ASSIGMENT – 01 :- [LINUX]
CYBER SECURITY

By

NAME :- **ganesh bharkade(div –b)**

PRN :- 46423000368

SUBMITED :- DR ANUP GRIDHAR SIR.

SUBJECT :- CYBER SECURITY

DEPARTMENT OF COMPUTER SCIENCE
TILAK MAHARASHTRA VIDYAPEETH PUNE

CERTIFICATE OF ORIGINALITY

I hereby certify that the project submitted under my name is ganesh bharkade the result of my independent efforts and creativity. I have diligently acknowledged all sources consulted and ensured the absence of any plagiarized content. Furthermore, I affirm that this project has not been previously submitted for publication elsewhere..

Name : ganesh bharkade

Allied Institution Tilak Maharashtr Vidyapeeth

Contact No: -9970271965

ACKNOWLEDGEMENT

I want to express my heartfelt thanks to Professor Anup Girdhar for their support and guidance during the research and writing of this project. Their expertise greatly improved the quality of my work.

I'm also grateful to my friends for their feedback and encouragement. Their input helped me refine my ideas.

Lastly, I want to thank my family for their unwavering support throughout this project. Their encouragement kept me motivated.

I'm thankful to everyone who contributed to the completion of this project.

Table of the Content

File Management:

Abstract:

This document explores the process of file management using a Unix-like operating system. Specifically, it outlines the steps involved in creating directories, generating files, copying, renaming, and moving files within a file system. The focus is on practical implementation, demonstrating commands and techniques commonly used in file management tasks.

Introduction:

File management is a fundamental aspect of computing, essential for organizing and accessing data efficiently. In this context, we delve into the Unix command line interface, a powerful tool for performing file operations. By understanding and mastering file management commands, users can navigate their file systems with ease and accomplish various tasks seamlessly.

Problem Statement:

While many users are familiar with basic file operations, such as creating directories and files, they may lack proficiency in more advanced tasks like copying, renaming, and moving files. This knowledge gap can hinder productivity and lead to inefficiencies in file organization and retrieval. Thus, there is a need to elucidate these concepts and provide practical guidance for users seeking to enhance their file management skills.

Objective:

The objective of this document is to elucidate the process of file management in a Unix-like environment through a series of practical examples. By following along with the outlined steps, readers will gain a deeper understanding of file manipulation commands and techniques. The ultimate goal is to empower users to leverage the full potential of their file systems and streamline their workflow.

Future Scope:

As technology evolves and computing environments become increasingly diverse, the principles of file management remain relevant. In the future, advancements in file systems and operating systems may introduce new features and capabilities, further enhancing the efficiency and usability of file management tools. Additionally, as users continue to generate and interact with vast amounts of data, the importance of effective file organization and navigation will only grow.

Conclusion:

In conclusion, file management is a foundational skill for computer users, enabling them to organize, access, and manipulate data effectively. By mastering the commands and techniques outlined in this document, users can enhance their productivity and streamline their workflow. Whether managing personal files or administering complex systems, a solid understanding of file management principles is invaluable.

1. Create a directory named "practice_files" and navigate

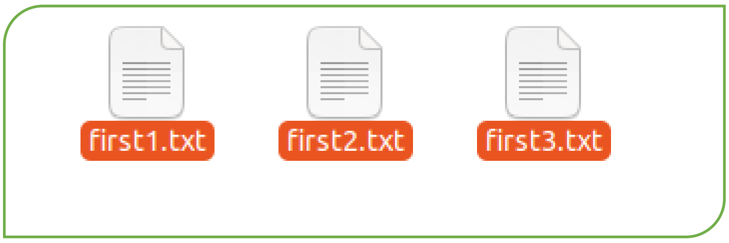
INTO IT.

```
mkdir practice file

cd practice_files
```

I. mkdir stands for "make directory."
II. It is used to create a new directory with the specified name.
III. cd stands for "change directory."
IV. It is used to move into the specified directory.

2. - Create files named “file1.txt” ,”file2.txt”and “file3.txt”

Output:-



I. cd is used to create empty files with the specified names.
II. Here, we are creating three files: "file1.txt," "file2.txt," and "file3.txt."

3 Copy "file1.txt" to "file4.txt" within the same directory:

```
cp file1.txt file4.txt
```

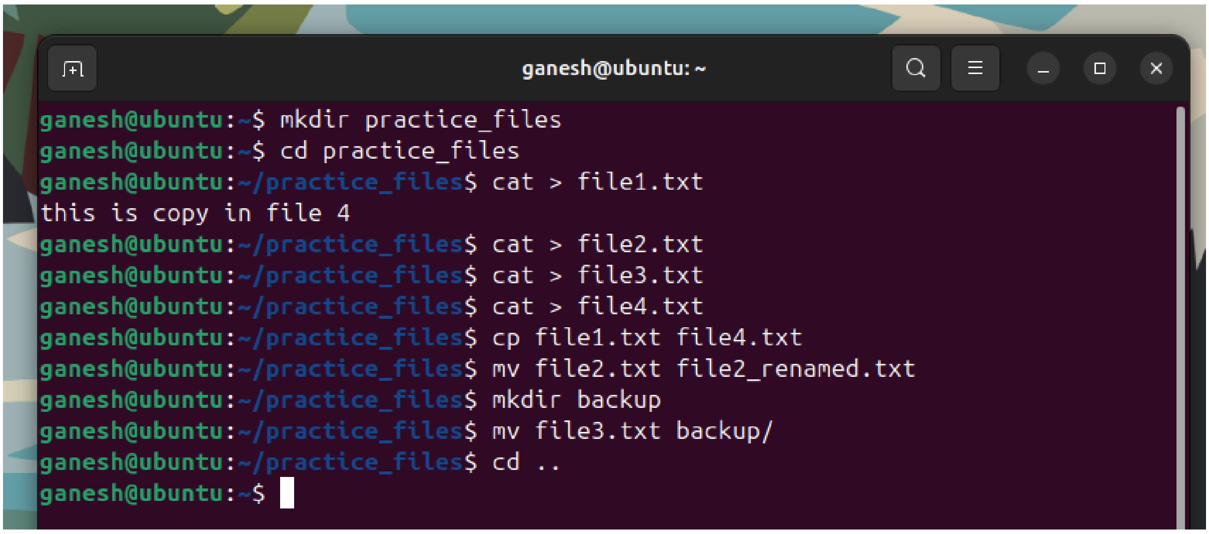
- cp stands for "copy."
- It is used to copy files or directories.
- In this case, it copies "file1.txt" and creates a new file named "file4.txt" with the same content as "file1.txt."

4.Rename "file2.txt" to "file2_renamed.txt":

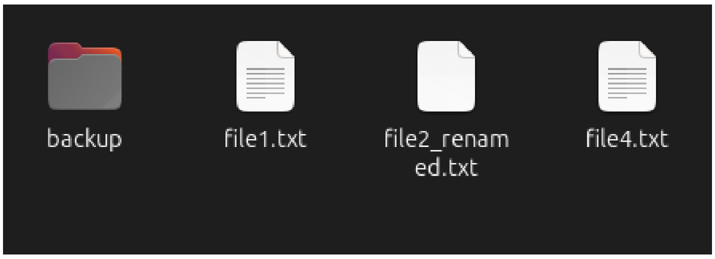
```
mv file2.txt file2_renamed.txt
```

I. mv stands for "move" but is commonly used for renaming as well.
II. It is used to move files or directories from one location to another.
III. In this case, it renames "file2.txt" to "file2_renamed.txt" within the same directory.

Output:-



```
ganesh@ubuntu:~$ mkdir practice_files
ganesh@ubuntu:~$ cd practice_files
ganesh@ubuntu:~/practice_files$ cat > file1.txt
this is copy in file 4
ganesh@ubuntu:~/practice_files$ cat > file2.txt
ganesh@ubuntu:~/practice_files$ cat > file3.txt
ganesh@ubuntu:~/practice_files$ cat > file4.txt
ganesh@ubuntu:~/practice_files$ cp file1.txt file4.txt
ganesh@ubuntu:~/practice_files$ mv file2.txt file2_renamed.txt
ganesh@ubuntu:~/practice_files$ mkdir backup
ganesh@ubuntu:~/practice_files$ mv file3.txt backup/
ganesh@ubuntu:~/practice_files$ cd ..
ganesh@ubuntu:~$ 
```



2. Permissions:

Abstract:

This document delves into the intricacies of file permissions within a Unix-like operating system environment. Specifically, it addresses the creation of a file named "secret.txt" and the implementation of restrictive access permissions to ensure that only the owner has read and write privileges. Furthermore, it explores the process of creating a new user named "guest" and testing whether this user can access "secret.txt".

Introduction:

File permissions are a crucial aspect of system security, governing who can access, modify, or execute files on a computer system. By properly configuring permissions, system administrators can enforce data privacy and integrity, mitigating the risk of unauthorized access or tampering. In this context, we focus on the creation of a file with restricted access and the testing of user permissions, providing insight into effective access control mechanisms.

Problem Statement:

The problem at hand is to create a file named "secret.txt" and restrict access to it so that only the owner can read and write to it. Additionally, we aim to create a new user named "guest" and assess whether this user can access "secret.txt". This scenario highlights the importance of implementing granular access controls to safeguard sensitive data from unauthorized access or manipulation.

Objective:

The primary objective of this document is to demonstrate the practical implementation of file permissions in a Unix-like environment. By following the outlined steps, readers will gain a comprehensive understanding of how to create files with restricted access and manage user permissions effectively. The ultimate goal is to promote best practices in access control and enhance system security.

Future Scope:

As technology continues to evolve, the landscape of file permissions and access control mechanisms will evolve accordingly. Future advancements may include enhancements to permission models, increased automation of access management processes, and integration with emerging technologies such as blockchain-based identity verification. Moreover, ongoing research in cybersecurity will inform the development of more robust security protocols to address evolving threats and vulnerabilities.

Conclusion:

In conclusion, effective management of file permissions is essential for maintaining the security and integrity of computer systems. By creating files with restricted access and carefully managing user permissions, organizations can mitigate the risk of data breaches and unauthorized access. However, it is imperative to stay abreast of evolving security threats and implement proactive measures to safeguard sensitive information.

a) Create a new file named “secret.txt”

- Cat>secret.txt

- Cd This command takes you to the user's home directory.
- The > symbol is used for output redirection.
- However, since the cd command doesn't produce any output, "secret.txt" is created as an empty file.
- The "secret.txt" file is created in the home directory but remains empty.

Output:-

b) Restrict access to "secret.txt" so that only the owner can read and write to it.

❖Chmod 600 secret.txt

- Here, chmod is the command to change file permissions. The 600 sets the file permissions to read and write for the owner (6 stands for read (4) + write (2)), and no permissions for group and others.

c) Create a new user named "guest" and test if they can access "secret.txt".

❖Sudo adduser guest

- This command adds a new user named "guest" to the system. You'll be prompted to set a password and provide additional information for the new user.

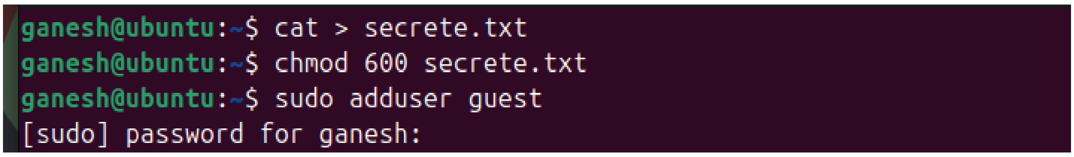
- sudo adduser guest: Adds a new user named "guest" to the system. You may need administrative privileges (sudo) to perform this action.

