Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavre

A Lab Report

on

"Operating System Lab-I"

[Code No.: COMP 307]

Submitted by

Rajat Dahal
Roll No.: 13

Submitted to

Ms. Rabina Shrestha
Department of Computer Science and Engineering

December 10, 2025

Questions

Q1: What is Linux?

Linux is a family of open-source, Unix-like operating systems based on the Linux kernel, developed by Linus Torvalds and first released on September 17, 1991. It handles system hardware, runs programs, and provides a secure environment for multitasking. It is widely used in servers, desktops, embedded systems, and even supercomputers. Popular Linux distributions include Arch-Linux, Ubuntu, RedHat, and many others.

Q2: The Linux Hierarchical File System

Linux organizes files using a hierarchical directory structure that starts at the root directory /. In Linux, everything is treated as a file or directory, and every path traces back to /. Frequently used directories include:

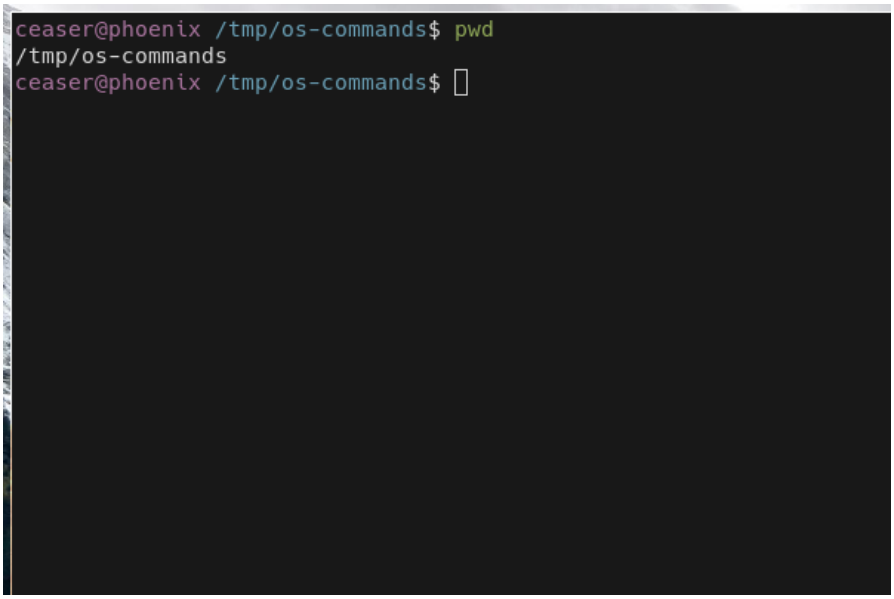
- / – Root Directory of the entire file system hierarchy
- /home – Contains personal directories and files for normal users
- /bin – Essential command binaries (executable programs) for all users
- /sbin – Essential system binaries (executable programs) for system administration
- /etc – Contains host-specific configuration files for system-wide settings
- /dev – Contains device files representing hardware components
- /proc – A virtual filesystem providing process and kernel information
- /tmp – Holds temporary files created by the system and users
- /usr – Contains the bulk of user applications, libraries, and documentation
- /var – Stores variable data like logs, caches, and mail spool files
- /boot – Contains the files needed to boot the system, including the kernel

Q3: Importance of Linux commands in Operating Systems

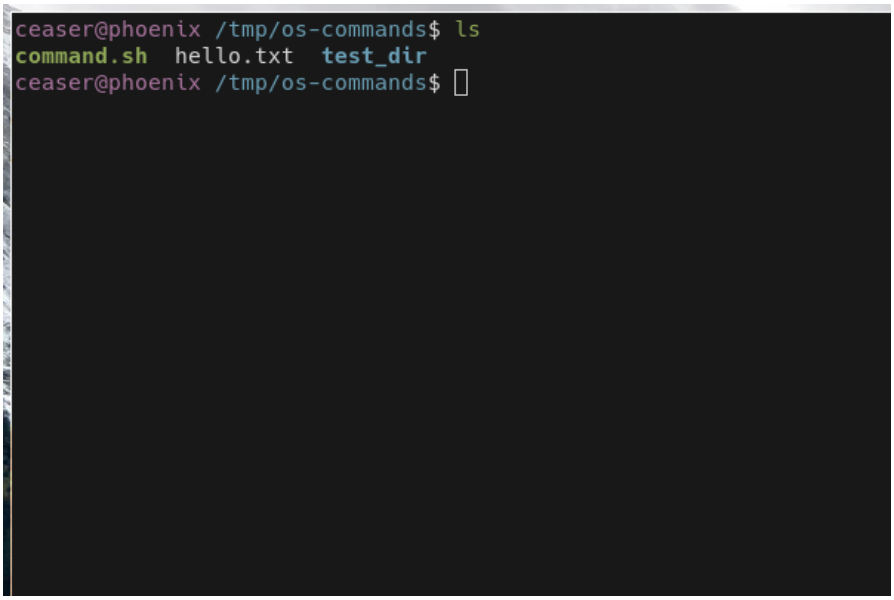
Linux commands are essential because they offer a powerful and direct way to interact with the operating system. They help users and administrators explore the file system, manage files and folders, inspect system performance, and automate workflows through scripts. Compared to graphical interfaces, command-line tools are faster, more efficient, and offer greater precision. Learning these commands improves productivity, troubleshooting skills, and overall understanding of system behavior, making them extremely valuable for developers, system administrators, and advanced users.

