  
netstat -ap | grep 80  
  
# Display the process id which is using port number 80
```

```
dmidecode --type memory  
  
# Display the physical memory attached to the Server  
  
dirs  
  
# Display the list of currently remembered directories
```

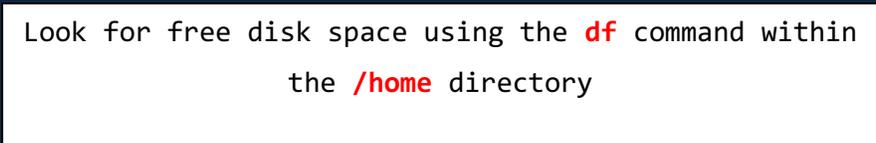
```
blkid -i /dev/sda  
  
# Display information about available block devices
```

```
crontab -e
```



This command runs crontab

```
[manju@localhost ~]$ df -h /home  
  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/sda3       18G  5.2G   13G  29% /
```



Look for free disk space using the **df** command within the **/home** directory

```
[manju@localhost ~]$ hostname -I ←
192.168.6.131 192.168.122.1
```

Display all local IP addresses of the host

```
badblocks -s /dev/sda
```

```
# Check for unreadable blocks on disk sda
```

```
tail -10 /var/log/messages
```

```
# Display the last 10 syslog messages
```

```
lsof -u manju
```

```
# List files opened by the user "manju"
```

```
sudo shutdown -r 2
```

```
# Shuts down and reboots the machine in 2 minutes
```

```
[manju@localhost ~]$ cat 1.txt
```

```
albert
```

```
[manju@localhost ~]$ cat 1.txt | tr a-z A-Z > 2.txt
```

```
[manju@localhost ~]$ cat 2.txt
```

```
ALBERT
```

```
cat /etc/passwd | column -t -s :
```

```
# Display the contents of "/etc/passwd" in column
```

```
nmcli d
# Display the status of all network interfaces

grep "^[:alnum:]" myfiles.txt
# Search for a line which will start with alphanumeric characters in "myfiles.txt"

grep "^[:alpha:]" myfiles.txt
# Search for a line which will start with alpha characters in "myfiles.txt"

grep "^[:blank:]" myfiles.txt
# Search for a line which will start with blank characters in "myfiles.txt"

grep "^[:digit:]" myfiles.txt
# Search for a line which will start with digit characters in "myfiles.txt"
```

```
grep "^[:lower:]" myfiles.txt
# Search for a line which will start with lowercase letters in "myfiles.txt"

grep "^[:punct:]" myfiles.txt
# Search for a line which will start with punctuation characters in "myfiles.txt"

grep "^[:graph:]" myfiles.txt
# Search for a line which will start with graphical characters in "myfiles.txt"

grep "^[:print:]" myfiles.txt
# Search for a line which will start with printable characters in "myfiles.txt"
```

```
grep "^[:space:]" myfiles.txt
```

```
# Search for a line which will start with space characters in "myfiles.txt"
```

```
grep "^[:upper:]" myfiles.txt
```

```
# Search for a line which will start with uppercase letters in "myfiles.txt"
```

```
grep "^[:xdigit:]" myfiles.txt
```

```
# Search for a line which will start with hexadecimal digits in "myfiles.txt"
```

```
vmstat -a
```

```
# Display active and inactive system memory
```

```
vmstat -s
```

```
# Display memory and scheduling statistics
```

```
vmstat -f
```

```
# Display number of forks created since system boot
```

```
vmstat -D
```

```
# Display a quick summary statistic of all disk activity
```

```
vmstat -d
```

```
# Display a detailed statistic on each disk usage
```

```
vmstat 5 -S M →
```

This command is used to update the statistics every five seconds and change the display units to megabytes

```
[manju@localhost ~]$ free -h --total
```

	total	used	free	shared	buff/cache	available
Mem:	976M	566M	75M	8.7M	334M	209M
Swap:	2.0G	84K	2.0G			
Total:	3.0G	566M	2.1G			

- hostname -s
- hostname --short

} Display the short version of the hostname

