

St. Francis Institute of Technology, Mumbai-400 103  
**Department Of Information Technology**

A.Y. 2024-2025  
Class: TE-ITA/B, Semester: V

Subject: **DevOps Lab**

**Experiment – 1: To understand DevOps: Principles, practices and DevOps  
Engineer role & responsibilities and learn basic Linux commands**

1. **Aim:** To prepare case study on DevOps and execute Linux commands
2. **Objectives:** After study of this experiment, the students will be able to
  - Understand the fundamentals of DevOps engineering and be fully proficient with DevOps terminologies, concepts, benefits and deployment options to meet business requirements
  - Demonstrate execution of Linux commands
3. **Outcomes:** After study of this experiment, the students will be able to
  - Understand the importance of DevOps thoroughly
  - Demonstrate the Linux commands
4. **Prerequisite:** Knowledge of software engineering and project management basic principles
5. **Requirements:** Personal Computer, Windows operating system, Ubuntu Operating system or Oracle virtual box or any online terminal, Internet Connection, Microsoft Word.
6. **Pre-Experiment Exercise:**  
**Brief Theory:** Refer shared material
7. **Laboratory Exercise**
  - a. **Procedure:**
    - a. Prepare case study based on following:
      - Introduction
      - Need
      - Usage
      - DevOps Lifecycle
      - Advantages and disadvantages
    - b. **Enlist following Linux commands with their explanation and execution screenshots**
      1. sudo
      2. Apt -get
      3. Ls
      4. Cd
      5. Ped
      6. Cp
      7. Mv
      8. Rm
      9. Mkdir
      10. History
      11. Df
      12. Du
      13. Free
      14. Uname -a
      15. Top
      16. Man
      17. Info

18. Passwd
19. Whatis
20. Date
21. W
22. Exit
23. Shutdown
24. Head
25. Tail
26. Echo
27. Grep
28. Zip
29. Unzip
30. <command name> -h or <command name> --help

## 8. Post-Experiments Exercise

### A. Extended Theory:

Nil

### B. Questions:

- What is Linux and basic components of Linux?
- What are the shells used in Linux?
- What are the top 10 devops tools that are used in the industry today?
- How will you approach when a project needs to implement devops?

### C. Conclusion:

- Write what was performed in the experiment.
- Write the significance of the topic studied in the experiment.

### D. References:

- <https://www.edureka.co/blog/interview-questions/linux-interview-questions-for-beginners/>
- <https://www.softwaretestinghelp.com/devops-interview-question11>
- <https://techlog360.com/basic-ubuntu-commands-terminal-shortcuts-linux-beginner/>
- <https://tutorials.ubuntu.com/tutorial/command-line-for-beginners#0>
- <https://techlog360.com/basic-ubuntu-commands-terminal-shortcuts-linux-beginner/>
- <https://www.edureka.co/blog/top-10-devops-tools/>
- <https://www.guru99.com/devops-tutorial.html>

## Laboratory Exercise:

### a. Prepare case study

#### CASE STUDY

##### Introduction:

DevOps is a transformative approach that integrates software development (Dev) and IT operations (Ops) to improve collaboration, efficiency, and quality in the software development lifecycle. This case study examines the necessity, usage, lifecycle, advantages, and disadvantages of DevOps in modern IT environments.

##### Need:

The demand for DevOps arises from several pressing challenges and goals:

1. **Rapid Delivery:** Businesses need to deliver software updates and new features quickly to remain competitive.
2. **Improved Collaboration:** Traditional silos between development and operations teams can lead to inefficiencies and misunderstandings.
3. **Quality and Reliability:** Consistent and automated testing and deployment processes are essential to ensure high-quality software.
4. **Scalability:** As businesses grow, their IT infrastructure needs to scale efficiently without compromising performance.
5. **Customer Satisfaction:** Fast and reliable updates improve user experience and satisfaction.

##### Usage:

DevOps is used across various industries to streamline operations and enhance productivity. Key usage areas include:

1. **Continuous Integration and Continuous Deployment (CI/CD):** Automating the process of integrating code changes and deploying them to production.
2. **Infrastructure as Code (IaC):** Managing and provisioning computing infrastructure through machine-readable scripts.
3. **Monitoring and Logging:** Continuously monitoring applications and infrastructure to detect and resolve issues proactively.
4. **Collaboration Tools:** Utilizing platforms like Slack, Jira, and Confluence to improve communication and project management.

## DevOps Lifecycle

The DevOps lifecycle encompasses several stages that ensure continuous delivery and improvement:

1. **Plan:** Define project requirements, set objectives, and plan sprints or release cycles.
2. **Code:** Write and review code collaboratively, ensuring adherence to best practices.
3. **Build:** Compile code into executable files, often involving automated testing.
4. **Test:** Conduct automated and manual tests to identify bugs and ensure quality.
5. **Release:** Deploy code to production environments, often using automated pipelines.
6. **Deploy:** Manage deployment processes to different environments (staging, production).
7. **Operate:** Maintain and monitor the system in production to ensure it runs smoothly.
8. **Monitor:** Continuously observe application performance and user feedback to identify areas for improvement.

## Advantages

1. **Faster Time to Market:** Accelerated development cycles and automated deployments lead to quicker releases.
2. **Improved Collaboration:** Breaking down silos fosters better teamwork and communication between Dev and Ops teams.
3. **Higher Quality:** Automated testing and continuous monitoring improve software reliability and performance.
4. **Scalability:** Automated processes and IaC allow for easy scaling of applications and infrastructure.
5. **Cost Efficiency:** Reducing manual processes and improving resource utilization can lower operational costs.