TEST

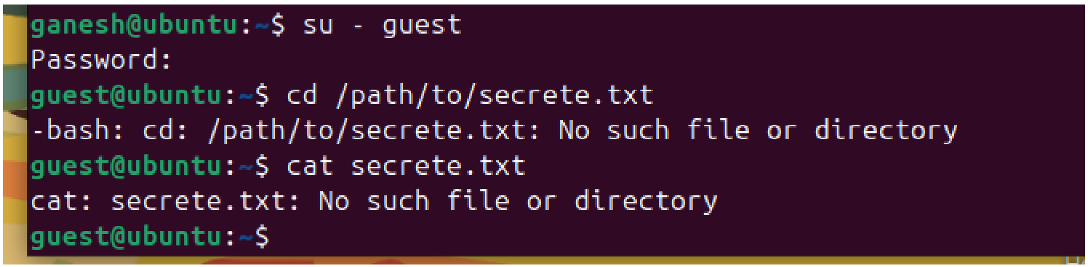
- Su guest
- Cat secret.txt

- The su guest command switches to the "guest" user, and cat secret.txt tries to display the contents of "secret.txt." Since the file has restrictive permissions (read and write only for the owner), the "guest" user should not be able to read the file.

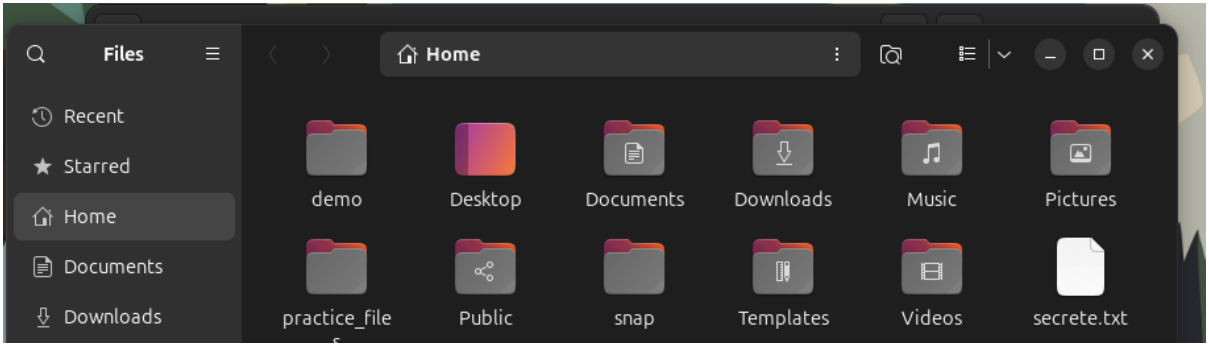
Output:-



```
ganesh@ubuntu:~$ cat > secrete.txt
ganesh@ubuntu:~$ chmod 600 secrete.txt
ganesh@ubuntu:~$ sudo adduser guest
[sudo] password for ganesh:
```



```
ganesh@ubuntu:~$ su - guest
Password:
guest@ubuntu:~$ cd /path/to/secrete.txt
-bash: cd: /path/to/secrete.txt: No such file or directory
guest@ubuntu:~$ cat secrete.txt
cat: secrete.txt: No such file or directory
guest@ubuntu:~$
```



:

3. Process Management

Abstract:

This document explores the management of processes within a Unix-like operating system environment. Specifically, it outlines the steps involved in checking running processes, identifying the Process ID (PID) of a specific process for termination, executing the termination command, and verifying the termination's success. The focus is on practical implementation, demonstrating commands and techniques commonly used in process management tasks.

Introduction:

Process management is a critical aspect of operating system administration, allowing users to monitor and control the execution of programs and services. In this context, we delve into the Unix command-line interface, a powerful tool for managing processes. By understanding and mastering process management commands, users can efficiently manage system resources and ensure the stability and performance of their systems.

Problem Statement:

While many users are familiar with basic process management tasks such as listing running processes, they may encounter difficulties when it comes to terminating specific processes. Identifying the correct process and executing termination commands accurately are essential to avoid unintended consequences, such as system instability or data loss. Thus, there is a need to elucidate these concepts and provide practical guidance for users seeking to enhance their process management skills.

Objective:

The objective of this document is to elucidate the process management tasks involved in checking running processes, identifying process IDs (PIDs), terminating specific processes, and confirming termination success. By following the outlined steps, readers will gain a deeper understanding of process management commands and techniques. The ultimate goal is to empower users to effectively manage processes and optimize system performance.

Future Scope:

As computing environments become increasingly complex and dynamic, the importance of efficient process management will only grow. Future advancements may include enhancements to process monitoring tools, automation of process termination based on predefined criteria, and integration with cloud computing platforms for centralized process management. Moreover, ongoing research in operating system design and performance optimization will inform the development of more sophisticated process management solutions.

Conclusion:

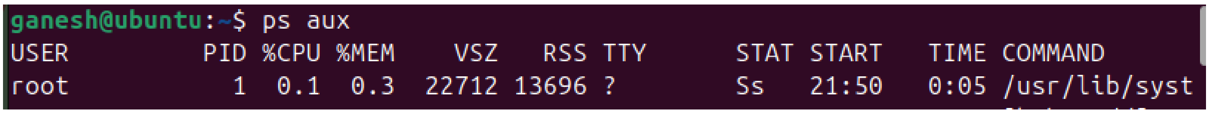
In conclusion, process management is a critical aspect of system administration, enabling users to monitor and control the execution of programs and services. By mastering process management commands and techniques, users can ensure the stability, security, and performance of their systems. However, it is essential to exercise caution when terminating processes to avoid unintended consequences. By following best practices and staying informed about the latest advancements in process management, users can effectively manage their systems and optimize resource utilization.

a) Check all running processes.

❖ Ps aux

- This command displays a detailed list of all running processes on your system, including the process ID (PID), the user associated with each process, the CPU and memory usage, and the command that started the process.

- Output:-



```
ganesh@ubuntu:~$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.1  0.3  22712 13696 ?        Ss   21:50   0:05 /usr/lib/syst
```

b.) Identify the PID (Process ID) of a process you want to terminate.

Terminate the process using its PID.

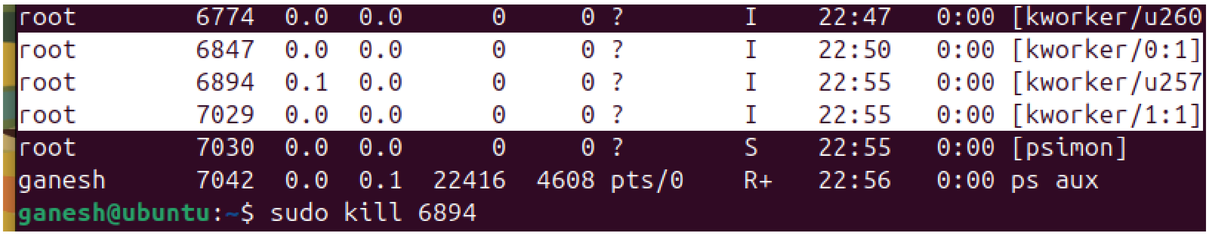
❖Kill <pid>

- I select to kill <pid> 7634

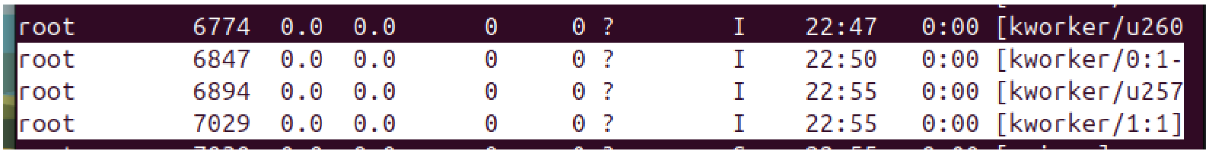
- This command sends a termination signal to the process with PID 7634.
- Ps aux for check id is kill or not
- After terminating the process, use this command to check if the process with PID 7634 is no longer running.

c.) Check again to confirm that the process has been terminated.

❖ Ps aux



```
root         6774  0.0  0.0      0     0 ?        I    22:47   0:00 [kworker/u260
root         6847  0.0  0.0      0     0 ?        I    22:50   0:00 [kworker/0:1]
root         6894  0.1  0.0      0     0 ?        I    22:55   0:00 [kworker/u257
root         7029  0.0  0.0      0     0 ?        I    22:55   0:00 [kworker/1:1]
root         7030  0.0  0.0      0     0 ?        S    22:55   0:00 [psimon]
ganesh       7042  0.0  0.1  22416  4608 pts/0    R+   22:56   0:00 ps aux
ganesh@ubuntu:~$ sudo kill 6894
```



```
root         6774  0.0  0.0      0     0 ?        I    22:47   0:00 [kworker/u260
root         6847  0.0  0.0      0     0 ?        I    22:50   0:00 [kworker/0:1-
root         6894  0.0  0.0      0     0 ?        I    22:55   0:00 [kworker/u257
root         7029  0.0  0.0      0     0 ?        I    22:55   0:00 [kworker/1:1]
```

4. System Information:

Abstract:

This document explores the retrieval and display of essential system information within a Unix-like operating system environment. Specifically, it outlines commands to obtain details about the system's CPU, memory, and disk usage. The focus is on practical implementation, demonstrating commands commonly used to gather system information efficiently.

Introduction:

Understanding system information is vital for system administrators and users alike, as it provides insights into system performance, resource utilization, and overall health. In this context, we delve into the Unix command-line interface, a powerful tool for retrieving and displaying system information. By mastering these commands, users can effectively monitor system resources and troubleshoot performance issues.

Problem Statement:

While many users are familiar with basic system commands, they may encounter difficulties when it comes to accessing detailed system information. Identifying the appropriate commands and interpreting their output accurately are essential to gain insights into system performance and diagnose potential issues. Thus, there is a need to elucidate these concepts and provide practical guidance for users seeking to enhance their system information retrieval skills.

Objective:

The objective of this document is to elucidate the commands and techniques involved in retrieving and displaying system information, including CPU details, memory utilization, and disk usage. By following the outlined steps, readers will gain a deeper understanding of system monitoring commands and techniques. The ultimate goal is to empower users to effectively monitor system resources and diagnose performance issues.

Future Scope:

As computing environments continue to evolve, the demand for efficient system monitoring tools will increase. Future advancements may include enhancements to system monitoring commands, integration with cloud-based monitoring platforms for centralized management, and automation of system health checks using artificial intelligence and machine learning algorithms. Moreover, ongoing research in system performance optimization will inform the development of more sophisticated monitoring solutions.

Conclusion:

In conclusion, understanding system information is crucial for maintaining system health and optimizing performance. By mastering system monitoring commands and techniques, users can effectively monitor system resources, identify performance bottlenecks, and take proactive measures to address issues. However, it is essential to stay informed about the latest advancements in system monitoring and adapt monitoring strategies accordingly. By following best practices and leveraging the power of system monitoring tools, users can ensure the stability and reliability of their systems.