```
hostname --all-ip-addresses
```

```
# Display All Network Addresses
```

```
date -r /etc/hosts
```

```
# Display Last Modified Timestamp of a Date File
```

```
[manju@localhost ~]$ cat 1.txt
```

```
Albert Einstein
```

```
[manju@localhost ~]$ cat 2.txt
```

```
Elsa Einstein
```

```
[manju@localhost ~]$ cat 1.txt > 2.txt
```

```
[manju@localhost ~]$ cat 2.txt
```

```
Albert Einstein
```

```
[manju@localhost ~]$ cat 12.txt
```

```
Albert Einstein
```

```
Elsa Einstein
```

```
[manju@localhost ~]$ cat -n 12.txt
```

```
1 Albert Einstein
```

```
2 Elsa Einstein
```

```
[manju@localhost ~]$ cat 1.txt
```

```
Albert Einstein
```

```
[manju@localhost ~]$ cat -e 1.txt
```

```
Albert Einstein$
```

```
sudo shutdown 08:00
```

```
# Shutdown the system at 8 AM in the morning
```

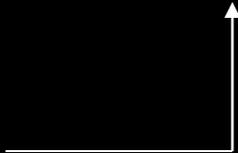
```
grep 'but\|is' phy.txt
```

```
# Search for the words "but" and "is" in the phy.txt file
```

```
grep 'is\|but\|of' phy.txt
```

```
# Search for the words "but", "is" and "of" in the phy.txt file
```

```
grep -e but -e is -e of phy.txt
```




```
echo "The system will be shutdown in 10 minutes." | wall
```



The message (The system will be shutdown in 10 minutes.) will be broadcasted to all users that are currently logged in

```
[manju@localhost ~]$ echo -e 'Albert Einstein'
```

```
Albert Einstein
```

```
[manju@localhost ~]$ echo -e 'Albert \c Einstein'
```

```
Albert [manju@localhost ~]$
```

```
ss --all
```

```
# List all listening and non-listening connections
```

```
ss --listen
```

```
# List only listening sockets
```

```
ss -t state listening
```

```
# Find all listening TCP connections
```

```
[manju@localhost ~]$ hostname -I | awk '{print $1}'
```

```
192.168.6.131
```



System's IP address

```
yum erase httpd
```

```
# Uninstall apache
```

- read has the value of 4
- write has the value of 2
- execute has the value of 1
- no permission has the value of 0

```
chmod 644 1.txt
```

- User: 6 = 4 + 2 (read and write)
- Group: 4 = 4 + 0 + 0 (read)
- Others: 4 = 4 + 0 + 0 (read)

- 7 = 4 + 2 + 1 (read, write and execute)
- 6 = 4 + 2 + 0 (read and write)
- 5 = 4 + 0 + 1 (read and execute)
- 4 = 4 + 0 + 0 (read)

```
rpm -qi httpd
```

```
# Display information about a particular package (apache)
```

```
sudo rpm -qa | wc -l
```

```
# Display the total number of packages installed
```

```
sudo repoquery -a --installed
```

```
# List all installed packages with the repoquery command
```

```
cat /var/log/boot.log
```

```
# Display all information related to booting operations
```

```
cat /var/log/maillog
```

```
# Display all information related to mail servers and archiving emails
```

```
cat /var/log/yum.log
```

```
# Display Yum command logs
```

```
mkdir -m777 myfiles
```

```
# Create a directory "myfiles" with read, write and execute permissions
```

```
rpm -qa centos-release
```

```
# Display CentOS version
```

```
ps -ALFH
```

```
# Get information about threads (LWP and NLWP)
```

- ps -eM
- ps axZ

Get Security Information of Linux Process

```
ps -auxf | sort -nr -k 4 | head -10
```

```
# Display the top 10 memory consuming process
```

```
ps -auxf | sort -nr -k 3 | head -10

# Display the top 10 CPU consuming process

sar -n DEV | more

# Monitor, collect and report Linux system activity
```

```
# create or overwrite "1.txt" file
echo "Albert Einstein" > 1.txt

# create or append to "1.txt" file
echo "Albert Einstein" >> 1.txt
```

```
grep -i "is" phy.txt

# Search for a given string in a file "phy.txt"

grep -A 3 -i "is" phy.txt

# Print the matched line and the following three lines

grep -r "is" *

# Recursively look for a given string in all files

export | grep ORACLE

# Display oracle related environment variables
```

```
chkconfig --list | grep network
```

```
# View the startup configuration of Linux network service
```

```
shutdown -r 18:30
```

```
# Shutdown the system immediately and reboot at 18:30
```

```
find /home -size +1024 -print
```

```
# Find files above 1MB in home directory
```

```
find /home -size +1024 -size -4096 -print
```

```
# Find files above 1Mb and below 4MB in home directory
```

```
netstat -ain
```

```
# Display the Kernel Interface table
```

```
sar -n SOCK | more
```

```
# Display networking Statistics
```

```
find /home -size +10000k
```

```
# Find files greater than 10000k in the home directory
```

```
ls -ld /home
```

```
# List information about the home directory instead of its contents
```

```
chmod go+=r 1.txt
```

```
# Add read permission for the owner and the group
```

```
chown manju 1.txt
```

```
# Change ownership of a file "1.txt" to user "manju"
```

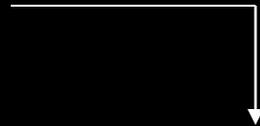
```
du -sh *
```