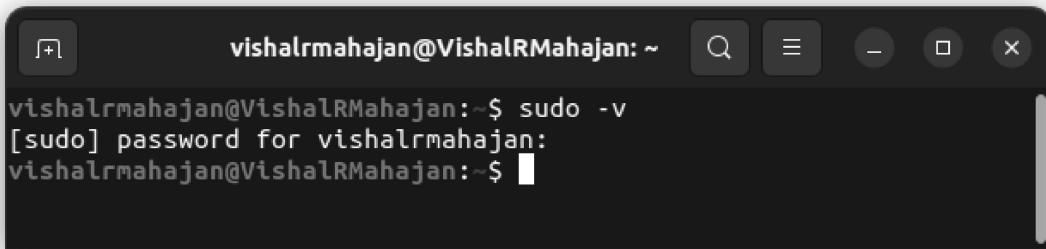
## Disadvantages

1. **Cultural Shift:** Implementing DevOps requires a significant cultural change, which can be challenging for some organizations.
2. **Complexity:** Integrating various tools and practices can be complex and require substantial expertise.
3. **Security Concerns:** Automated processes and rapid releases may introduce security vulnerabilities if not managed properly.
4. **Initial Investment:** Setting up DevOps practices and tools can be costly and resource-intensive initially.
5. **Resistance to Change:** Some team members may resist the shift to DevOps, preferring traditional methods.

**b. Enlist following Linux commands with their explanation and execution screenshots**

1. **sudo** : Allows a permitted user to execute a command as the superuser or another user, as specified by the security policy.

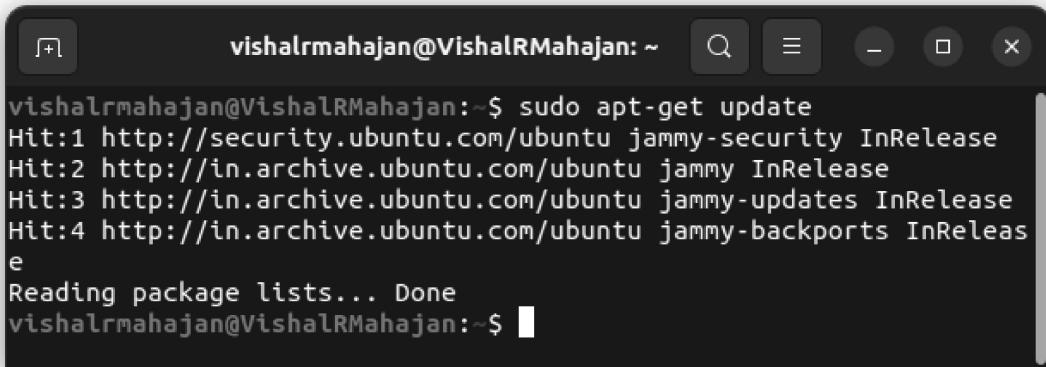
- **Usage:** sudo [options] command
- **Example:** sudo apt-get update



A screenshot of a terminal window titled "vishalrmahajan@VishalRMahajan: ~". The window shows the command "sudo -v" being run, followed by a password prompt: "[sudo] password for vishalrmahajan:". The terminal has a dark background with light-colored text and standard window controls at the top.

2. **apt-get** : A command-line tool for handling packages in Ubuntu. It is used to install, remove, upgrade, and manage software packages.

- **Usage:** apt-get [options] command
- **Common Commands and Examples:**
  - Update the package list:
    - sudo apt-get update
  - Upgrade all installed packages:
    - sudo apt-get upgrade



A screenshot of a terminal window titled "vishalrmahajan@VishalRMahajan: ~". The window shows the command "sudo apt-get update" being run. The output includes several "Hit" messages indicating files are being downloaded from repositories, followed by the message "Reading package lists... Done". The terminal has a dark background with light-colored text and standard window controls at the top.

### 3. Ls : Lists directory contents.

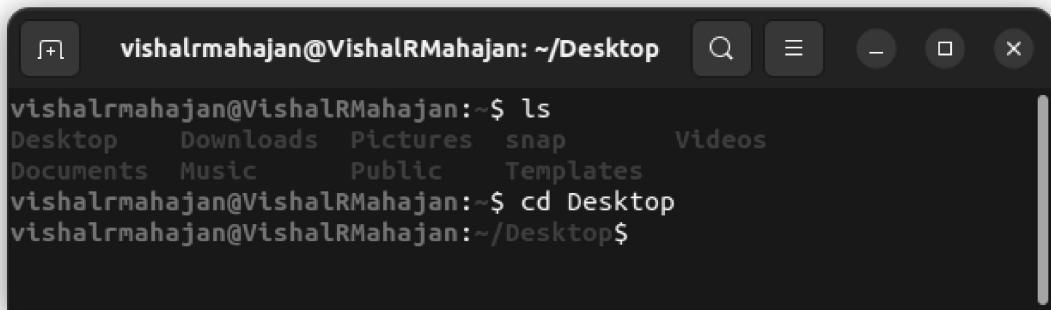
- **Usage:** ls [options] [file|directory]
- **Example:** ls -l /home



```
vishalrmahajan@VishalRMahajan:~$ ls
Desktop Downloads Pictures snap      Videos
Documents Music     Public   Templates
vishalrmahajan@VishalRMahajan:~$
```

### 4. cd: Changes the current directory.

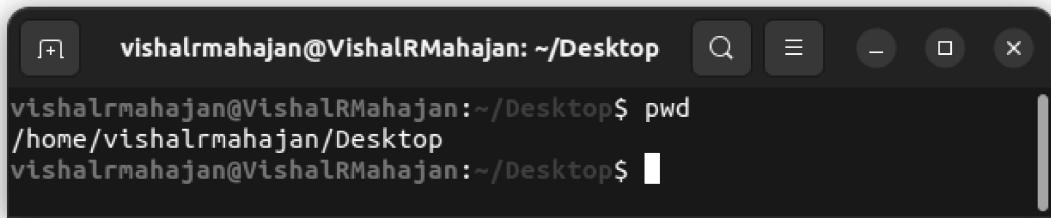
- **Usage:** cd [directory]
- **Example:** cd /home/user



```
vishalrmahajan@VishalRMahajan:~/Desktop$ ls
Desktop Downloads Pictures snap      Videos
Documents Music     Public   Templates
vishalrmahajan@VishalRMahajan:~/Desktop$ cd Desktop
vishalrmahajan@VishalRMahajan:~/Desktop$
```