Linux Commands

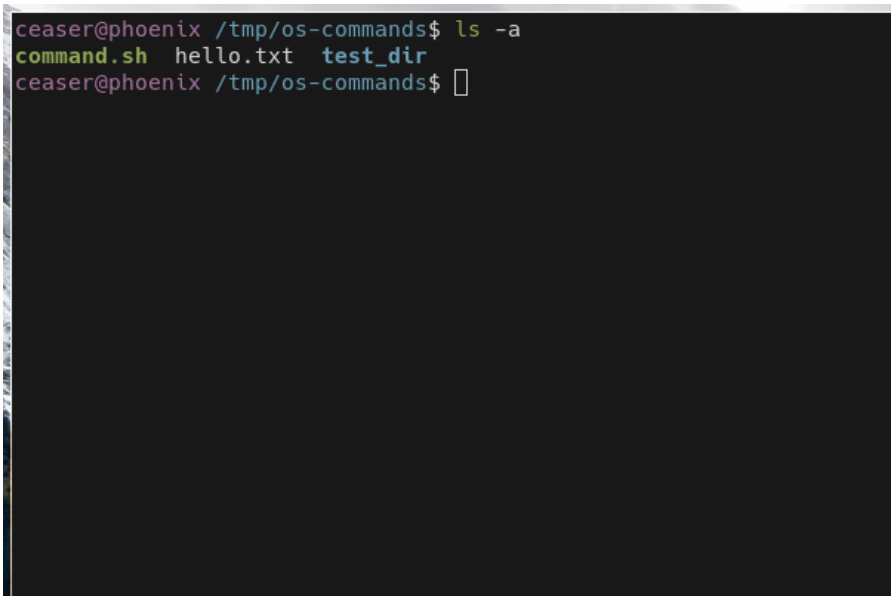
1. **`pwd`**
   The pwd command shows the directory you are currently in. It helps identify your exact location in the Linux filesystem, which is useful when moving across folders, writing scripts, or verifying paths before performing operations. Since Linux follows a root-based directory structure, pwd ensures you always know your working directory.



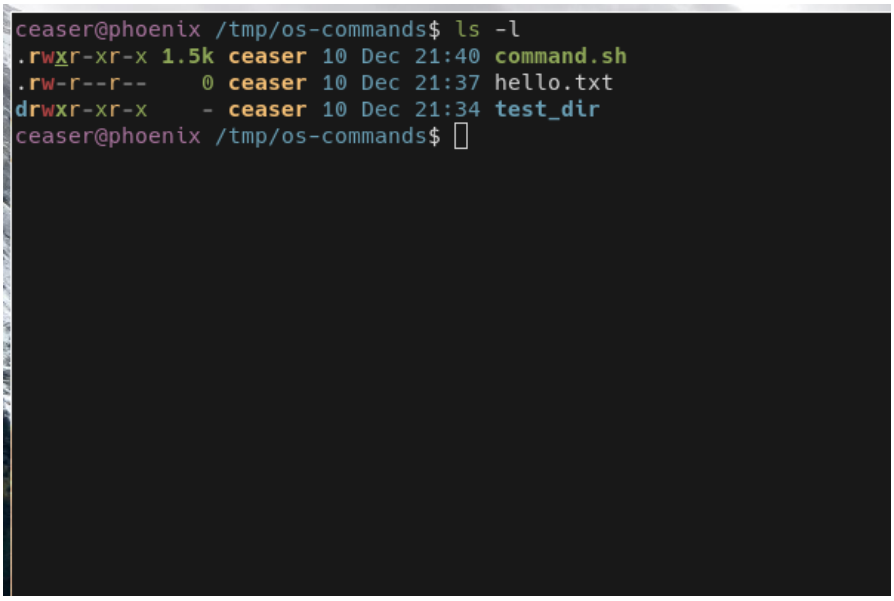
2. **`ls`**
   The ls command displays the files and directories present in your current folder. It provides a quick view of directory contents and is one of the most commonly used commands. By default, it only shows non-hidden files.