```
# Display the disk usages of the files in the current directory
```

```
du -sh .[!.*] *
```

```
# Display the disk usages of the files (including hidden files) in the current directory
```

```
du -sch .[!.*] *
```



Display the total disk usage of the files
(including hidden files) in the current directory

```
du --threshold=1G -sh .[!.*] *
```



Display only files with more than 1GB in size which
located under current directory

```
iostat -kx
```

```
# Display general information about the disk operations in real time
```

```
netstat -ntlp
```

```
# Display open TCP sockets
```

```
netstat -nulp
```

```
# Display open UDP sockets
```

```
netstat -nxlp
```

```
# Display open Unix sockets
```

Parted is a well-known command line tool that allows us to easily manage hard disk partitions

```
sudo yum install parted
```

```
# Install parted
```

```
parted -v
```

```
# Check Parted version
```

```
parted -l
```

```
# Lists partition layout on all block devices
```

```
parted -m
```

```
# Displays machine parseable output
```

```
dmidecode -q | less
```

```
# Display BIOS information
```

```
systemctl --failed
```

```
# List failed services
```

```
losetup
```

```
# Display information about all loop devices
```

```
quit
```

```
# Exit the parted shell
```

- `getfacl --access 1.txt`
- `getfacl -a 1.txt`

Display the file access control list of a file "1.txt"

- `getfacl -n 1.txt`
- `getfacl --numeric 1.txt`

List the numeric user and group IDs w.r.t file "1.txt"

```
sudo tcpdump -D
```

```
# List of all available network interfaces in the system
```

```
[manju@localhost ~]$ xz myfile.txt
```

```
[manju@localhost ~]$ ls | grep myfile
```

```
myfile
```

```
myfile.txt.xz
```

Compress a file "myfile.txt" using xz command

```
[manju@localhost ~]$ free -t | awk 'NR == 2 {print $3/$2*100}'
```

```
61.852
```

```
[manju@localhost ~]$ free -t | awk 'FNR == 2 {print $3/$2*100}'
```

```
61.852
```

Display Memory Utilization


```
[manju@localhost ~]$ free -t | awk 'NR == 3 {print $3/$2*100}'
```

```
2.54155
```

```
[manju@localhost ~]$ free -t | awk 'FNR == 3 {print $3/$2*100}'
```

```
2.54155
```

Display Swap Utilization

```
[manju@localhost ~]$ free -t | awk 'FNR == 2 {printf("%.2f% \n"), $3/$2*100}'
```

```
61.86%
```

```
[manju@localhost ~]$ free -t | awk 'NR == 2 {printf("%.2f% \n"), $3/$2*100}'
```

```
61.86%
```

Display Memory Utilization with Percent Symbol and two decimal places

```
[manju@localhost ~]$ free -t | awk 'FNR == 3 {printf("%.2f% \n"), $3/$2*100}'
```

```
2.65%
```

```
[manju@localhost ~]$ free -t | awk 'NR == 3 {printf("%.2f% \n"), $3/$2*100}'
```

```
2.65%
```

Display Swap Utilization with Percent Symbol and two decimal places

```
[manju@localhost ~]$ top -b -n1 | grep ^%Cpu | awk '{cpu+=$9}END{print 100-cpu/NR}'
```

```
100
```

Display CPU Utilization

```
[manju@localhost ~]$ top -b -n1 | grep ^%Cpu | awk '{cpu+=$9}END{printf("%.2f% \n"), 100-cpu/NR}'
```

```
100.00%
```

Display CPU Utilization with Percent Symbol and two decimal places

```
swapon -s
```

```
# Print swap usage summaries
```

```
swapon -a
```

```
# Activate all of swap space
```

```
swapoff -a
```

```
# Deactivate all of swap space
```

```
[manju@localhost ~]$ cat /etc/system-release
```

```
CentOS Linux release 7.3.1611 (Core)
```

```
alias -p
```

```
# List all Aliases
```

```
lsof -i :8080
```

```
# Check which process is running on port 8080
```

Display the version of CentOS