### 5. Pwd : Prints the current working directory.

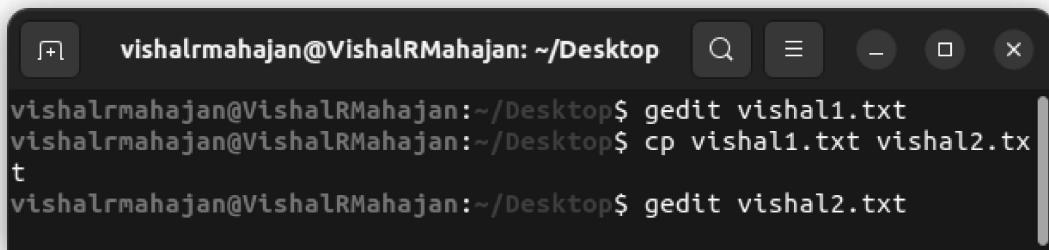
- **Usage:** pwd
- **Example:** pwd



```
vishalrmahajan@VishalRMahajan:~/Desktop$ pwd
/home/vishalrmahajan/Desktop
vishalrmahajan@VishalRMahajan:~/Desktop$
```

## 6. Cp : Copies files and directories.

- **Usage:** cp [options] source destination
- **Example:** cp file1.txt /home/user/



A screenshot of a terminal window titled "vishalrmahajan@VishalRMahajan: ~/Desktop". The window shows the following command sequence:  
vishalrmahajan@VishalRMahajan:~/Desktop\$ gedit vishal1.txt  
vishalrmahajan@VishalRMahajan:~/Desktop\$ cp vishal1.txt vishal2.txt  
vishalrmahajan@VishalRMahajan:~/Desktop\$ gedit vishal2.txt



7. **Mv** : Moves or renames files and directories.

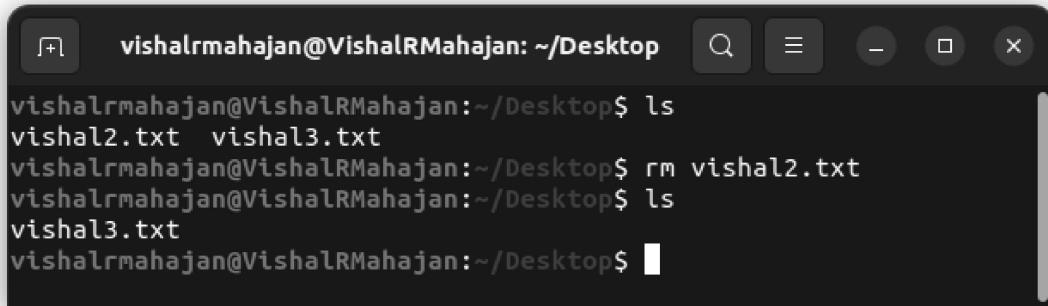
- **Usage:** mv [options] source destination
- **Example:** mv file1.txt file2.txt



```
vishalrmahajan@VishalRMahajan: ~/Desktop$ gedit vishal1.txt
vishalrmahajan@VishalRMahajan: ~/Desktop$ mv vishal1.txt vishal3.txt
vishalrmahajan@VishalRMahajan: ~/Desktop$ gedit vishal3.txt
vishalrmahajan@VishalRMahajan: ~/Desktop$ gedit vishal1.txt
vishalrmahajan@VishalRMahajan: ~/Desktop$
```

**8. Rm** : Removes files or directories.

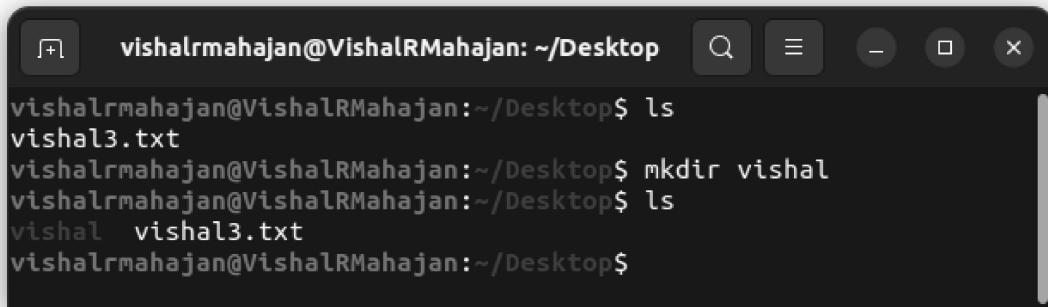
- **Usage:** rm [options] file
- **Example:** rm file1.txt



```
vishalrmahajan@VishalRMahajan: ~/Desktop$ ls  
vishal2.txt vishal3.txt  
vishalrmahajan@VishalRMahajan: ~/Desktop$ rm vishal2.txt  
vishalrmahajan@VishalRMahajan: ~/Desktop$ ls  
vishal3.txt  
vishalrmahajan@VishalRMahajan: ~/Desktop$
```

**9. Mkdir** : Creates directories.

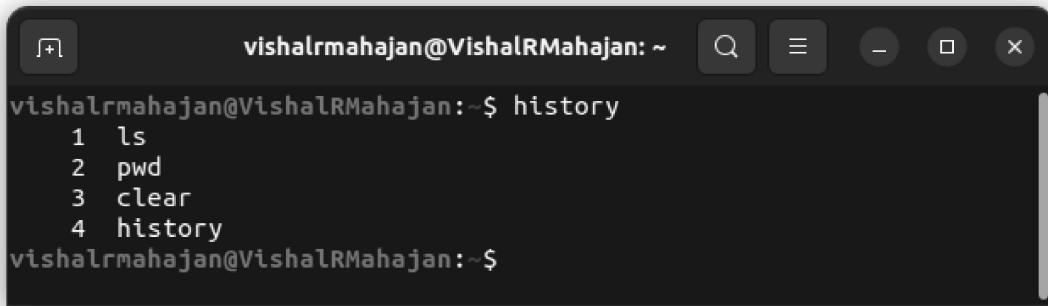
- **Usage:** mkdir [options] directory
- **Example:** mkdir new\_folder



```
vishalrmahajan@VishalRMahajan: ~/Desktop$ ls  
vishal3.txt  
vishalrmahajan@VishalRMahajan: ~/Desktop$ mkdir vishal  
vishalrmahajan@VishalRMahajan: ~/Desktop$ ls  
vishal vishal3.txt  
vishalrmahajan@VishalRMahajan: ~/Desktop$
```