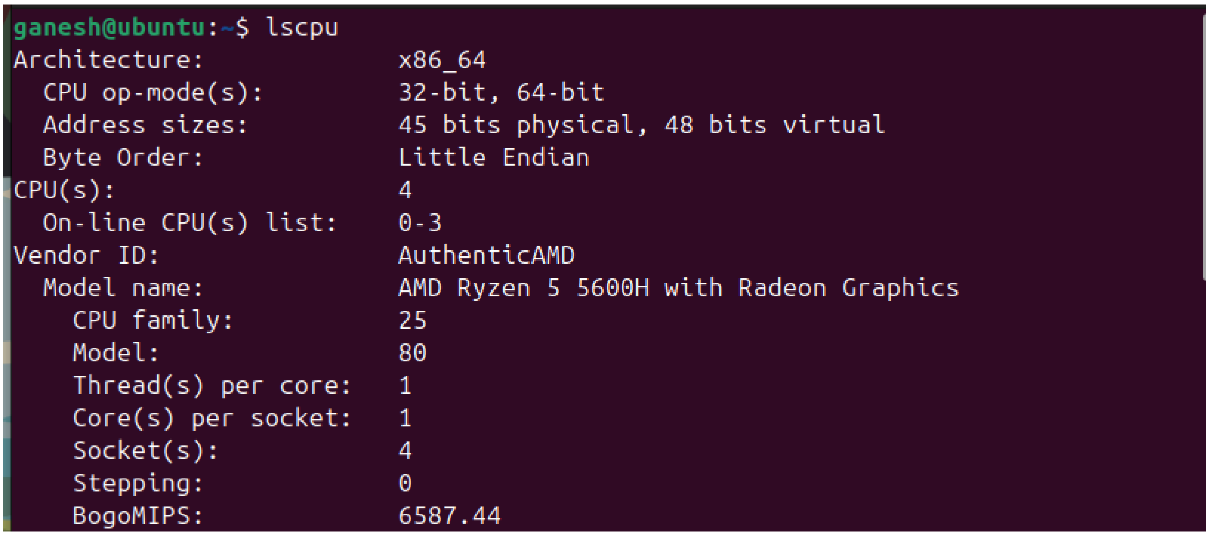
a) Display information about the system CPU.

❖ Cpuinfo

❖ lscpu

- This will provide you with detailed information about your CPU, including the architecture, model name, number of cores, threads per core, and other relevant details. If you need more specific information.

OUTPUT:-



```
ganesh@ubuntu:~$ lscpu
Architecture:            x86_64
  CPU op-mode(s):        32-bit, 64-bit
  Address sizes:         45 bits physical, 48 bits virtual
  Byte Order:            Little Endian
CPU(s):                  4
  On-line CPU(s) list:   0-3
Vendor ID:               AuthenticAMD
  Model name:            AMD Ryzen 5 5600H with Radeon Graphics
    CPU family:          25
    Model:               80
    Thread(s) per core:  1
    Core(s) per socket:  1
    Socket(s):           4
    Stepping:            0
    BogoMIPS:            6587.44
```

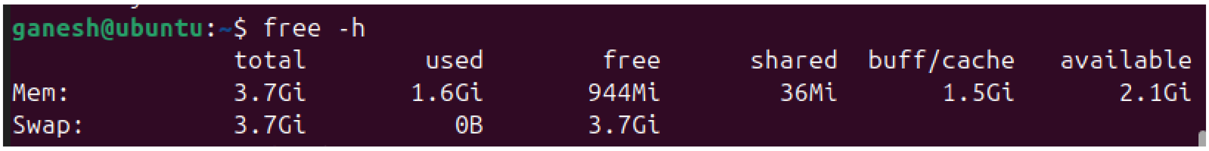
b.) Display information about the systems memory.

❖Free -h

Explanation:

free: Command used to display information about system memory.

-h: Human-readable option, which formats the output in a more easily understandable way (e.g., using MB, GB).



```
ganesh@ubuntu:~$ free -h
               total        used        free      shared  buff/cache   available
Mem:           3.7Gi       1.6Gi       944Mi        36Mi       1.5Gi       2.1Gi
Swap:          3.7Gi          0B       3.7Gi
```

This command will show you information about total, used, and free memory, both physical (RAM) and swap space, in a human-readable format.

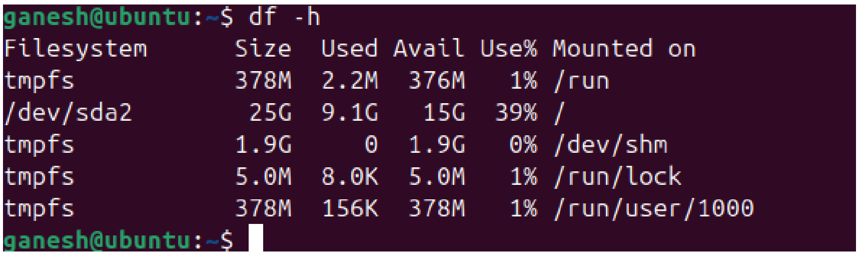
c.) Display the disk usage of the system.

❖Df -h

Explanation:

- ❖ df: Command used to display information about disk space usage.
- ❖ -h: Human-readable option, which formats the output in a more easily understandable way (e.g., using KB, MB, GB).
- ❖ This command will show you information about the disk space usage on all mounted filesystems, including total, used, and available space, as well as the percentage used.

Output:-



```
ganesh@ubuntu:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           378M  2.2M  376M   1% /run
/dev/sda2        25G  9.1G   15G  39% /
tmpfs           1.9G     0  1.9G   0% /dev/shm
tmpfs           5.0M  8.0K  5.0M   1% /run/lock
tmpfs           378M  156K  378M   1% /run/user/1000
ganesh@ubuntu:~$ 
```

5. Package Management:

Abstract:

This document explores the management of software packages within a Unix-like operating system environment. It outlines the steps involved in installing, removing, and searching for packages using package managers such as apt or yum. The focus is on practical implementation, demonstrating commands commonly used for package management tasks.

Introduction:

Package management is a fundamental aspect of system administration, enabling users to install, update, and remove software packages with ease. In this context, we delve into the Unix command-line interface and package management utilities, which streamline the process of managing software dependencies and configurations. By mastering package management commands, users can efficiently manage software installations and ensure system stability.

Problem Statement:

While package management utilities simplify the installation and management of software packages, users may encounter challenges when performing tasks such as installing, removing, or searching for packages. Identifying the appropriate commands and understanding their usage is essential to perform these tasks accurately and efficiently. Thus, there is a need to elucidate these concepts and provide practical guidance for users seeking to enhance their package management skills.

Objective:

The objective of this document is to elucidate the commands and techniques involved in package management tasks, including installing, removing, and searching for packages. By following the outlined steps, readers will gain a deeper understanding of package management utilities and their usage. The ultimate goal is to empower users to effectively manage software installations and dependencies, thereby optimizing system performance and functionality.

Future Scope:

As software ecosystems continue to evolve, the demand for efficient package management solutions will increase. Future advancements may include enhancements to package management utilities, integration with cloud-based package repositories for centralized management, and automation of package installation and configuration tasks using configuration management tools. Moreover, ongoing research in software packaging and distribution will inform the development of more sophisticated package management systems.

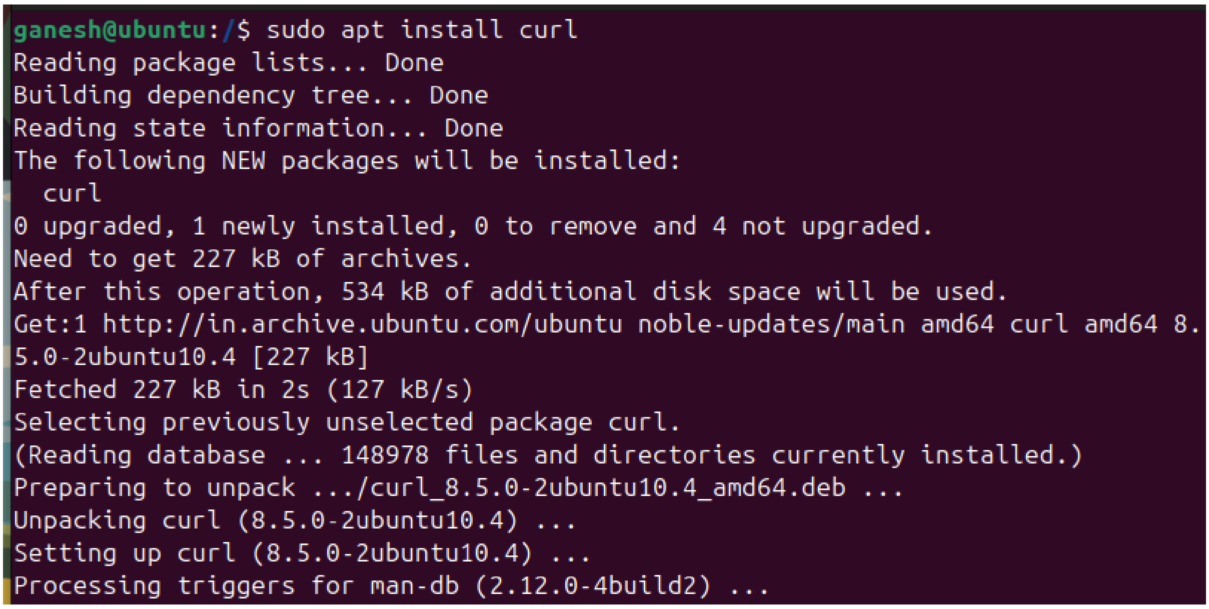
Conclusion:

In conclusion, package management is a critical aspect of system administration, enabling users to manage software installations and dependencies effectively. By mastering package management commands and techniques, users can streamline the process of installing, updating, and removing software packages, thereby ensuring system stability and functionality. However, it is essential to stay informed about the latest advancements in package management and adapt management strategies accordingly to meet evolving software requirements.

a.) Install a new package using the package manager .

- sudo apt update: This updates the package list on your system. It's a good practice to run this before installing new packages.
- sudo apt install curl: This installs curl on your system. The sudo command is used to execute the installation with administrative privileges. After entering your password, the system will download and install curl

Output:-



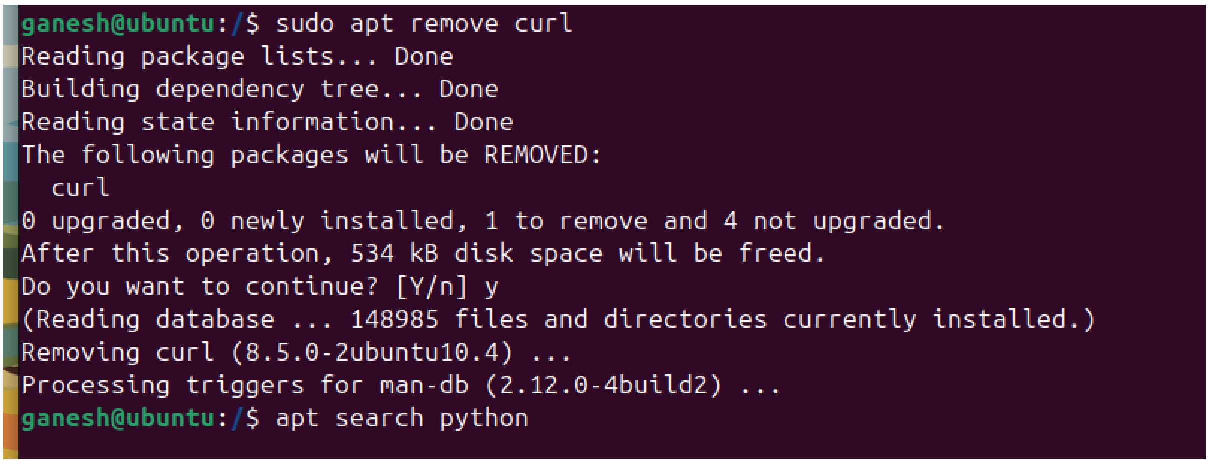
```
ganesh@ubuntu:/$ sudo apt install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 4 not upgraded.
Need to get 227 kB of archives.
After this operation, 534 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 curl amd64 8.
5.0-2ubuntu10.4 [227 kB]
Fetched 227 kB in 2s (127 kB/s)
Selecting previously unselected package curl.
(Reading database ... 148978 files and directories currently installed.)
Preparing to unpack .../curl_8.5.0-2ubuntu10.4_amd64.deb ...
Unpacking curl (8.5.0-2ubuntu10.4) ...
Setting up curl (8.5.0-2ubuntu10.4) ...
Processing triggers for man-db (2.12.0-4build2) ...
```

b.) Remove the package you installed.