```
sudo netstat -anp | grep tcp | grep LISTEN

# Display the various in-use ports and the process using it

sudo netstat -anp | grep 8080

# Display the process listening on port 8080
```

```
printf "%s\n" *

# Prints the files and directories that are in the current directory

printf "%s\n" */

# Prints only the directories in the current directory

printf "%s\n" *.{gif,jpg,png}

# Lists only some image files
```

```
[manju@localhost ~]$ alias x='date' # create an alias

[manju@localhost ~]$ x # preview the alias

Fri Oct 7 03:51:39 PDT 2022

[manju@localhost ~]$ unalias x # remove the alias

[manju@localhost ~]$ x

bash: x: command not found...
```

```
[manju@localhost ~]$ x="alan"; printf '%s\n' "${x^}"
```

Alan

```
[manju@localhost ~]$ x="alan"; printf '%s\n' "${x^^}"
```

ALAN

```
[manju@localhost ~]$ x="alan"; declare -u name="$x"; echo "$name"
```

ALAN

```
find . -name "xyz[a-z][0-9]"
```

Find directories and files with names starting with "xyz" and ending with an alpha character after a one-digit

```
find . -mmin -120
```

Search for files changed during the previous two hours

```
find . -mmin +120
```

Search for files that haven't been updated in the past two hours

```
find . -mtime -3
```

Find files that have been modified within the last 3 days

```
find . -mtime +3
```

Find files that have not been modified within the last 3 days

```
[manju@localhost ~]$ names="Albert Alan John Mary"; x=(${names// / }); echo ${x[0]}
```

Albert

```
[manju@localhost ~]$ names="Albert Alan John Mary"; x=(${names// / }); echo ${x[3]}
```

Mary

```
names="Albert+Alan+John+Mary";
```

```
x=(${names//+/ });
```

```
echo ${x[0]}
```

```
# Output: Albert
```

```
x=(hello world); echo "${x[@]/#/A}"
```

```
# Output: Ahello Aworld
```

```
names="Albert+Alan+John+Mary";
```

```
x=(${names//+/ });
```

```
echo ${x[3]}
```

```
# Output: Mary
```

```
[manju@localhost ~]$ awk '{print $2}' <<< "Alan Mathison Turing"
```

Mathison

```
[manju@localhost ~]$ awk '{print $1}' <<< "Alan Mathison Turing"
```

Alan

```
x='4 * 2'; echo "$x"
```

```
# prints 4 * 2
```

```
x='4 * 2'; echo $x
```

```
# prints 4, the list of files in the current directory, and 2
```

```
x='4 * 2'; echo "$(($x))"
```

```
# prints 8
```

```
[manju@localhost ~]$ x="ALAN"; printf '%s\n' "${x,}"
```

```
aLAN
```

```
[manju@localhost ~]$ x="ALAN"; printf '%s\n' "${x,,}"
```

```
alan
```

```
[manju@localhost ~]$ x="Alan"; echo "${x~~}"
```

```
aLAN
```

```
[manju@localhost ~]$ x="Alan"; echo "${x~}"
```

```
alan
```

```
[manju@localhost ~]$ x='You are a genius'; echo "${x/a/A}"
```

```
You Are a genius
```

```
[manju@localhost ~]$ x='You are a genius'; echo "${x//a/A}"
```

```
You Are A genius
```

```
[manju@localhost ~]$ x='You are a genius'; echo "${x/%s/N}"
```

```
You are a geniN
```

```
[manju@localhost ~]$ x='You are a genius'; echo "${x/s/}"
```

```
You are a geni
```

```
[manju@localhost ~]$ x='You are a genius'; echo "${x#*a}"
```

re a genius

```
[manju@localhost ~]$ x='You are a genius'; echo "${x#*g}"
```

enius

```
[manju@localhost ~]$ foo=25; i=foo; echo ${i}
```

foo

```
[manju@localhost ~]$ foo=25; i=foo; echo ${!i}
```

25

```
[manju@localhost ~]$ x='You are a genius'; echo "${x%a*}"
```

You are

```
[manju@localhost ~]$ x='You are a genius'; echo "${x%%a*}"
```

You

```
[manju@localhost ~]$ x=Bob-Dev-Fox; echo ${x%%-*}
```

Bob

```
[manju@localhost ~]$ x=Bob-Dev-Fox; echo ${x%-*}
```

Bob-Dev

```
[manju@localhost ~]$ x=Bob-Dev-Fox; echo ${x##*-}
```

Fox

```
[manju@localhost ~]$ x=Bob-Dev-Fox; echo ${x#*-}
```

Dev-Fox

```
find . -type f -path '*/Documents/*'
```

```
# Find only files within a folder called Documents
```

```
find . -type f -path '*/Documents/*' -o -path '*/Downloads/*'
```

```
# Find only files within a folder called Documents or Downloads
```

```
find . -type f -not -path '*/Documents/*'
```

```
# Find all files except the ones contained in a folder called Documents
```

```
find . -type f -not -path '*log' -not -path '*/Documents/*'
```

```
# Find all files except the ones contained in a folder called Documents or log files
```