**10. History** : Shows the command history.

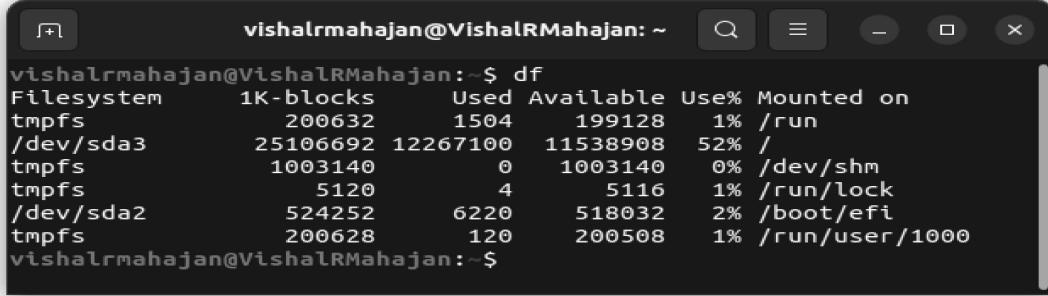
- **Usage:** history [options]
- **Example:** history



```
vishalrmahajan@VishalRMahajan: ~$ history  
1 ls  
2 pwd  
3 clear  
4 history  
vishalrmahajan@VishalRMahajan: ~$
```

**11. Df** : Reports file system disk space usage.

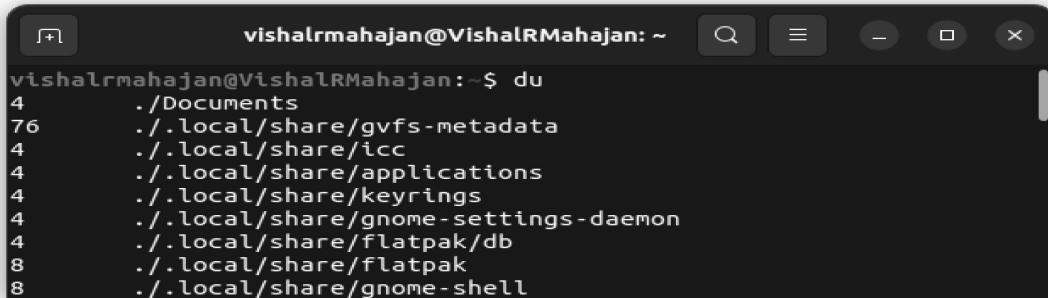
- **Usage:** df [options] [file]
- **Example:** df -h



```
vishalrmahajan@VishalRMahajan:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
tmpfs             200632     1504    199128   1% /run
/dev/sda3       25106692 12267100   11538908  52% /
tmpfs            1003140        0   1003140   0% /dev/shm
tmpfs              5120        4     5116   1% /run/lock
/dev/sda2       524252     6220     518032   2% /boot/efi
tmpfs             200628     120     200508   1% /run/user/1000
vishalrmahajan@VishalRMahajan:~$
```

## 12. Du : Estimates file space usage.

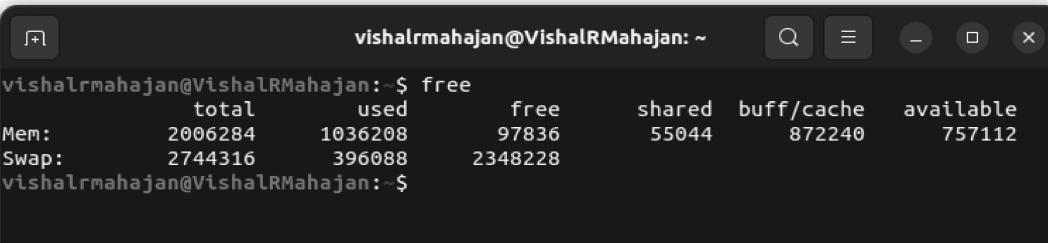
- **Usage:** du [options] [file|directory]
- **Example:** du -sh /home/user



```
vishalrmahajan@VishalRMahajan:~$ du
4      ./Documents
76     ./local/share/gvfs-metadata
4      ./local/share/icc
4      ./local/share/applications
4      ./local/share/keyrings
4      ./local/share/gnome-settings-daemon
4      ./local/share/flatpak/db
8      ./local/share/flatpak
8      ./local/share/gnome-shell
```

## 13. Free : Displays the amount of free and used memory in the system.

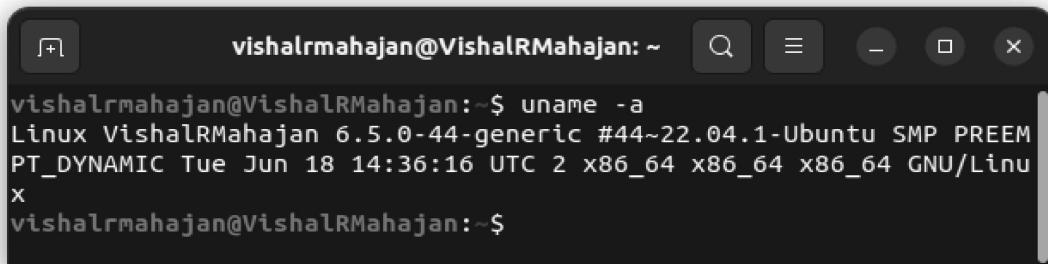
- **Usage:** free [options]
- **Example:** free -h



```
vishalrmahajan@VishalRMahajan:~$ free
total        used         free      shared  buff/cache   available
Mem:      2006284      1036208      97836      55044      872240      757112
Swap:     2744316      396088     2348228
vishalrmahajan@VishalRMahajan:~$
```