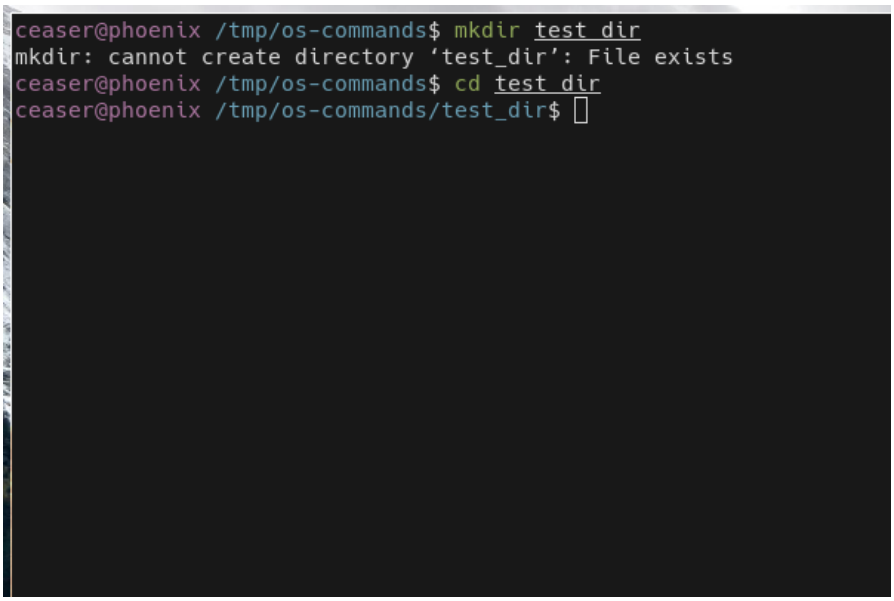
3. **`ls -a`**
   This variation of ls lists every file, including hidden ones. Hidden files in Linux begin with a dot (.), such as .bashrc or .config, and typically store configuration data.



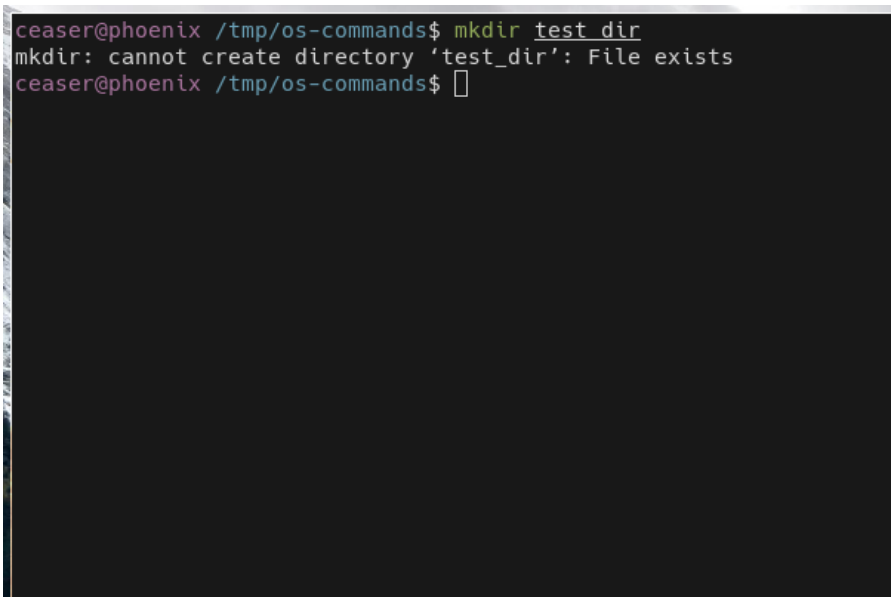
4. **`ls -l`**
   Using -l shows a detailed listing format. It includes permissions, owner and group information, file size, and the last modification timestamp. This view is especially useful for managing permissions and understanding file security.



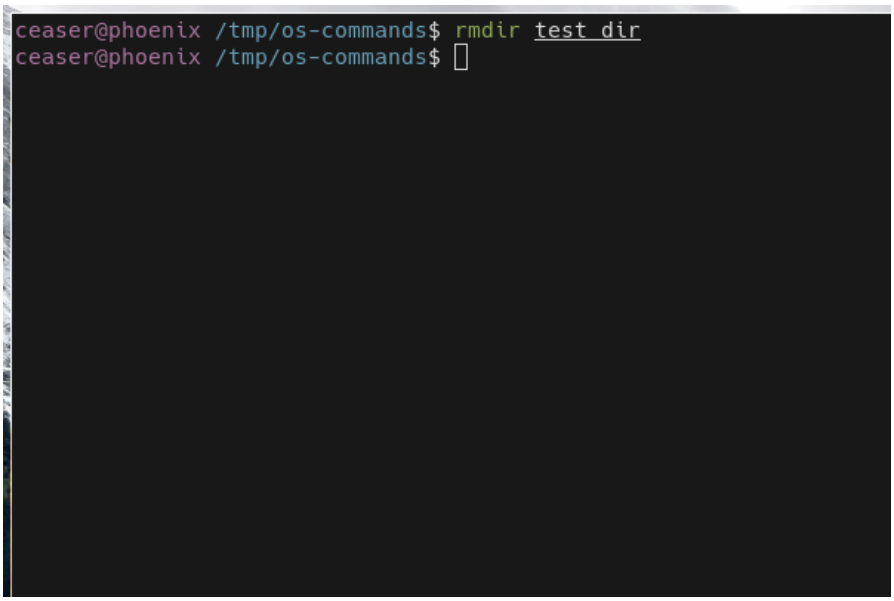
5. **`cd`**
   The cd command is used to change directories. It allows movement throughout the Linux file system. You can enter subdirectories, go back with `cd ..`, or jump to a specific directory using an absolute path.