- sudo: Execute the command with administrative privileges.

- apt: The package manager used in Ubuntu.
- remove: Command to uninstall a package.
- curl: The package name to be removed (curl)

output:-



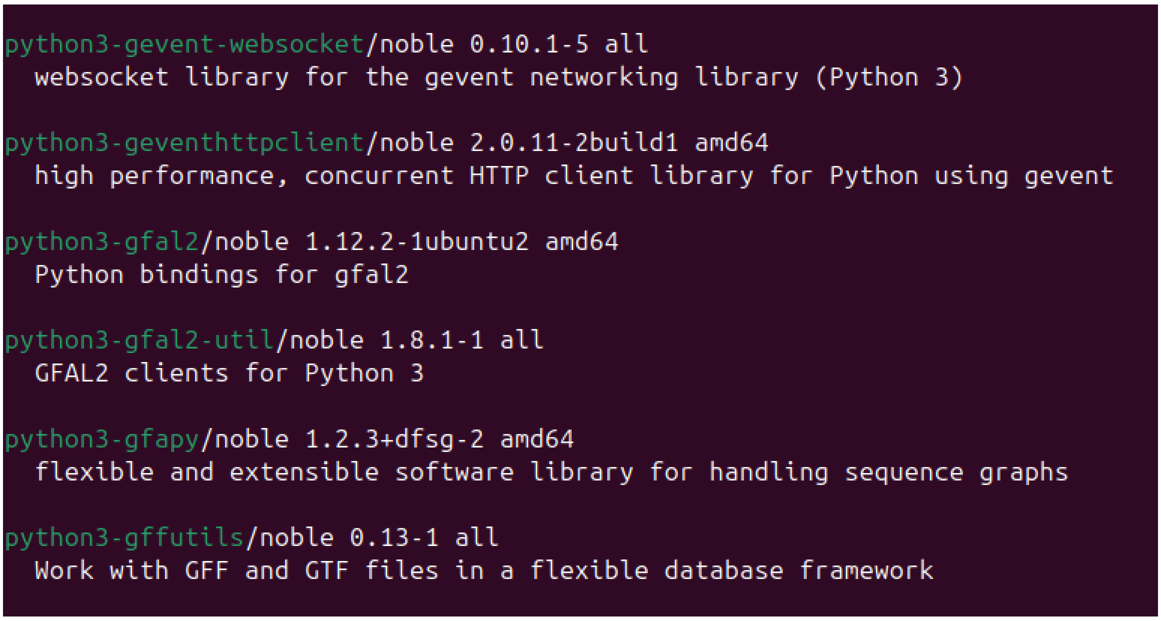
```
ganesh@ubuntu:/$ sudo apt remove curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be REMOVED:
  curl
0 upgraded, 0 newly installed, 1 to remove and 4 not upgraded.
After this operation, 534 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 148985 files and directories currently installed.)
Removing curl (8.5.0-2ubuntu10.4) ...
Processing triggers for man-db (2.12.0-4build2) ...
ganesh@ubuntu:/$ apt search python
```

c.) Search for a package related to Python and list its details.

❖apt search python

- apt: The package manager used in Ubuntu.
- search: Command to search for packages.
- python: The keyword used for searching.

Output:



```
python3-gevent-websocket/noble 0.10.1-5 all
  websocket library for the gevent networking library (Python 3)

python3-geventhttpclient/noble 2.0.11-2build1 amd64
  high performance, concurrent HTTP client library for Python using gevent

python3-gfal2/noble 1.12.2-1ubuntu2 amd64
  Python bindings for gfal2

python3-gfal2-util/noble 1.8.1-1 all
  GFAL2 clients for Python 3

python3-gfapy/noble 1.2.3+dfsg-2 amd64
  flexible and extensible software library for handling sequence graphs

python3-gffutils/noble 0.13-1 all
  Work with GFF and GTF files in a flexible database framework
```

This command will provide a list of packages related to Python, along with their descriptions. You can then choose a specific package and use the apt show command to get detailed information. For example, if you want to get details about the package named "python3":

6.) User and Group Management:

Abstract:

This document explores user and group management within a Unix-like operating system environment. It outlines the steps involved in creating a new user, setting a password for the user, creating a new group, and adding the user to the group. The focus is on practical implementation, demonstrating commands commonly used for user and group management tasks.

Introduction:

User and group management is essential for controlling access to system resources and maintaining security within a Unix-like operating system. By creating users and groups, administrators can assign permissions and define access rights to files, directories, and other resources. In this context, we delve into the Unix command-line interface, exploring commands and techniques for managing users and groups effectively.

Problem Statement:

While user and group management utilities simplify the process of creating users and groups, administrators may encounter challenges when performing tasks such as setting passwords and managing group memberships. Identifying the appropriate commands and understanding their usage is essential to perform these tasks accurately and securely. Thus, there is a need to elucidate these concepts and provide practical guidance for users seeking to enhance their user and group management skills.

Objective:

The objective of this document is to elucidate the commands and techniques involved in user and group management tasks, including creating users, setting passwords, creating groups, and managing group memberships. By following the outlined steps, readers will gain a deeper understanding of user and group management utilities and their usage. The ultimate goal is to empower users to effectively control access to system resources and maintain security within their environments.

Future Scope:

As computing environments become increasingly complex and dynamic, the demand for efficient user and group management solutions will increase. Future advancements may include enhancements to user and group management utilities, integration with identity management systems for centralized management, and automation of user and group provisioning tasks using configuration management tools. Moreover, ongoing research in cybersecurity will inform the development of more robust access control mechanisms to address evolving security threats.

Conclusion:

In conclusion, user and group management is a critical aspect of system administration, enabling administrators to control access to system resources and maintain security effectively. By mastering user and group management commands and techniques, administrators can streamline the process of creating users, assigning permissions, and managing group memberships, thereby ensuring the integrity and confidentiality of system data. However, it is essential to stay informed about the latest advancements in user and group management and adapt management strategies accordingly to meet evolving security requirements.

a.) Create a new user named testuser.

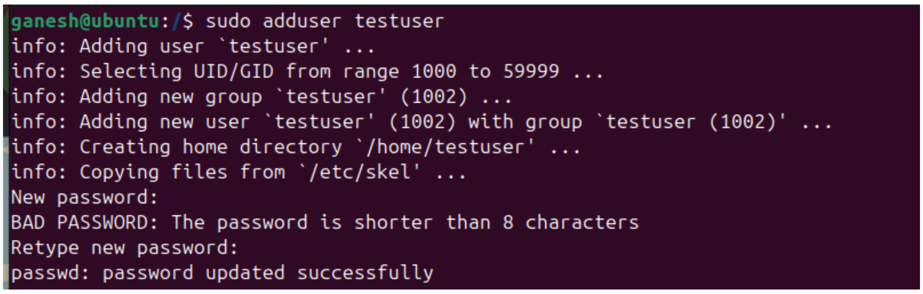
b.) Set a password for "testuser".

❖sudo adduser testuser

- ❖ sudo: Execute the command with administrative privileges.
- ❖ adduser: Command to add a new user.
- ❖ testuser: The username you want to create.

- You will be prompted to set a password for the new user.
- Enter and confirm the password when prompted.
- You can press Enter for other optional information like full name, room number, etc., or provide the information as needed.
- After completing these steps, the "testuser" should be created, and you can log in using this username and the associated password.

Output:-



```
ganesh@ubuntu:/$ sudo adduser testuser
info: Adding user `testuser' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `testuser' (1002) ...
info: Adding new user `testuser' (1002) with group `testuser (1002)' ...
info: Creating home directory `/home/testuser' ...
info: Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
```

C.)**Create a new group named "testgroup".**

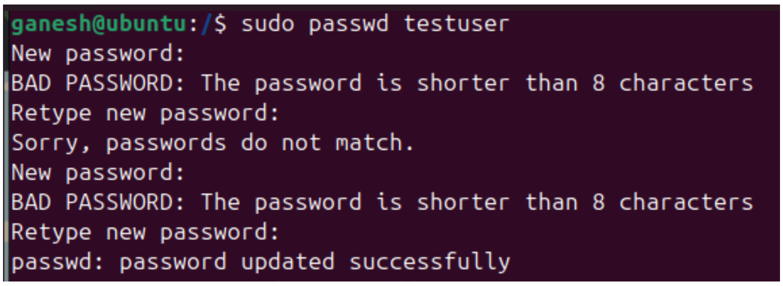
❖Sudo addgroup testgroup

- ❖ sudo: Execute the command with administrative

privliges.

- ❖ addgroup: Command to add a new group.
- ❖ testgroup: The name of the group you want to create.

Output:-



```
ganesh@ubuntu:/$ sudo passwd testuser
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
Sorry, passwords do not match.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
```

d.) Add "testuser" to "testgroup".

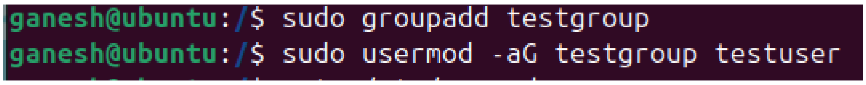
```
❖Sudo usermod -aG testgroup testuser
```

- ❖ sudo: Execute the command with administrative privileges.
- ❖ usermod: Command to modify user account settings.
- ❖ -aG: Options to append user to supplementary group(s).
- ❖ testgroup: The name of the group you want to add the user to.
- ❖ testuser: The name of the user you want to add to the group.

- ✓ After running this command, the user "testuser" will be added to the group "testgroup". To verify, you can check the group memberships of "testuser" using the id command:

```
❖id testuser
```

Output:-



```
ganesh@ubuntu:/$ sudo groupadd testgroup
ganesh@ubuntu:/$ sudo usermod -aG testgroup testuser
```

7.) Networking:

Abstract:

This document delves into networking tasks within a Unix-like operating system environment, focusing on practical implementation and command-line utilities commonly used for network configuration and troubleshooting. It outlines steps such as displaying network configuration, pinging remote servers, checking network interface status, utilizing port scanners for IP details, retrieving country and contact details of IPs, and examining DNS details of domain names.

Introduction:

Networking is a fundamental aspect of modern computing, facilitating communication and data exchange between devices and systems. In Unix-like operating systems, network configuration and troubleshooting are often performed using command-line utilities. This document explores these utilities, providing insights into their usage and practical applications.

Problem Statement:

While networking utilities offer powerful capabilities for managing and diagnosing network-related issues, users may encounter challenges when performing tasks such as retrieving network configuration details, troubleshooting connectivity problems, and gathering information about remote hosts. Identifying the appropriate commands and understanding their output is essential to effectively address these challenges.

Objective:

The objective of this document is to elucidate the commands and techniques involved in networking tasks within a Unix-like environment. By following the outlined steps, readers will gain a deeper understanding of network configuration, troubleshooting, and information retrieval processes. The document aims to empower users to effectively manage and troubleshoot network-related issues.

Future Scope:

As networking technologies continue to evolve, the demand for advanced networking tools and capabilities will increase. Future advancements may include enhancements to network diagnostic utilities, integration with cloud-based networking platforms for centralized management, and automation of network configuration tasks using orchestration tools. Moreover, ongoing research in network security and performance optimization will inform the development of more sophisticated networking solutions.