## 14. Uname -a : Prints all system information.

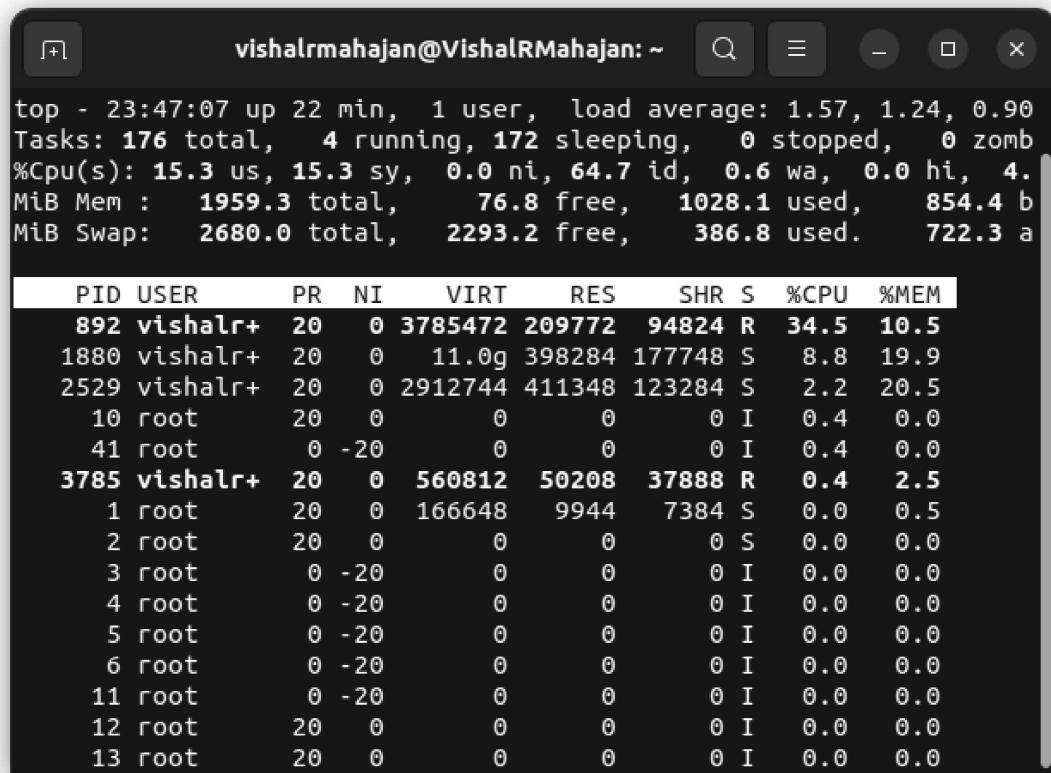
- **Usage:** uname -a
- **Example:** uname -a



```
vishalrmahajan@VishalRMahajan:~$ uname -a
Linux VishalRMahajan 6.5.0-44-generic #44~22.04.1-Ubuntu SMP PREEM
PT_DYNAMIC Tue Jun 18 14:36:16 UTC 2 x86_64 x86_64 x86_64 GNU/Linu
x
vishalrmahajan@VishalRMahajan:~$
```

**15. Top :** Displays tasks and system resource usage.

- **Usage:** top
- **Example:** top



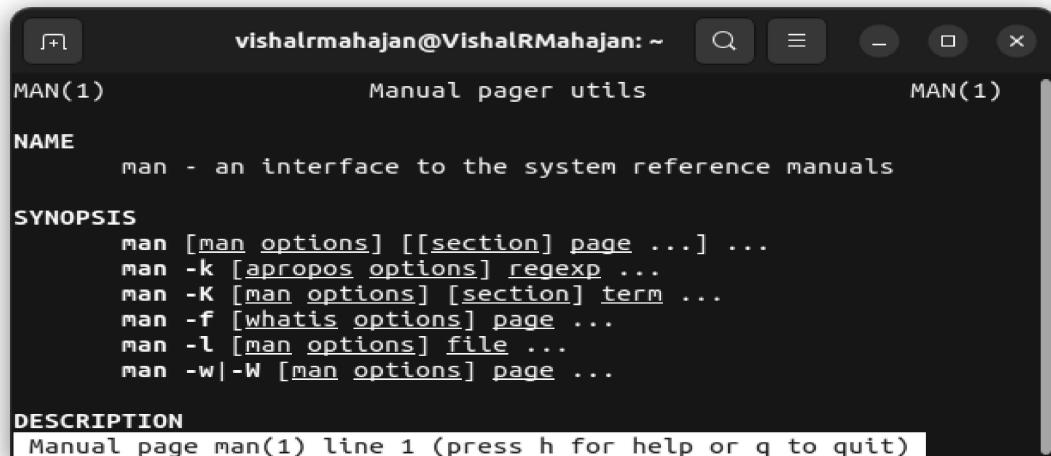
vishalrmahajan@VishalRMahajan: ~

```
top - 23:47:07 up 22 min, 1 user, load average: 1.57, 1.24, 0.90
Tasks: 176 total, 4 running, 172 sleeping, 0 stopped, 0 zomb
%Cpu(s): 15.3 us, 15.3 sy, 0.0 ni, 64.7 id, 0.6 wa, 0.0 hi, 4.
MiB Mem : 1959.3 total, 76.8 free, 1028.1 used, 854.4 b
MiB Swap: 2680.0 total, 2293.2 free, 386.8 used. 722.3 a

PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM
 892 vishalr+  20   0 3785472 209772  94824 R 34.5 10.5
1880 vishalr+  20   0 11.0g 398284 177748 S 8.8 19.9
2529 vishalr+  20   0 2912744 411348 123284 S 2.2 20.5
 10 root      20   0      0      0      0 I 0.4 0.0
 41 root      0 -20      0      0      0 I 0.4 0.0
 3785 vishalr+  20   0 560812 50208 37888 R 0.4 2.5
  1 root      20   0 166648  9944  7384 S 0.0 0.5
  2 root      20   0      0      0      0 S 0.0 0.0
  3 root      0 -20      0      0      0 I 0.0 0.0
  4 root      0 -20      0      0      0 I 0.0 0.0
  5 root      0 -20      0      0      0 I 0.0 0.0
  6 root      0 -20      0      0      0 I 0.0 0.0
 11 root      0 -20      0      0      0 I 0.0 0.0
 12 root      20   0      0      0      0 I 0.0 0.0
 13 root      20   0      0      0      0 I 0.0 0.0
```