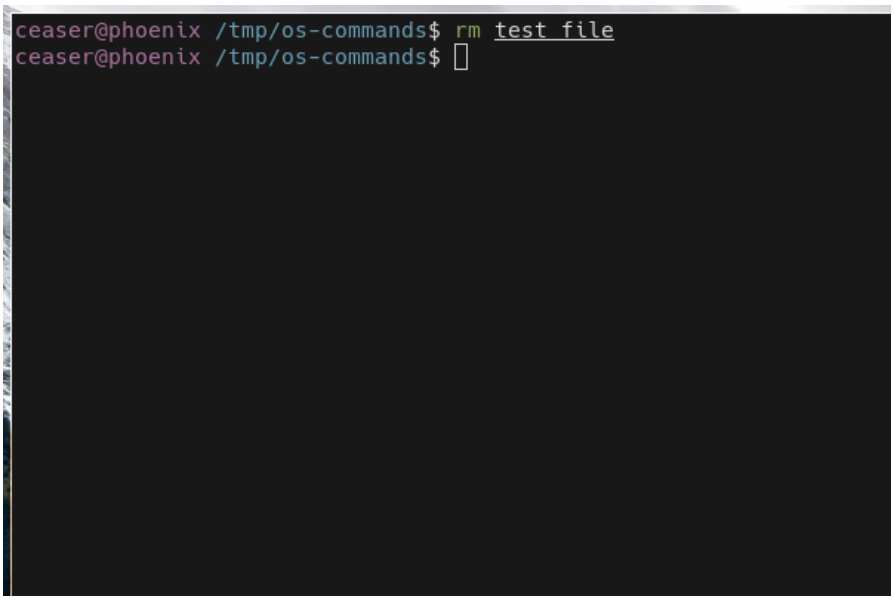
6. **`mkdir`**
   mkdir creates new directories. It is useful for organizing files by grouping them into folders. You can create several directories at once or even nested directory structures using `mkdir -p`.



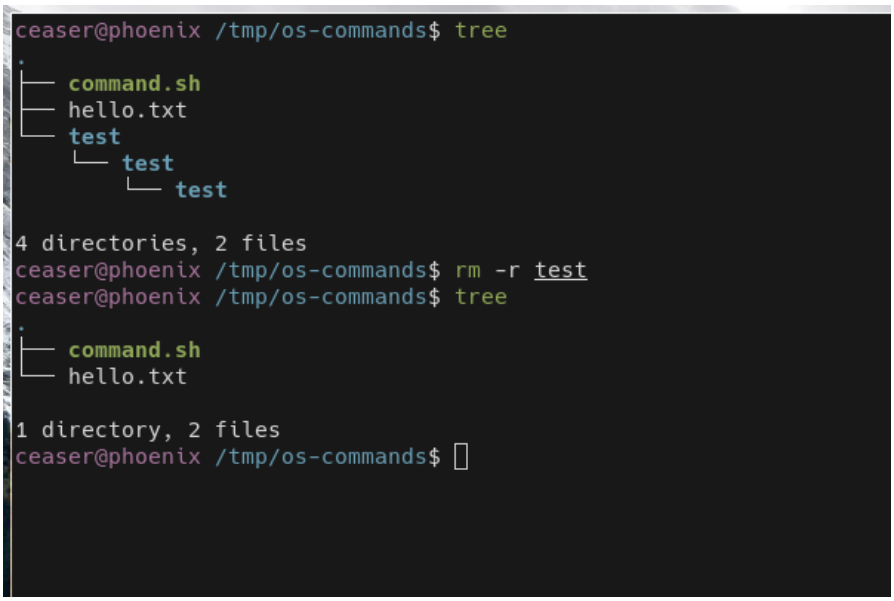
7. **`rmdir`**
   This command removes an empty directory. It cannot delete directories that still contain files. It is mainly used for deleting unused or temporary empty folders. A directory must already exist before it can be removed.



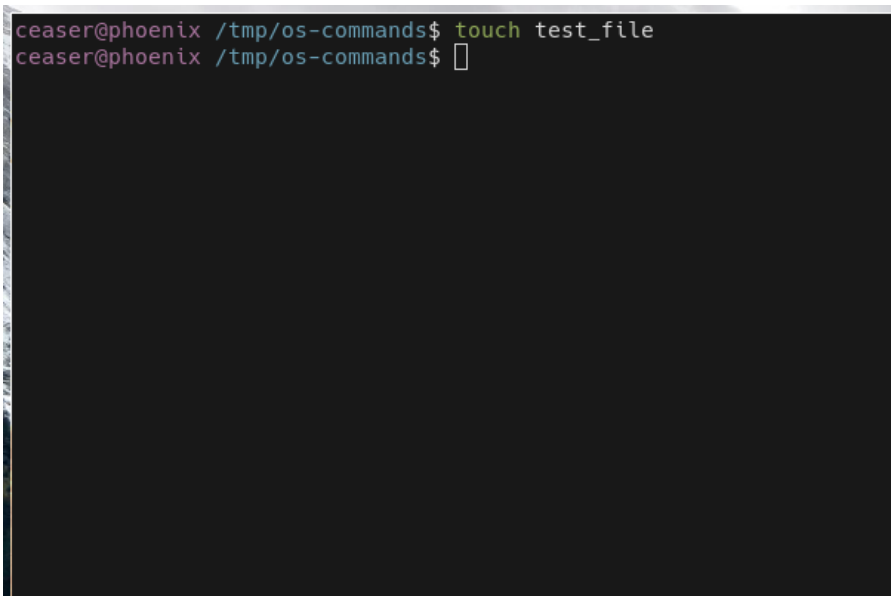
8. **`rm`**
   The rm command permanently deletes files. There is no recycle bin in Linux, so deleted items are not easily recovered. It can also remove multiple files at once.