```
[manju@localhost ~]$ find /dev -type b
```

```
/dev/sr0  
/dev/sda3  
/dev/sda2  
/dev/sda1  
/dev/sda
```

Block devices

```
[manju@localhost ~]$ echo '16 / 5' | bc
```

```
3
```

```
[manju@localhost ~]$ echo '16 / 5' | bc -l
```

```
3.20000000000000000000
```

```
find . -maxdepth 1 -type f -name "*.txt"
```

```
# Find every.txt file from the current directory alone
```

```
[manju@localhost ~]$ echo "$(printf "%04d" "${x}")"
```

```
0000
```



```
[manju@localhost ~]$ echo "$(printf "%05d" "${x}")"
```

```
00000
```

```
[manju@localhost ~]$ echo "\"'\\""
```

```
"'"
```

```
[manju@localhost ~]$ echo '3 5 + p' | dc
```

```
8
```

```
[manju@localhost ~]$ dc <<< '3 5 + p'
```

```
8
```

```
[manju@localhost ~]$ echo '3 5 * p' | dc
```

```
15
```

```
[manju@localhost ~]$ dc <<< '3 5 * p'
```

```
15
```

```
[manju@localhost ~]$ expr 'Alan Turing' : 'Ala\(.*\)ring'
```

```
n Tu
```

```
[manju@localhost ~]$ echo '12 == 12 && 18 > 12' | bc
```

```
1 (True)
```

```
[manju@localhost ~]$ echo '12 == 13 && 18 > 12' | bc
```

```
0 (False)
```

```
[manju@localhost ~]$ expr PQRSTUVWXYZ : PQRS
```

```
4
```

Display the number of matching characters

```
ls -ral
```

```
# Listing of all files in reverse alphabetical order
```

- `ls -tl`
- `ls -trl`

```
# List the files such that the one that was most recently edited is at the top of the list
```