Conclusion:

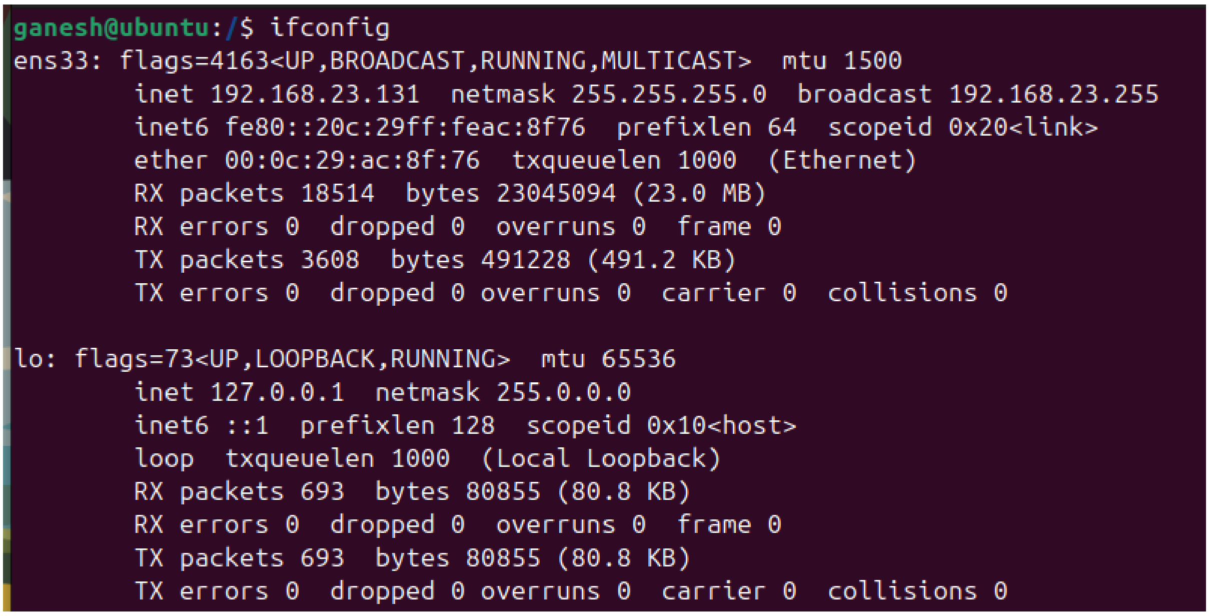
In conclusion, networking is a critical component of modern computing, enabling communication and data exchange across diverse environments. By mastering networking utilities and techniques, users can effectively manage network configuration, diagnose connectivity issues, and gather information about remote hosts. However, it is essential to stay informed about the latest advancements in networking technologies and adapt networking strategies accordingly to meet evolving requirements.

a.) Display the current network configuration.

❖ifconfig

- ❖ ifconfig: Command to configure network interfaces and display their configuration.
- ❖ commands will provide information about network interfaces, IP addresses, and other network-related configurations.

Output:-



```
ganesh@ubuntu:/$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.23.131  netmask 255.255.255.0  broadcast 192.168.23.255
        inet6 fe80::20c:29ff:feac:8f76  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:ac:8f:76  txqueuelen 1000  (Ethernet)
        RX packets 18514  bytes 23045094 (23.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3608  bytes 491228 (491.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 693  bytes 80855 (80.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 693  bytes 80855 (80.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

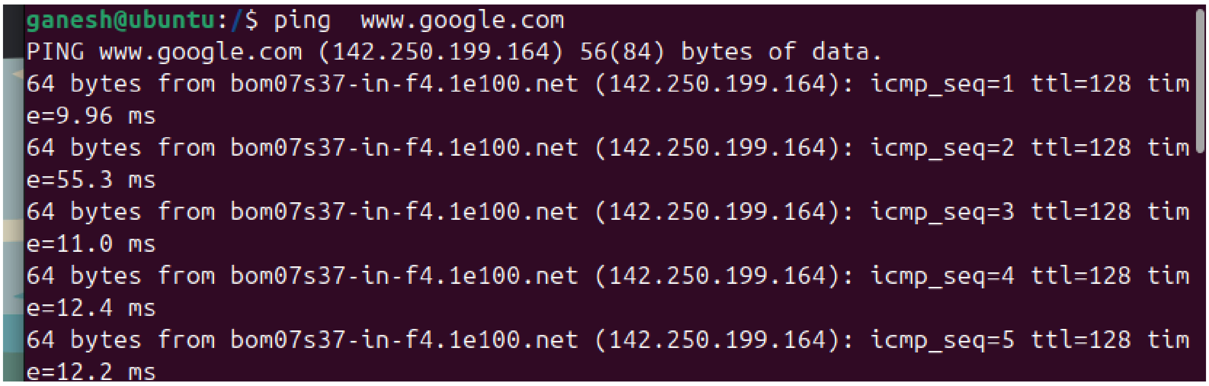
b.) Ping a remote server (e.g., "ping

www.google.com.

❖Ping www.google.com

- ❖ This command sends ICMP echo requests to the specified domain (e.g., "www.google.com") to check network connectivity and measure round-trip time.

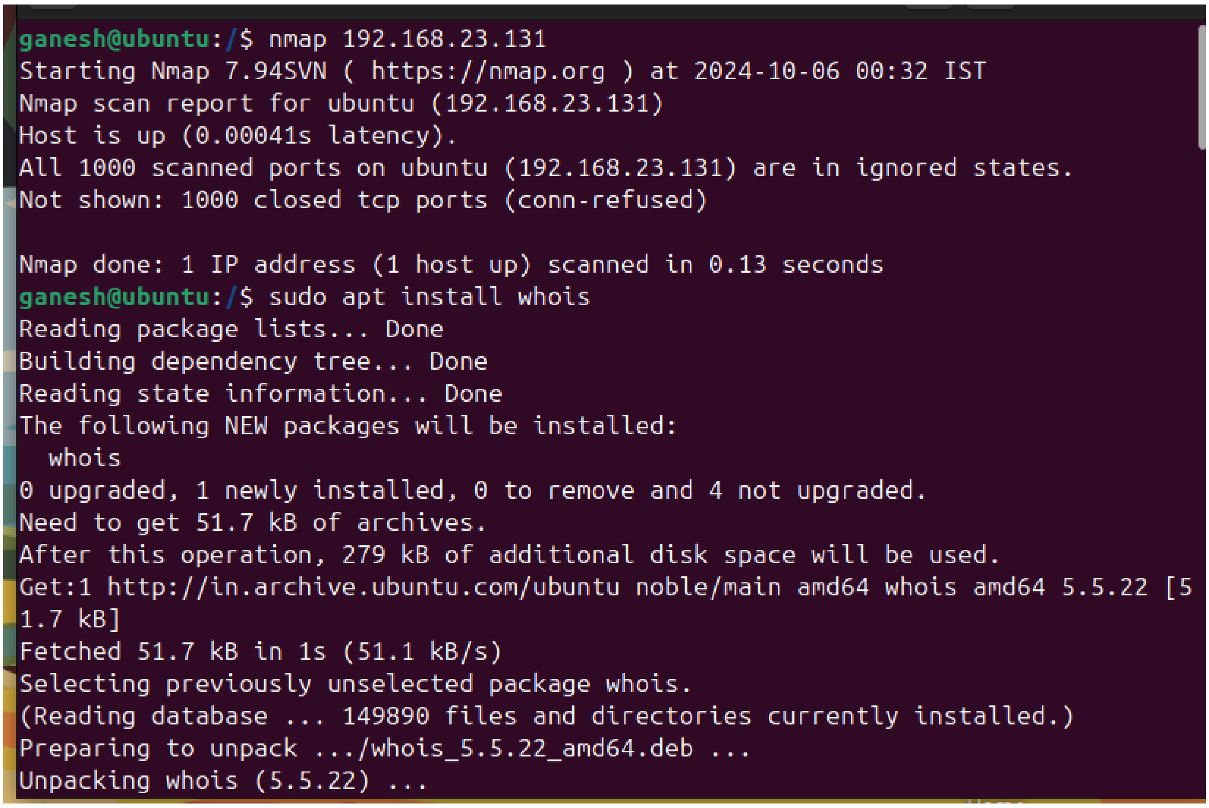
Output:-



```
ganesh@ubuntu:/$ ping  www.google.com
PING www.google.com (142.250.199.164) 56(84) bytes of data.
64 bytes from bom07s37-in-f4.1e100.net (142.250.199.164): icmp_seq=1 ttl=128 tim
e=9.96 ms
64 bytes from bom07s37-in-f4.1e100.net (142.250.199.164): icmp_seq=2 ttl=128 tim
e=55.3 ms
64 bytes from bom07s37-in-f4.1e100.net (142.250.199.164): icmp_seq=3 ttl=128 tim
e=11.0 ms
64 bytes from bom07s37-in-f4.1e100.net (142.250.199.164): icmp_seq=4 ttl=128 tim
e=12.4 ms
64 bytes from bom07s37-in-f4.1e100.net (142.250.199.164): icmp_seq=5 ttl=128 tim
e=12.2 ms
```

c.) Use Port Scanner to identify the details of any IP:

- This command performs a port scan on the specified IP address to identify open ports and services running on them
- Output:-

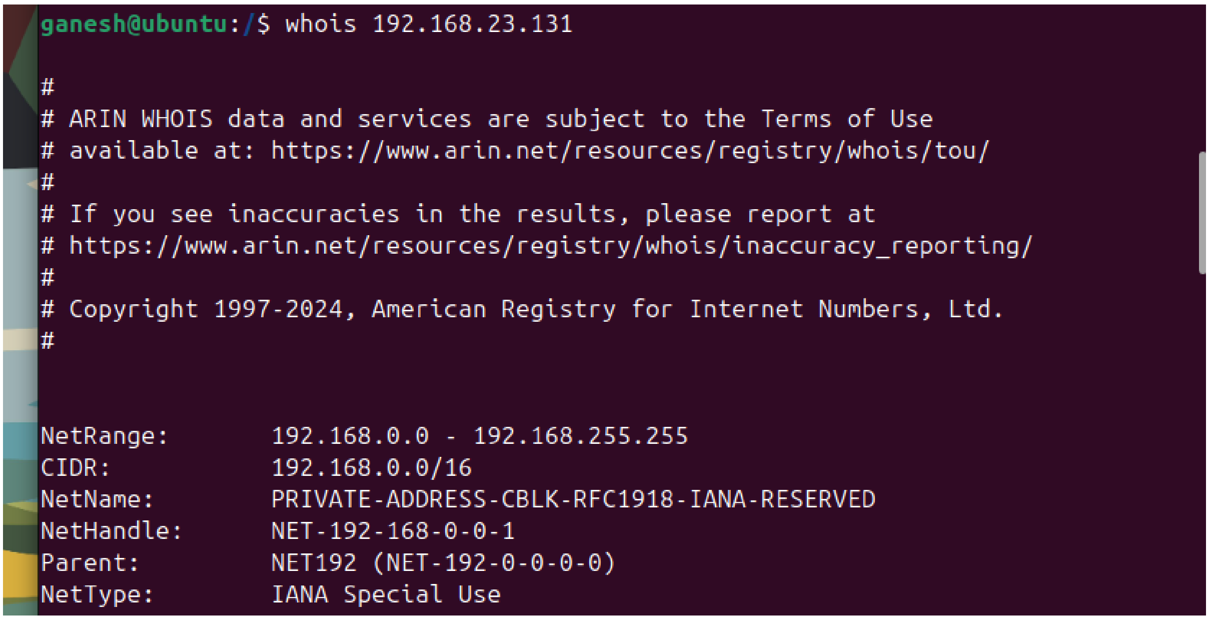