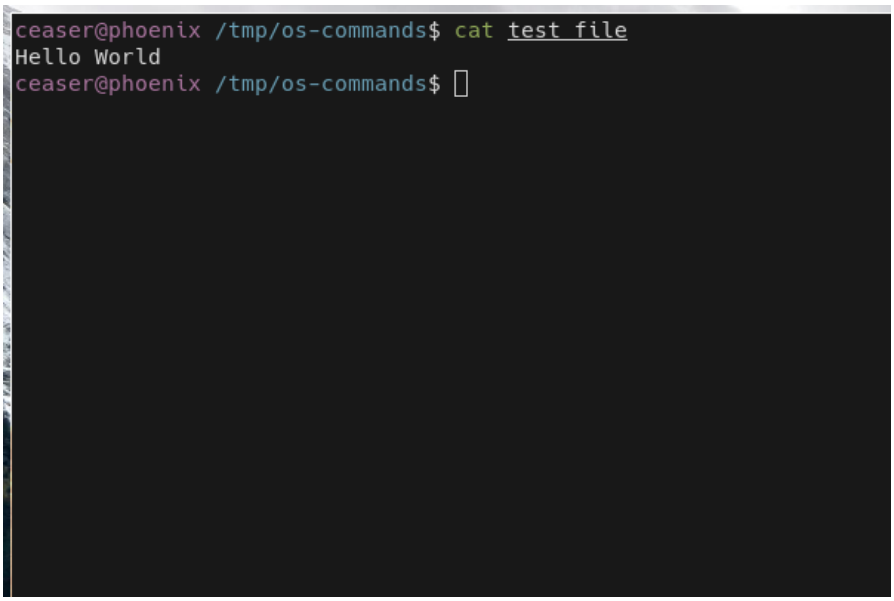
9. **`rm -r folder_name`**
   The -r option enables recursive deletion, meaning it removes a directory along with all its contents—files and subdirectories. This command is powerful and must be used carefully, as it will erase everything inside the specified folder.



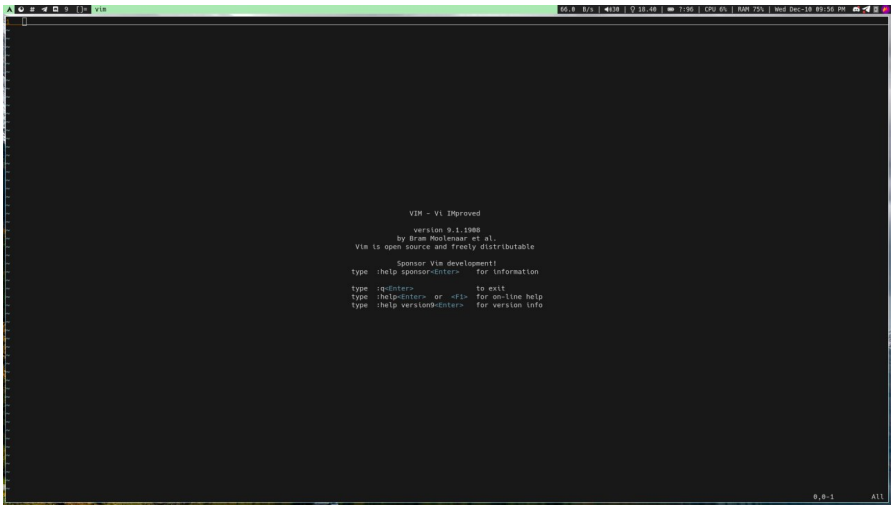
10. **`touch`**
    touch is used to create a new empty file or refresh the timestamp of an existing one. It is often used in scripting or to set up placeholder files.



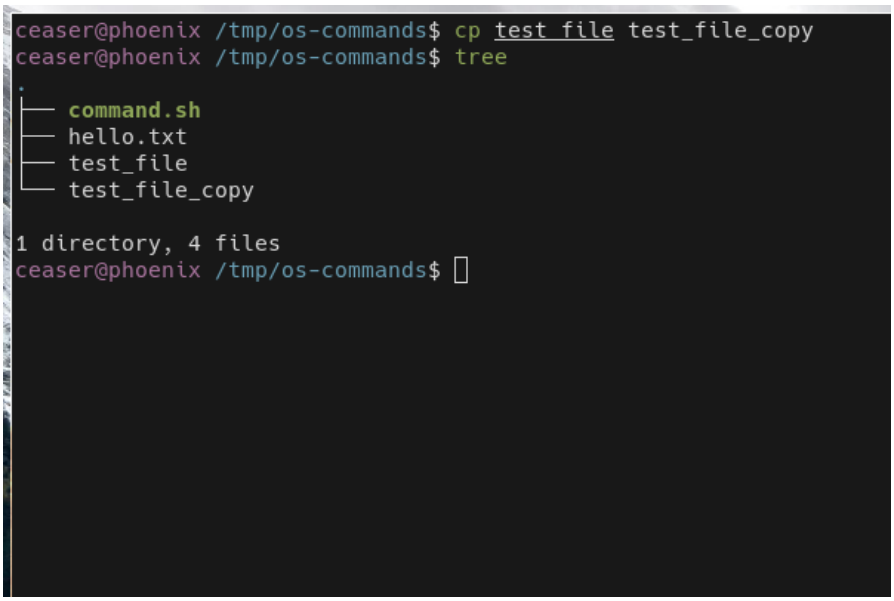
11. **`cat`**
    The cat command outputs the contents of a file to the terminal. It is also capable of merging files or creating new ones when paired with redirection operators.