**16. Man :** Displays the manual for a command.

- **Usage:** man [command]
- **Example:** man man



vishalrmahajan@VishalRMahajan: ~

```
MAN(1)          Manual pager utils          MAN(1)

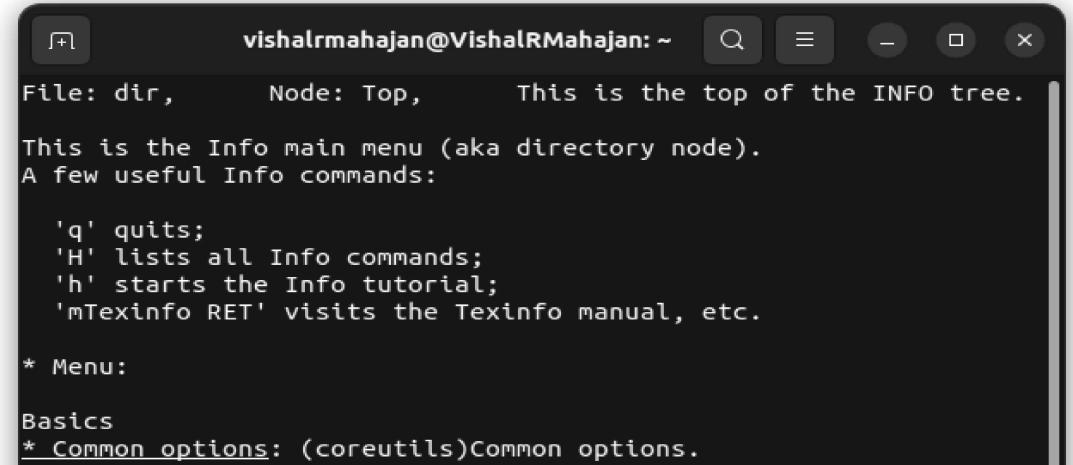
NAME
man - an interface to the system reference manuals

SYNOPSIS
man [man options] [[section] page ...] ...
man -k [apropos options] regexp ...
man -K [man options] [section] term ...
man -f [whatis options] page ...
man -l [man options] file ...
man -w|-W [man options] page ...

DESCRIPTION
Manual page man(1) line 1 (press h for help or q to quit)
```

**17. Info:** Displays the info documentation for a command.

- **Usage:** info [command]
- **Example:** info ls



vishalrmahajan@VishalRMahajan: ~

```
File: dir,      Node: Top,      This is the top of the INFO tree.

This is the Info main menu (aka directory node).
A few useful Info commands:

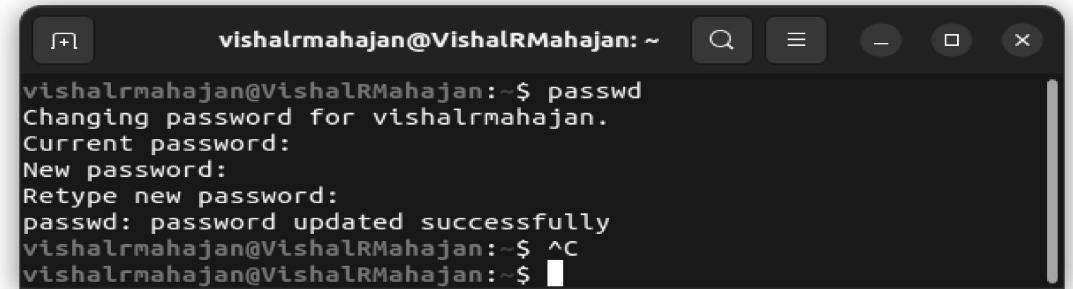
'q' quits;
'H' lists all Info commands;
'h' starts the Info tutorial;
'mTexinfo RET' visits the Texinfo manual, etc.

* Menu:

Basics
* Common options: (coreutils)Common options.
```

**18. Passwd :** Changes the user password.

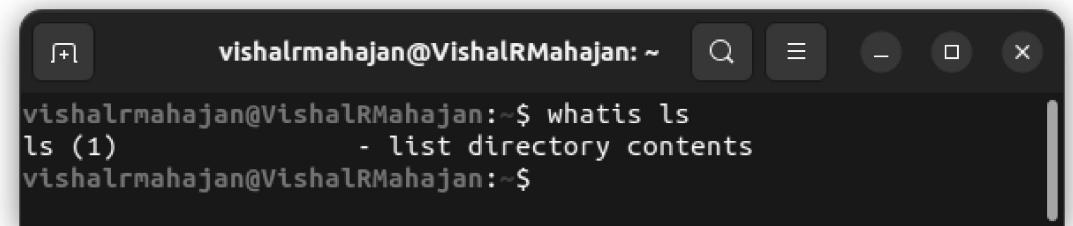
- **Usage:** passwd [options] [username]
- **Example:** passwd



```
vishalrmahajan@VishalRMahajan: ~$ passwd
Changing password for vishalrmahajan.
Current password:
New password:
Retype new password:
passwd: password updated successfully
vishalrmahajan@VishalRMahajan: ~$ █
vishalrmahajan@VishalRMahajan: ~$ █
```

**19. Whatis :** Displays a brief description of a command.

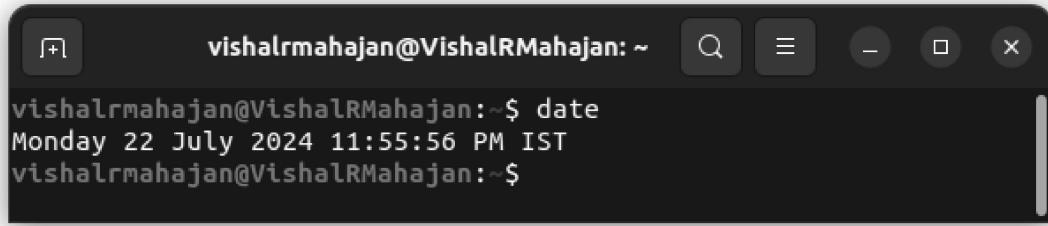
- **Usage:** whatis [command]
- **Example:** whatis ls



```
vishalrmahajan@VishalRMahajan: ~$ whatis ls
ls (1)          - list directory contents
vishalrmahajan@VishalRMahajan: ~$
```