```
ganesh@ubuntu:/$ nmap 192.168.23.131
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-06 00:32 IST
Nmap scan report for ubuntu (192.168.23.131)
Host is up (0.00041s latency).
All 1000 scanned ports on ubuntu (192.168.23.131) are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
ganesh@ubuntu:/$ sudo apt install whois
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  whois
0 upgraded, 1 newly installed, 0 to remove and 4 not upgraded.
Need to get 51.7 kB of archives.
After this operation, 279 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu noble/main amd64 whois amd64 5.5.22 [5
1.7 kB]
Fetched 51.7 kB in 1s (51.1 kB/s)
Selecting previously unselected package whois.
(Reading database ... 149890 files and directories currently installed.)
Preparing to unpack .../whois_5.5.22_amd64.deb ...
Unpacking whois (5.5.22) ...
```

d.) Check the Country and contact details of any IP:

- ❖ This command retrieves WHOIS information for the specified IP address, including details about the organization, country, and contact information.

Output: -

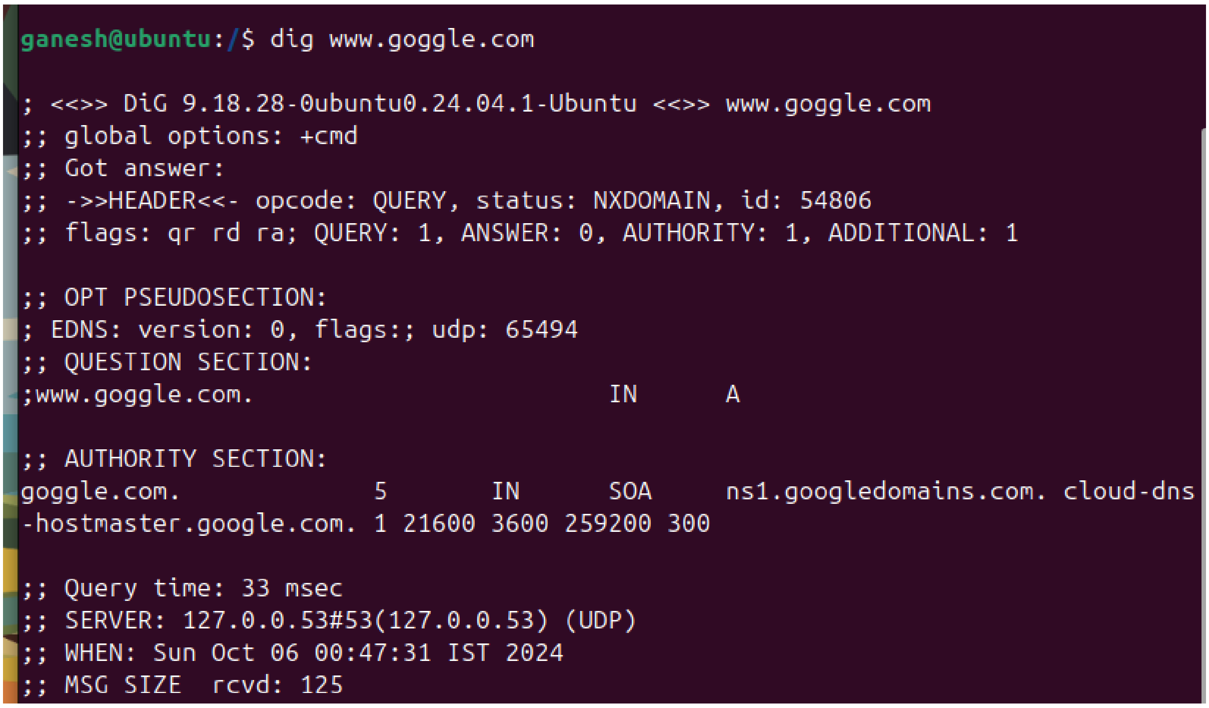


```
ganesh@ubuntu:/$ whois 192.168.23.131

#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
# Copyright 1997-2024, American Registry for Internet Numbers, Ltd.
#


NetRange:       192.168.0.0 - 192.168.255.255
CIDR:           192.168.0.0/16
NetName:        PRIVATE-ADDRESS-CBLK-RFC1918-IANA-RESERVED
NetHandle:      NET-192-168-0-0-1
Parent:         NET192 (NET-192-0-0-0-0)
NetType:        IANA Special Use
```

e.) Check the DNS details of any Domain Name

- This command performs a DNS lookup for the specified domain name to retrieve its IP address and other DNS-related information.

Output:-



```
ganesh@ubuntu:/$ dig www.goggle.com

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> www.goggle.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 54806
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.goggle.com.                        IN      A

;; AUTHORITY SECTION:
goggle.com.             5       IN      SOA     ns1.googledomains.com. cloud-dns
-hostmaster.google.com. 1 21600 3600 259200 300

;; Query time: 33 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sun Oct 06 00:47:31 IST 2024
;; MSG SIZE  rcvd: 125
```

8.Text Processing and Searching:

Abstract:

This document explores text processing and searching techniques within a Unix-like operating system environment. It outlines the steps involved in creating text files, searching for specific words or patterns using grep, replacing text using sed, extracting specific columns or fields using awk, and searching for files using find command. The focus is on practical implementation, demonstrating commands and utilities commonly used for text manipulation and searching tasks.

Introduction:

Text processing and searching are essential tasks in data analysis, system administration, and software development. In Unix-like operating systems, command-line utilities provide powerful tools for manipulating and searching text files efficiently. This document delves into these utilities, providing insights into their usage and practical applications.

Problem Statement:

While command-line utilities offer powerful capabilities for text processing and searching, users may encounter challenges when performing tasks such as searching for specific words or patterns, replacing text, extracting data from structured files, and searching for files based on specific criteria. Identifying the appropriate commands and understanding their syntax and options is essential to perform these tasks accurately and efficiently.

Objective:

The objective of this document is to elucidate the commands and techniques involved in text processing and searching within a Unix-like environment. By following the outlined steps, readers will gain a deeper understanding of text manipulation utilities such as grep, sed, awk, and find. The document aims to empower users to effectively manipulate and search text files, automate repetitive tasks, and streamline data analysis processes.

Future Scope:

As data volumes continue to grow, the demand for efficient text processing and searching tools will increase. Future advancements may include enhancements to text processing utilities, integration with machine learning algorithms for natural language processing tasks, and automation of text analysis workflows using scripting languages and workflow orchestration tools. Moreover, ongoing research in data mining and information retrieval will inform the development of more sophisticated text processing and searching solutions.

Conclusion:

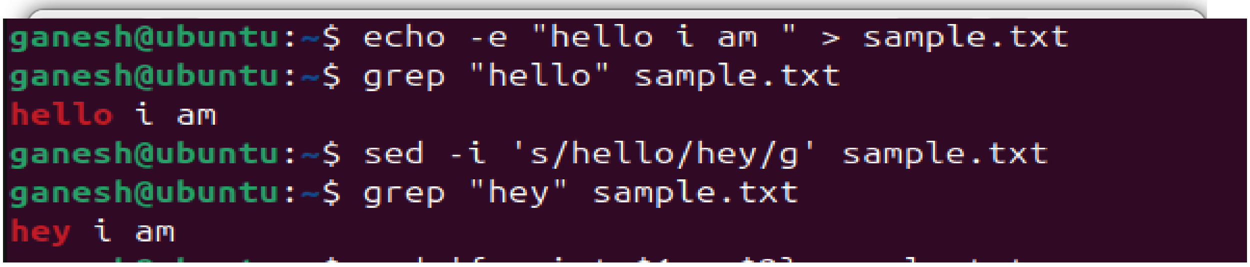
In conclusion, text processing and searching are integral components of data analysis and system administration tasks. By mastering text manipulation utilities such as grep, sed, awk, and find, users can efficiently perform tasks such as searching for specific patterns, replacing text, extracting data, and searching for files within a directory hierarchy. However, it is essential to stay informed about the latest advancements in text processing and searching techniques and adapt strategies accordingly to meet evolving requirements.

a.) Create a text file with some sample text.

❖Echo “ text”>filename

- echo "This is a sample text file.": Outputs the specified text.
- >: Redirects the output to a file named "sample.txt".

Output:-

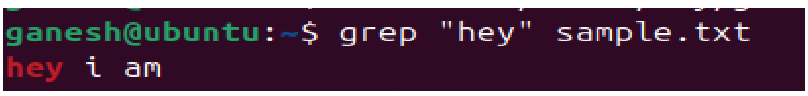


```
ganesh@ubuntu:~$ echo -e "hello i am " > sample.txt
ganesh@ubuntu:~$ grep "hello" sample.txt
hello i am
ganesh@ubuntu:~$ sed -i 's/hello/hey/g' sample.txt
ganesh@ubuntu:~$ grep "hey" sample.txt
hey i am
```

b.) Use grep to search for a specific word or pattern within the file.

❖grep "practice" data.txt

- grep "specific_word": Searches for the specified word ("specific_word").
- sample.txt: Specifies the file to search within.



```
ganesh@ubuntu:~$ grep "hey" sample.txt
hey i am
```

c.) Use sed to replace a word or pattern within the file.

❖sed -i 's/old_word/new_word/g' sample.txt

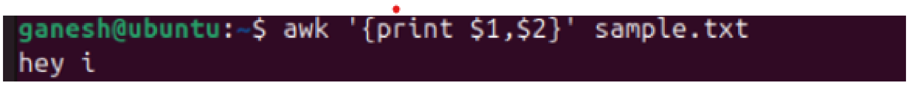
- sed -i: Invokes sed in-place editing mode to modify the file directly.
- 's/old_word/new_word/g': Specifies the substitution pattern, replacing "old_word" with "new_word" globally.

d.) Use awk to extract specific columns or fields from a text.file.

❖awk '{print $1, $3}' sample.tct

- sed -i: Invokes sed in-place editing mode to modify the file directly.
- 's/old_word/new_word/g': Specifies the substitution pattern, replacing "old_word" with "new_word" globally.
- sample.txt: Specifies the file to modify.

Output:-

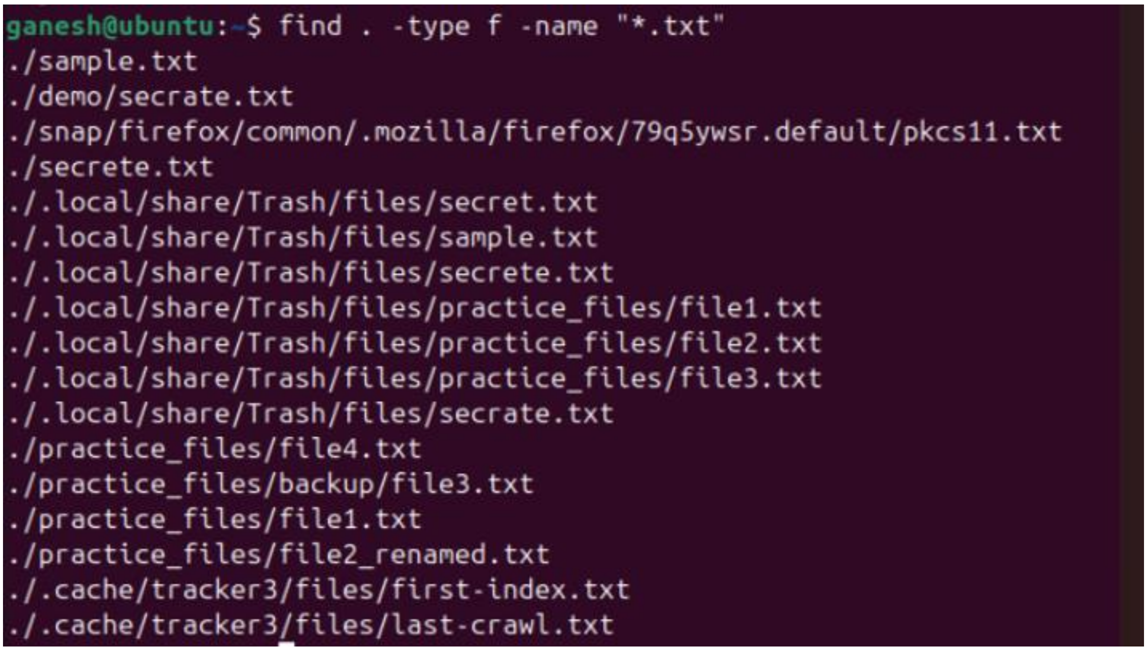


```
ganesh@ubuntu:~$ awk '{print $1,$2}' sample.txt
hey i
```

e.) Use find to search for files matching specific criteria within a directory hirarchary.

❖Find/path/to/directory -type f-name “*.txt”

- find /path/to/directory: Specifies the directory to start the search from.
- -type f: Specifies to search for files (not directories).
- -name "*.txt": Specifies to search for files with the ".txt" extension.
- These commands allow you to perform various text file operations like creation, searching, replacing, extracting, and finding files based on specific criteria within a directory hierarchy.