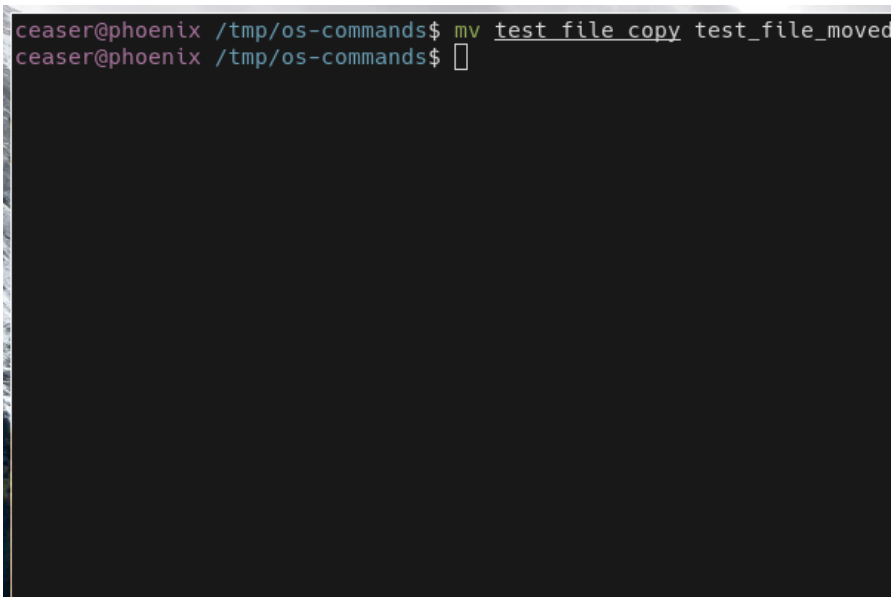
12. **`nano, vi, jed`**
    These are command-line text editors. nano is simple and user-friendly, vi (or vim) is a robust editor favored by programmers, and jed provides a lightweight editing experience. They allow creating and modifying files directly from the terminal.



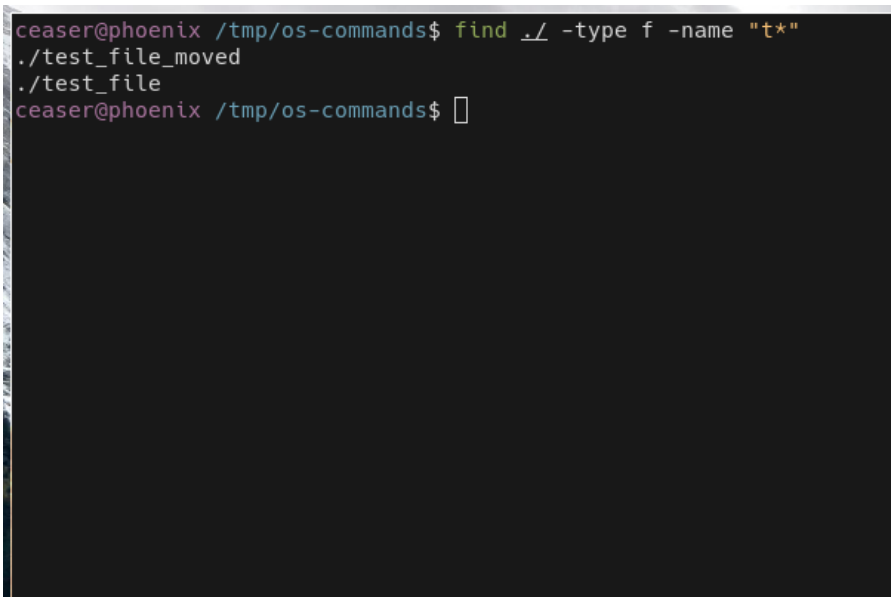
13. **`cp`**
    cp is used to copy files from one location to another. Directories can also be copied using the -r flag. This command is essential for backups, file duplication, and organization.