**20. Date :** Displays or sets the system date and time.

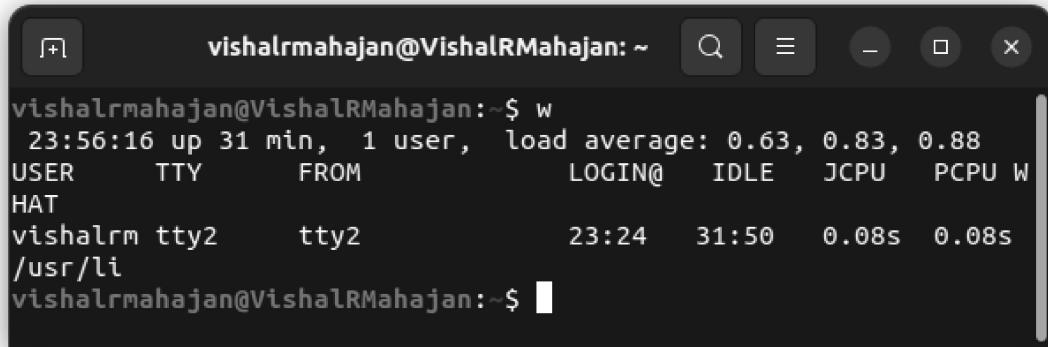
- **Usage:** date [options]
- **Example:** date



```
vishalrmahajan@VishalRMahajan: ~$ date
Monday 22 July 2024 11:55:56 PM IST
vishalrmahajan@VishalRMahajan: ~$
```

21. **W** : Shows who is logged on and what they are doing.

- **Usage:** w [options]
- **Example:** w



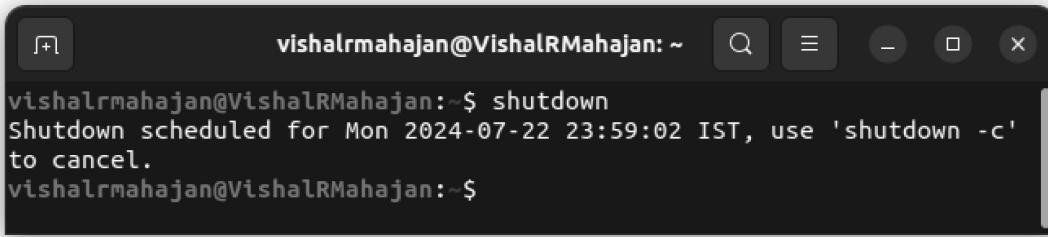
```
vishalrmahajan@VishalRMahajan: ~$ w
23:56:16 up 31 min,  1 user,  load average: 0.63, 0.83, 0.88
USER      TTY      FROM          LOGIN@    IDLE   JCPU   PCPU W
HAT
vishalrm  tty2      tty2          23:24    31:50   0.08s  0.08s
/usr/li
vishalrmahajan@VishalRMahajan: ~$
```

22. **Exit** : Exits the shell or terminates a session.

- **Usage:** exit
- **Example:** exit

23. **Shutdown**: Brings the system down.

- **Usage:** shutdown [options] [time] [message]
- **Example:** shutdown



```
vishalrmahajan@VishalRMahajan: ~$ shutdown
Shutdown scheduled for Mon 2024-07-22 23:59:02 IST, use 'shutdown -c'
to cancel.
vishalrmahajan@VishalRMahajan: ~$
```

**24. Head :** Outputs the first part of files.

- **Usage:** head [options] [file]
- **Example:** head -n 4 file.txt



A screenshot of a text editor window titled "vishal.txt". The content of the file is as follows:

```
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
10 10
```

The status bar at the bottom shows "Plain Text" and "Tab Width: 8". The cursor is positioned at the end of line 10.

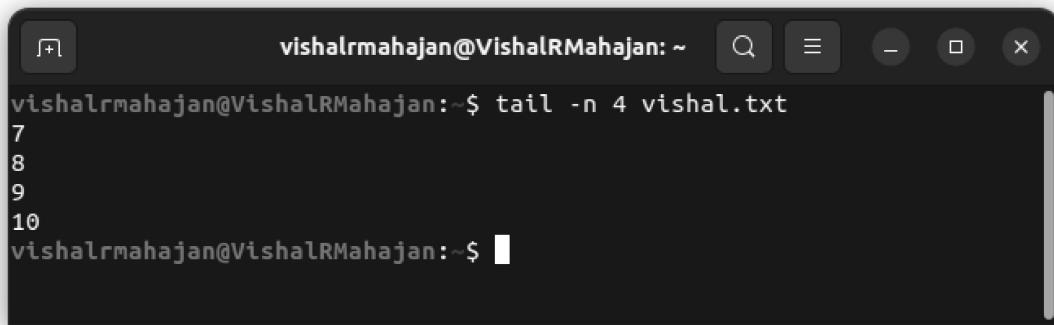


A screenshot of a terminal window. The user has run the command `head -n 4 vishal.txt`. The output is:

```
vishalrmahajan@VishalRMahajan:~$ gedit vishal.txt
vishalrmahajan@VishalRMahajan:~$ head -n 4 vishal.txt
1
2
3
4
vishalrmahajan@VishalRMahajan:~$
```

**25. Tail :** Outputs the last part of files.

- **Usage:** tail [options] [file]
- **Example:** tail -n 10 file.txt

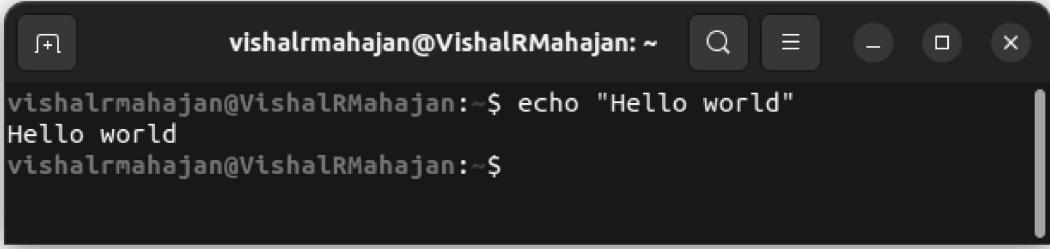


A screenshot of a terminal window. The user has run the command `tail -n 4 vishal.txt`. The output is:

```
vishalrmahajan@VishalRMahajan:~$ tail -n 4 vishal.txt
7
8
9
10
vishalrmahajan@VishalRMahajan:~$
```