```
ganesh@ubuntu:~$ find . -type f -name "*.txt"
./sample.txt
./demo/secrate.txt
./snap/firefox/common/.mozilla/firefox/79q5ywsr.default/pkcs11.txt
./secrete.txt
./.local/share/Trash/files/secret.txt
./.local/share/Trash/files/sample.txt
./.local/share/Trash/files/secrete.txt
./.local/share/Trash/files/practice_files/file1.txt
./.local/share/Trash/files/practice_files/file2.txt
./.local/share/Trash/files/practice_files/file3.txt
./.local/share/Trash/files/secrate.txt
./practice_files/file4.txt
./practice_files/backup/file3.txt
./practice_files/file1.txt
./practice_files/file2_renamed.txt
./.cache/tracker3/files/first-index.txt
./.cache/tracker3/files/last-crawl.txt
```

9.) Archiving and Compression:

Concept:- of archiving and compression.

- Archiving: is the process of combining multiple files or directories into a single file, making it easier to transfer or store them.
- Compression: reduces the size of files to save disk space and speed up data transfer.

- Archiving involves combining multiple files or directories into a single file, making it easier to transfer or store them. Compression, on the other hand, reduces the size of files to save disk space and speed up data transfer. Two commonly used tools for archiving and compression in Linux are tar and zip .
- It come to solve the problem are following:-
    - The problem mainly solved by this utility is the need for a straightforward and efficient method to compress and archive files and directories in Ubuntu. Specifically, it addresses the following challenges:

    - Ease of Use: Many users, especially those not familiar with complex command-line tools, struggle to create compressed archives in Ubuntu. This utility provides a user-friendly interface to simplify the process.

    - Versatility: Different users may have different preferences for compression formats or specific requirements for archiving. By supporting both tar/gzip and zip formats, the utility accommodates a wide range of needs.

- Overall, the utility solves the problem of providing a convenient, versatile, efficient, and reliable solution for creating compressed archives in Ubuntu, catering to the needs of both novice and experienced users.

a.) Create a compressed archive of the “practice file” directory.

Command:-

- To create a compressed archive of the "practice_files" directory, you can use the tar command with the z option to compress the archive using gzip:

```
tar -czvf practice_files.tar.gz practice_files
```

- tar: The command-line utility used for archiving files.
- -czvf: Options for creating a compressed archive (c for create, z for gzip compression, v for verbose output, f for specifying the output file).
- practice_files.tar.gz: The name of the output compressed archive file.
- practice_files: The name of the directory to be archived.

- This command will create a compressed archive named "practice_files.tar.gz" containing all the files and directories within the "practice_files" directory.

b.) Extract the contents of the archive into a new directory.

- Extracting the contents of an archive into a new directory involves taking the compressed archive file and decompressing its contents while preserving the directory structure into a specified location. Here's a step-by-step explanation:

1. Choose Archive: Select the compressed archive file (e.g., .tar.gz, .zip) from which you want to extract the contents.
2. Specify Extraction Location: Decide where you want to extract the contents of the archive. This typically involves creating a new directory where the contents will be placed. You can choose an existing directory or create a new one for this purpose.
3. Execute Extraction Command: Use the appropriate extraction command based on the format of the archive file:

For .tar.gz files:

```
tar -xzvf archive_name.tar.gz -C /path/to/new_directory
```

- Replace archive_name.tar.gz with the name of your archive file, and /path/to/new_directory with the path to the new directory where you want to extract the contents.

For .zip files:

```
unzip archive_name.zip -d /path/to/new_directory
```

- Replace archive_name.zip with the name of your archive file, and /path/to/new_directory with the path to the new directory where you want to extract the contents.

- Extraction Process: When you execute the extraction command, the contents of the archive will be decompressed and extracted into the specified directory. This process may take some time depending on the size of the archive and the performance of your system.

- Check Extraction: Once the extraction process is complete, you can navigate to the specified directory to verify that the contents have been successfully extracted. You should see the original directory structure as it was stored in the archive, with all files and subdirectories intact.

- By following these steps, you can extract the contents of an archive into a new directory, making it

c.) View the contents of the extracted directory

- To view the contents of the extracted directory in Linux, you can use the ls command. Here's a detailed explanation:

a) Open Terminal: Launch the Terminal application on your Ubuntu system. You can typically find it in the applications menu or by searching for "Terminal" in the system's search function.

b) Navigate to the Extracted Directory: Use the cd command to change the current directory to the location where you extracted the archive. For example, if you extracted the archive to a directory named "extracted_files" in your home directory, you would use the following command:

```
cd ~/extracted_files
```

2.List Contents with ls: Once you're in the extracted directory, use the ls command to list its contents. Simply type:

```
ls
```

- ❖ This command will display a list of files and directories in the current directory.

Understanding ls Output:

- Each entry in the list represents a file or directory present in the extracted directory.
- Files are listed by their names, while directories are also listed with their names, and are marked with a trailing slash (/) character.
- If there are many files in the directory, they may not all be displayed on a single screen. You can scroll through the list using the scrollbar or use the ls command with options for better formatting.

- **View Detailed Information:** Optionally, you can view detailed information about each file and directory using the -l (long) option with ls. This provides additional details such as file permissions, ownership, size, and modification date.

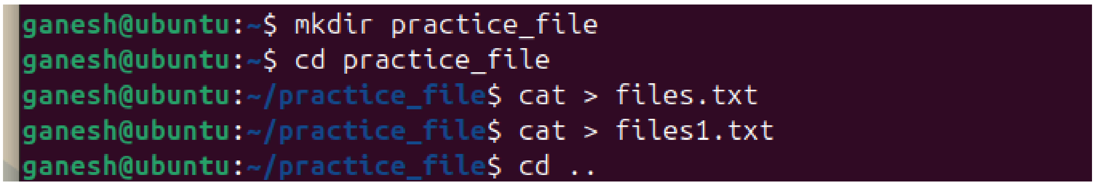
```
ls -l
```

- View Hidden Files: By default, the ls command does not show hidden files (those whose names start with a dot). To view hidden files as well, use the -a (all) option:

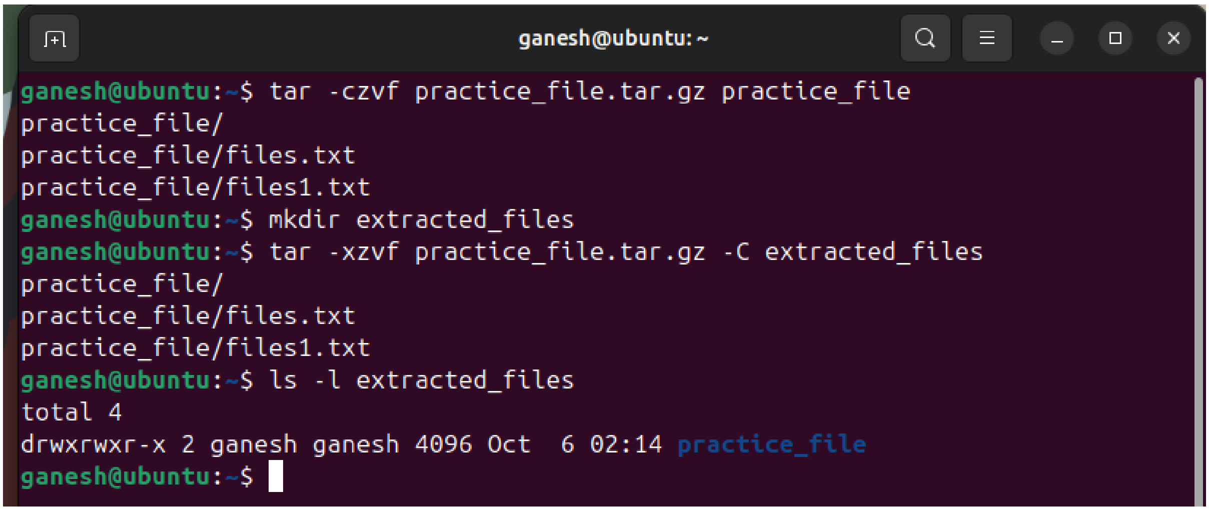
```
Ls -a
```

- By following these steps, you can effectively view the contents of the extracted directory in Ubuntu using the Terminal, gaining insight into the files and directories that were extracted from the archive.

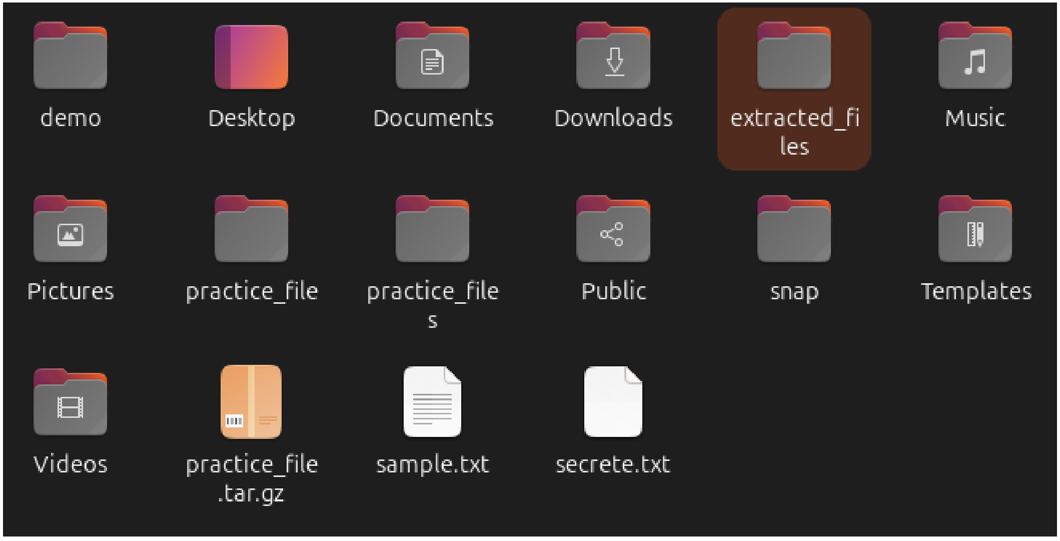
Output of all types of command are:-



```
ganesh@ubuntu:~$ mkdir practice_file
ganesh@ubuntu:~$ cd practice_file
ganesh@ubuntu:~/practice_file$ cat > files.txt
ganesh@ubuntu:~/practice_file$ cat > files1.txt
ganesh@ubuntu:~/practice_file$ cd ..
```