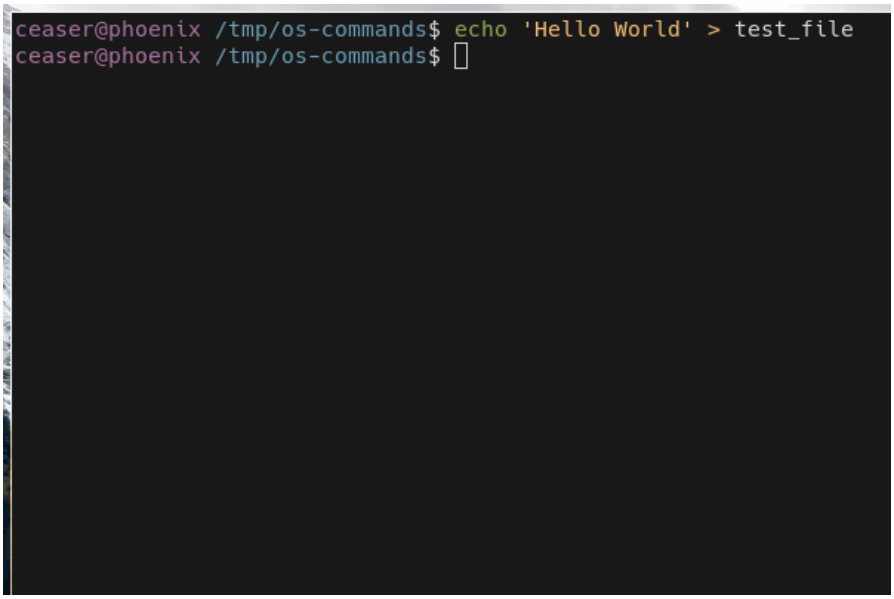
14. **`mv`**
    mv moves or renames files and directories. Renaming is just moving a file within the same folder with a new name. It is commonly used for reorganizing or updating file names.



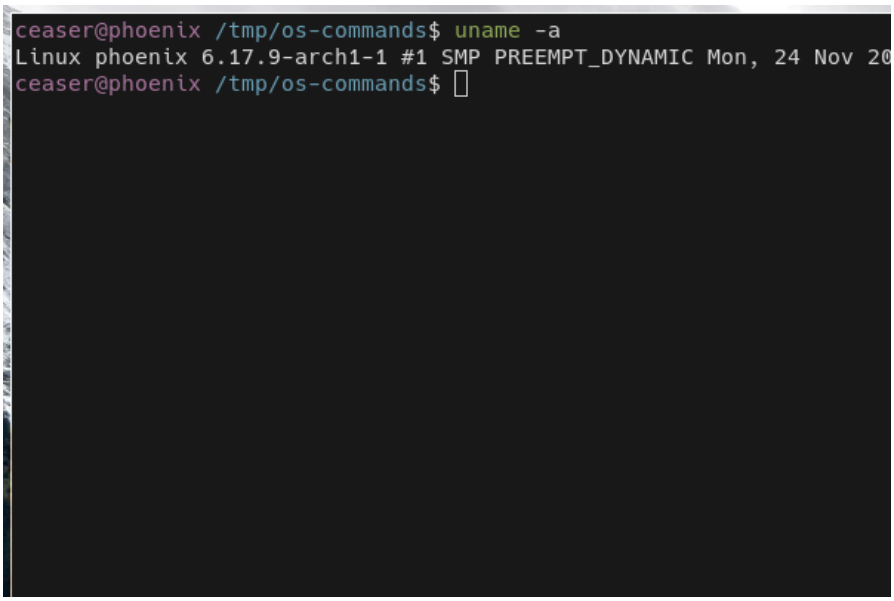
15. **`locate`**
    This command finds files based on their name. It uses a prebuilt system database, making searches extremely fast. It is helpful when looking for misplaced or forgotten files.



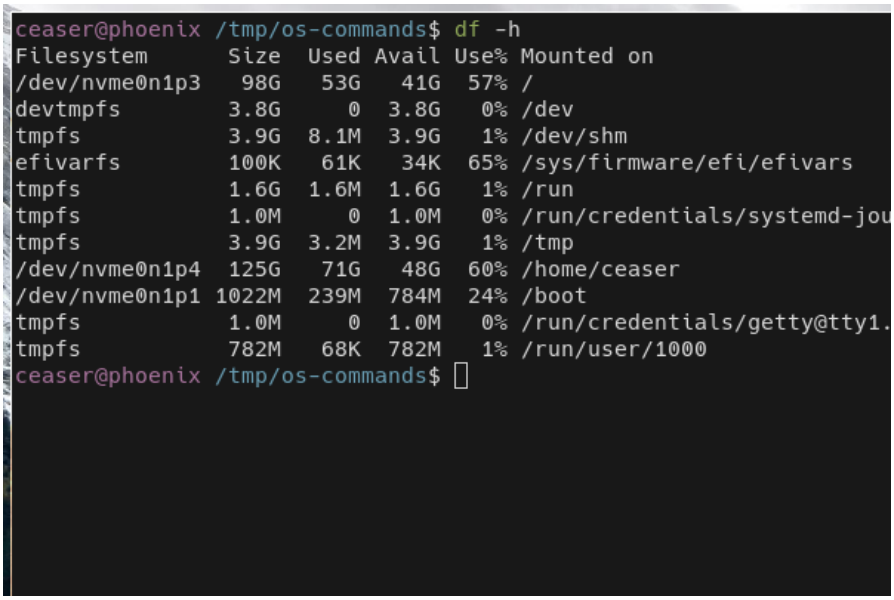
16. **`echo`**
    echo prints text or variable values to the terminal. It is frequently used in scripts to display messages, check variable states, or write text into files via redirection.