```
find . -regex ".*\(\.sh\|\.txt\) $"
```

```
# Find .sh or .txt files
```

```
[manju@localhost ~]$ find . -iregex ".*\(\.sh\|\.pdf\) $"
```

```
./bc.pdf
```

```
./1.PDF
```

```
./data.sh
```

```
./1.sh
```

```
./2.SH
```

```
./1.pdf
```

```
./2.sh
```

```
find . -type f -print
```

```
# List only regular files
```

```
[manju@localhost ~]$ echo "alan+alan+alan+alan" | xargs -d +
```

```
alan alan alan alan
```

```
[manju@localhost ~]$ echo "alan+alan+alan+alan" | xargs -d + -n 2
```

```
alan alan
```

```
alan alan
```

```
[manju@localhost ~]$ echo -e "2\nalbert\n" > 1.txt
```

```
[manju@localhost ~]$ cat 1.txt
```

```
2
```

```
albert
```

- `ps -eLf --sort -nlwp | head`
- `ps -eLf`

Display information about process threads



"Linux is a superbly polished copy of an antique - shinier than the original, perhaps, but still defined by it."

— Jaron Lanier, *You Are Not a Gadget*

Best Linux Books that Every Superuser Should Read:

- **How Linux Works: What Every Superuser Should Know**

Book by Brian Ward

- **The Linux Programming Interface**

Book by Michael Kerrisk

- **Linux pocket guide**

Book by Daniel J. Barrett

- **Linux for Beginners**

Book by Jason Cannon

- **How Linux Works: What Every Superuser Should Know**

Book by Brian Ward

- **Linux Kernel Development**

Book by Robert Love

- **Linux: The Complete Reference**

Book by Richard Petersen

- **Linux in a Nutshell**

Book by Ellen Siever and Robert Love

- **Linux Basics for Hackers: Getting Started with Networking, Scripting, and Security in Kali**

Book by OccupyTheWeb

- **Linux Command Line and Shell Scripting Bible**
Book by Christine Bresnahan and Richard BLUM
- **Linux Administration: The Linux Operating System and Command Line Guide for Linux Administrators**
Book by Jason Cannon
- **The Art of Unix Programming**
Book by Eric S. Raymond
- **The Linux Command Line, 2nd Edition: A Complete Introduction**
Book by William Shotts
- **Linux Bible**
Book by Christopher Negus
- **Linux System Programming: Talking Directly to the Kernel and C Library**
Book by Robert Love
- **A Practical Guide to Linux Commands, Editors, and Shell Programming**
Book by Mark G. Sobell
- **Linux for Beginners and Command Line Kung Fu**
Book by Jason Cannon
- **Linux Device Drivers**
Book by Alessandro Rubini, Greg Kroah-Hartman, and Jonathan Corbet
- **Advanced Linux programming**
Book by Alex Samuel, Jeffrey Oldham, and Mark Mitchell
- **Understanding the Linux Kernel**

Book by Daniel Pierre Bovet and Marco Cesati

- **Learn Linux Quickly: A Beginner-friendly Guide to Getting Up and Running with the World's Most Powerful Operating System**

Book by Ahmed Alkabary

- **Linux administration**

Book by Wale Soyinka

- **Linux For Dummies**

Book by Richard Blum

- **Linux Essentials**

Book by Christine Bresnahan and Richard BLUM

- **The Linux Command Line Beginner's Guide**

Book by Jonathan Moeller

- **Linux All-in-One for Dummies**

Book by Emmett Dulaney

- **Learning the bash Shell**

Book by Cameron Newham

- **Linux for Developers: Jumpstart Your Linux Programming Skills**

Book by William "Bo" Rothwell

- **Lfm: Linux Field Manual**

Book by Tim Bryant

- **CompTIA Linux+ Study Guide: Exam XK0-005**

Book by Christine Bresnahan and Richard BLUM

- **sed & awk**

Book by Arnold Robbins and Dale Dougherty

- **Linux From Scratch**

Book by Gerard Beekmans



Linux is a complex example of the wisdom of crowds. It's a good example in the sense that it shows you can set people to work in a decentralized way - that is, without anyone really directing their efforts in a **particular direction** - and still trust that they're going to come up with good answers.

- James Surowiecki

One final thought:

If you feel that this information has been useful to you, please take a moment to share it with your friends on LinkedIn, Facebook and Twitter. Consider writing a brief review on **Google Play Books** if you feel that this book has helped you in your Linux Admin career and you have learned something worthwhile.

In the big view, I believe Linux to be incredibly amazing. This is a fantastic platform that serves as a neutral, dependable hub for developers and enterprise organizations to build, manage, and scale open technology projects and ecosystems. I want to spread my passion to as many individuals as I can. I also hope that this is not the end of your learning.

Thank you!

Did I miss any commands? Let me know! Email me manjunath5496@gmail.com





ARE YOU A NEW GNU/LINUX USER WHO WANTS TO LEARN MORE THAN JUST DRAGGING WINDOWS AROUND AND CLICKING YOUR MOUSE? THIS BOOK IS DESIGNED FOR USERS WHO ARE COMPLETELY NEW TO THE GNU/LINUX COMMAND LINE AND WANT TO LEARN THE FUNDAMENTALS. IN THIS BOOK, BASIC LINUX COMMANDS FOR NAVIGATION, FILE AND DIRECTORY ADMINISTRATION, SOFTWARE INSTALLATION, AND ELEMENTARY SYSTEM TROUBLESHOOTING ARE COVERED. YOU WILL DISCOVER HOW TO USE LINUX COMMANDS IN THIS BOOK. OVER THE YEARS, LINUX HAS SEEN SIGNIFICANT TRANSFORMATION AND IS CURRENTLY REGARDED AS ONE OF THE TOP OPERATING SYSTEMS IN THE WORLD. AS A SYSTEM ADMINISTRATOR, YOU MAINTAIN THE FUNCTIONALITY OF THE GLOBAL COMPUTING INFRASTRUCTURE. PROBLEMS MUST BE FIXED, SYSTEMS MUST BE MAINTAINED, AND SECURITY MUST BE MAINTAINED AT ALL TIMES. I HOPE THESE COMMANDS WILL BE USEFUL AND ENABLE YOU TO PERFORM YOUR WORK MORE EFFECTIVELY.





This free edition was downloaded from
www.obooko.com

No charge has been made to access or download this book.
You may freely use this digital copy in accordance with the terms and
conditions of the license attributed to the work.