```
ganesh@ubuntu:~$ tar -czvf practice_file.tar.gz practice_file
practice_file/
practice_file/files.txt
practice_file/files1.txt
ganesh@ubuntu:~$ mkdir extracted_files
ganesh@ubuntu:~$ tar -xzvf practice_file.tar.gz -C extracted_files
practice_file/
practice_file/files.txt
practice_file/files1.txt
ganesh@ubuntu:~$ ls -l extracted_files
total 4
drwxrwxr-x 2 ganesh ganesh 4096 Oct  6 02:14 practice_file
ganesh@ubuntu:~$ 
```



10. Deploy LAMP Server:

a.)Follow all the Instruction to Install Apache2, PHP and MYSQL/MariaDB on Ubuntu.

ABOUT LAMP SERVER:-

- LAMP is an open-source Web development platform that uses ***Linux*** as the operating system, ***Apache*** as the Web server, ***MySQL*** as the relational database management system and ***PHP/Perl/Python*** as the object-oriented scripting language.
- Sometimes LAMP is referred to as a LAMP stack because the platform has four layers. Stacks can be built on different operating systems.
- LAMP is a example of a web service stack, named as an ***acronym***. The LAMP components are largely interchangeable and not limited to the original selection. LAMP is suitable for building dynamic web sites and web applications.

- Since its creation, the LAMP model has been adapted to another component, though typically consisting of free and open-source software.

- ❖ Introduction:

- ➢ In the contemporary digital landscape, deploying web applications efficiently is crucial for businesses and developers alike. Among the various technologies available, the LAMP (Linux, Apache, MySQL/MariaDB, PHP) stack stands out as a robust and widely adopted solution for hosting dynamic websites and web applications. This report outlines the process of deploying a LAMP server on Ubuntu, elucidating the installation steps, configuration procedures, and associated commands.

- ❖ Problem Statement:

- ➢ The deployment of a LAMP server entails a series of intricate steps, encompassing the installation and configuration of multiple components. Novice users may find it challenging to navigate through this process due to the technical complexity involved. Moreover, ensuring the seamless integration of each component and troubleshooting potential issues demand meticulous attention to detail.

- ❖ Objective:

- ➢ The primary objective of this report is to provide a comprehensive guide for deploying a LAMP server on Ubuntu, catering to users with varying levels of expertise. By elucidating the installation steps, configuring essential settings, and presenting relevant commands and screenshots, this report aims to streamline the deployment process and empower users to set up a functional LAMP environment effectively.

- ❖ Future Scope:

- ➢ The deployment of a LAMP server serves as a foundational step for hosting web applications and websites. Building upon this setup, users can explore advanced configurations, such as implementing security measures, optimizing performance, and integrating additional components or technologies. Furthermore, the acquired knowledge and skills can be leveraged to delve into other web hosting solutions and expand the scope of server administration expertise.

- ❖ Conclusion:

- In conclusion, the deployment of a LAMP server on Ubuntu facilitates the establishment of a robust and versatile web hosting environment. Through the systematic execution of installation procedures, configuration settings, and validation checks, users can create a stable platform for hosting dynamic web content and supporting diverse web applications. By adhering to the outlined guidelines and leveraging the provided resources, users can embark on their journey towards mastering server administration and web development.

- The diagram illustrates the sequential flow of steps involved in deploying a LAMP server on Ubuntu, depicting the interconnections and dependencies among different components.

- Detailed explanations of each command used during the installation and configuration process are provided, elucidating their purposes and step-by-step implementation procedures.

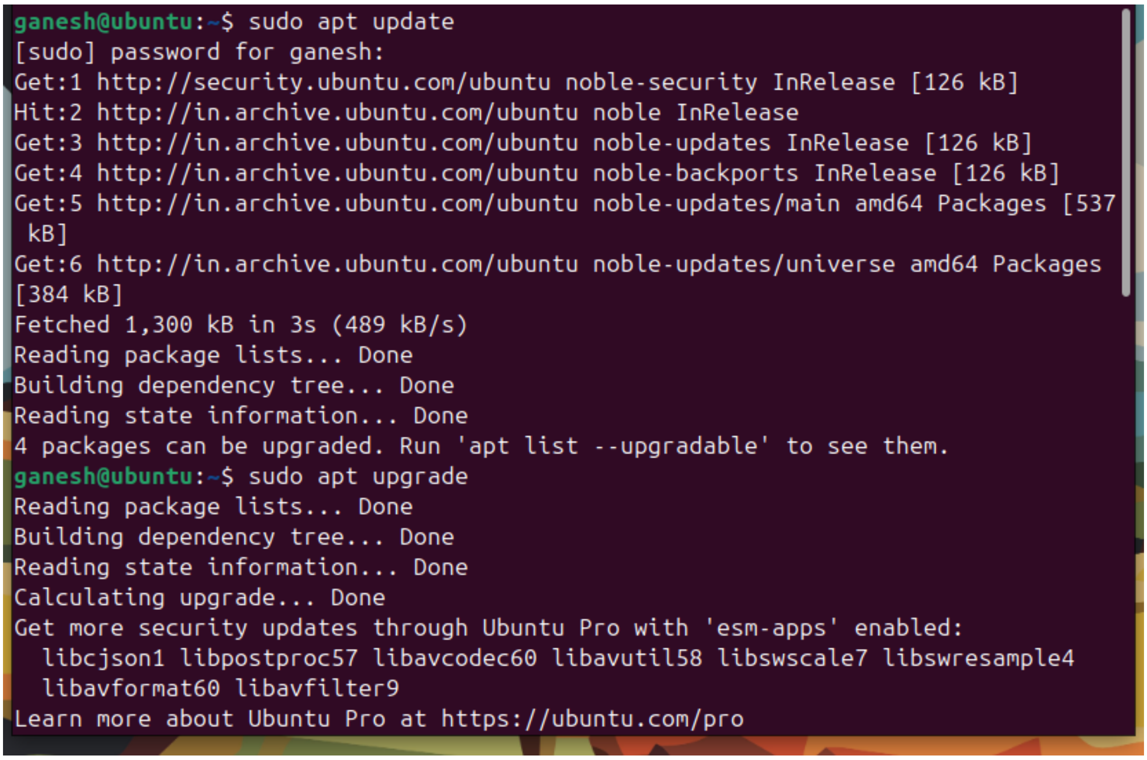
Commands for installation are:-

1.) Update Package Lists:

```
sudo apt update
```

- sudo: This command is used to execute subsequent commands with superuser privileges, which are necessary for system-wide operations like installing or updating software.
- apt: This is the package management tool used in Ubuntu. It's used for handling packages, including installing, updating, and removing software.
- update: This subcommand for apt updates the package lists, downloading information about the latest versions of packages from the repositories specified in /etc/apt/sources.list.

output:-



```
ganesh@ubuntu:~$ sudo apt update
[sudo] password for ganesh:
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://in.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://in.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [537
 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages
[384 kB]
Fetched 1,300 kB in 3s (489 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
ganesh@ubuntu:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
Get more security updates through Ubuntu Pro with 'esm-apps' enabled:
  libcjson1 libpostproc57 libavcodec60 libavutil58 libswscale7 libswresample4
  libavformat60 libavfilter9
Learn more about Ubuntu Pro at https://ubuntu.com/pro
```

2.) Install Apache:

```
sudo apt install apache2
```

- apt install: This subcommand installs the specified package and any dependencies it requires.
- apache2: This is the package name for the Apache web server in Ubuntu.

3.) Restart Apache:

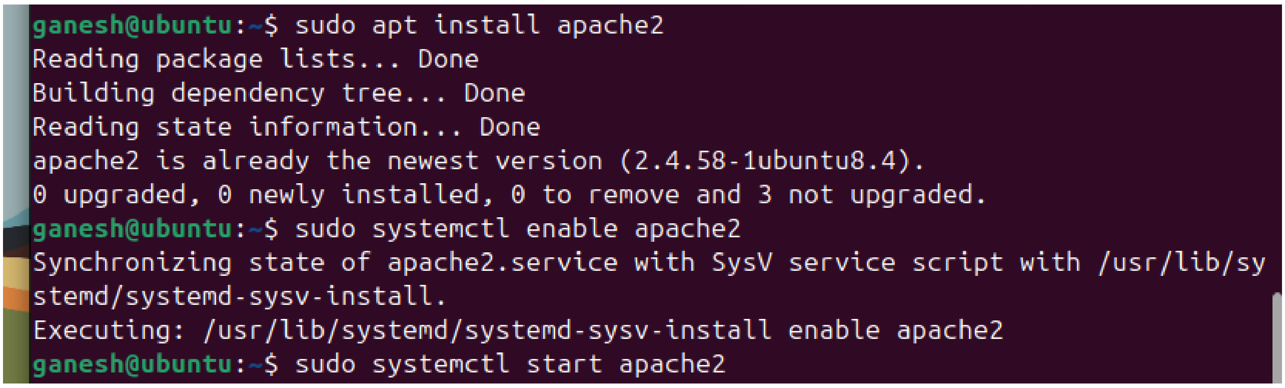
```
sudo systemctl restart apache2
```

sudo systemctl restart apache2

- systemctl: This command is used to manage system services in Ubuntu.
- start: This subcommand tells systemctl to start the specified service.

  apache2: This is the name of the Apache service.

Output:-



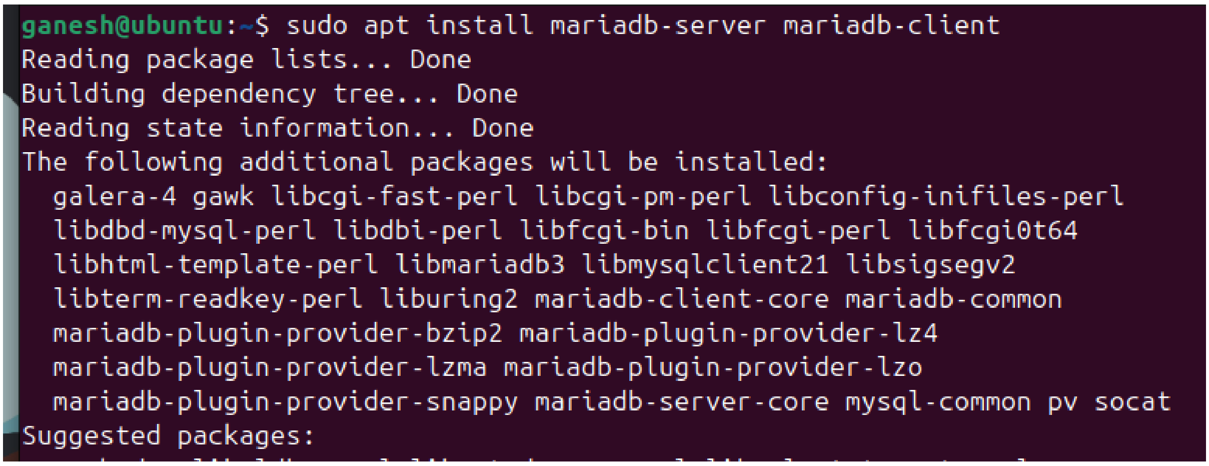
```
ganesh@ubuntu:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.4).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
ganesh@ubuntu:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/sy
stemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
ganesh@ubuntu:~$ sudo systemctl start apache2
```

4.) Install MySQL/MariaDB:

```
sudo apt install mysql-server
```

- mysql-server: This package installs the MySQL server, which is a popular relational database management system.
-
- mariadb-client: This is the package for the MariaDB client software, which allows you to interact with the MariaDB server. You can run SQL queries, manage databases, and perform administrative tasks from the command line using the

Output:-



```
ganesh@ubuntu:~$ sudo apt install mariadb-server mariadb-client
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  galera-4 gawk libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl
  libdbd-mysql-perl libdbi-perl libfcgi-bin libfcgi-perl libfcgi0t64
  libhtml-template-perl libmariadb3 libmysqlclient21 libsigsegv2
  libterm-readkey-perl liburing2 mariadb-client-core mariadb-common
  mariadb-plugin-provider-bzip2 mariadb-plugin-provider-lz4
  mariadb-plugin-provider-lzma mariadb-plugin-provider-lzo
  mariadb-plugin-provider-snappy mariadb-server-core mysql-common pv socat
Suggested packages:
```

TILAK MAHARASHTRA VIDYAPEETH

PUNE

विद्या बोधकरी यशःसुखकरी विद्या गुरूणां गुरुः

Estd. 1921

Thank you.