## 26. Echo : Displays a line of text.

- **Usage:** echo [options] [string]
- **Example:** echo "Hello, World!"



```
vishalrmahajan@VishalRMahajan:~$ echo "Hello world"
Hello world
vishalrmahajan@VishalRMahajan:~$
```

## 27. Grep : Searches for patterns in files.

- **Usage:** grep [options] pattern [file]
- **Example:** grep "hello" file.txt



Open ▾ New vishal.txt ~/ Save ≡ - □ ×

```
1 Vishal
2 Rajesh
3 Mahajan
```

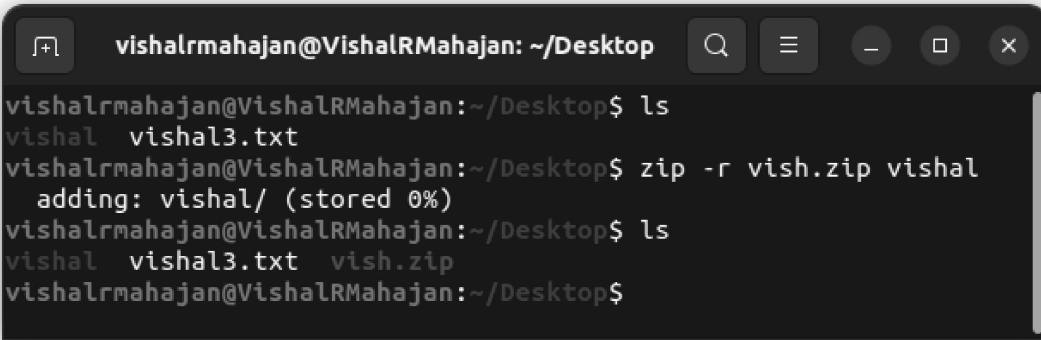
Saving File "/... Plain Text ▾ Tab Width: 8 ▾ Ln 3, Col 8 ▾ INS



```
vishalrmahajan@VishalRMahajan:~$ grep Vishal vishal.txt
Vishal
vishalrmahajan@VishalRMahajan:~$
```

## 28. Zip : Packages and compresses files.

- **Usage:** zip [options] output.zip file1 file2
- **Example:** zip archive.zip file1.txt file2.txt



```
vishalrmahajan@VishalRMahajan:~/Desktop$ ls
vishal  vishal3.txt
vishalrmahajan@VishalRMahajan:~/Desktop$ zip -r vish.zip vishal
    adding: vishal/ (stored 0%)
vishalrmahajan@VishalRMahajan:~/Desktop$ ls
vishal  vishal3.txt  vish.zip
vishalrmahajan@VishalRMahajan:~/Desktop$
```

**29. Unzip :** Extracts compressed files from a ZIP archive.

- **Usage:** unzip [options] zipfile
- **Example:** unzip archive.zip

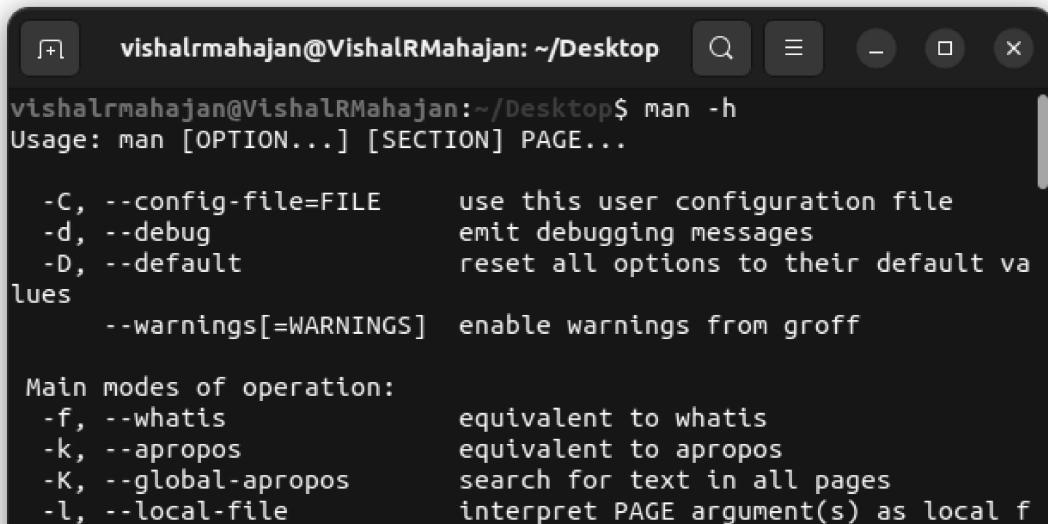


```
vishalrmahajan@VishalRMahajan: ~/Desktop$ unzip vish.zip
Archive: vish.zip
vishalrmahajan@VishalRMahajan: ~/Desktop$
```

A screenshot of a terminal window titled "vishalrmahajan@VishalRMahajan: ~/Desktop". The window shows the command "unzip vish.zip" being run, followed by the output "Archive: vish.zip". The window has standard OS X-style controls at the top.

**30. <command name> -h or <command name> --help:** Displays help information for a command.

- **Usage:** [command] -h or [command] --help
- **Example:** ls --help



```
vishalrmahajan@VishalRMahajan: ~/Desktop$ man -h
Usage: man [OPTION...] [SECTION] PAGE...

-C, --config-file=FILE      use this user configuration file
-d, --debug                 emit debugging messages
-D, --default               reset all options to their default va
lues
--warnings[=WARNINGS]      enable warnings from groff

Main modes of operation:
-f, --whatis                equivalent to whatis
-k, --apropos               equivalent to apropos
-K, --global-apropos        search for text in all pages
-l, --local-file             interpret PAGE argument(s) as local f
```

A screenshot of a terminal window titled "vishalrmahajan@VishalRMahajan: ~/Desktop". The window shows the command "man -h" being run, displaying the usage information for the man command. The window has standard OS X-style controls at the top.