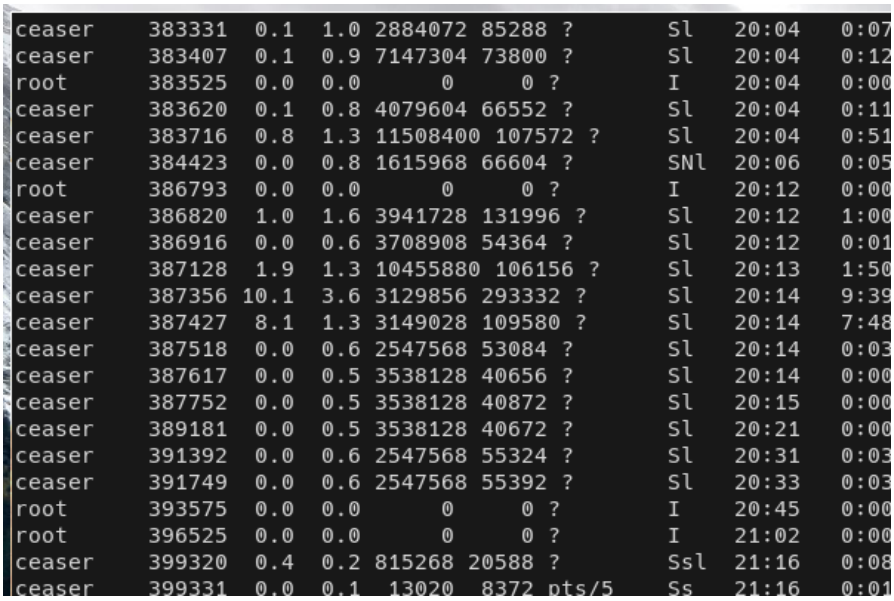
17. **`uname -a`**
    Shows full details about the system, such as kernel version, hardware architecture, operating system name, and hostname. It is useful for diagnosing issues and checking system information.



18. **`df -h`**
    Displays disk usage in a human-readable format (such as MB or GB). It shows available and used space, total capacity, and mounted file systems. It is useful for monitoring disk storage.

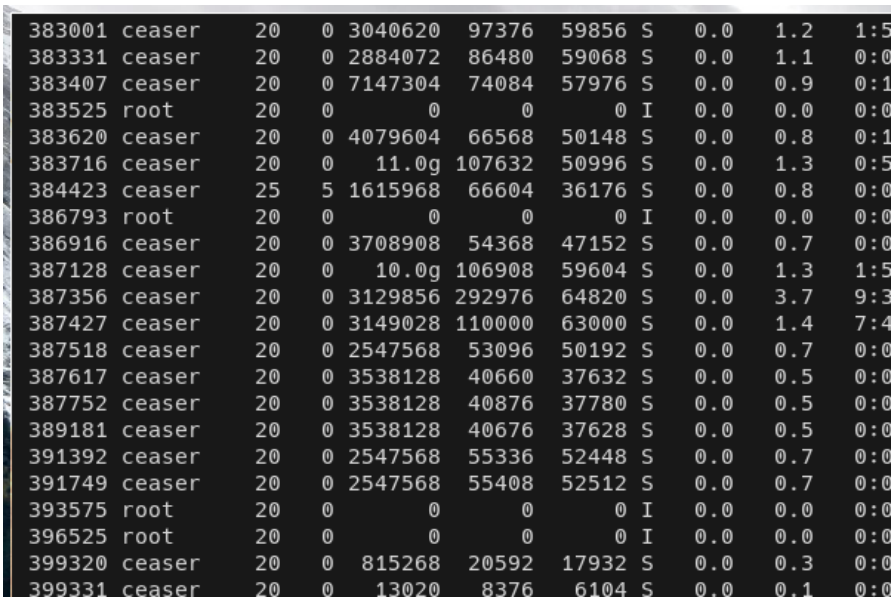


19. **`ps -u $USER`**
    Lists all active processes belonging to the current user. It displays process IDs, CPU usage, memory consumption, and the command that launched them. This helps identify unnecessary or frozen processes.



```
ceaser    383331  0.1  1.0 2884072 85288 ?       Sl   20:04   0:07
ceaser    383407  0.1  0.9 7147304 73800 ?       Sl   20:04   0:12
root      383525  0.0  0.0       0     0 ?       I    20:04   0:00
ceaser    383620  0.1  0.8 4079604 66552 ?       Sl   20:04   0:11
ceaser    383716  0.8  1.3 11508400 107572 ?     Sl   20:04   0:51
ceaser    384423  0.0  0.8 1615968 66604 ?       SNl  20:06   0:05
root      386793  0.0  0.0       0     0 ?       I    20:12   0:00
ceaser    386820  1.0  1.6 3941728 131996 ?      Sl   20:12   1:00
ceaser    386916  0.0  0.6 3708908 54364 ?       Sl   20:12   0:01
ceaser    387128  1.9  1.3 10455880 106156 ?     Sl   20:13   1:50
ceaser    387356 10.1  3.6 3129856 293332 ?      Sl   20:14   9:39
ceaser    387427  8.1  1.3 3149028 109580 ?      Sl   20:14   7:48
ceaser    387518  0.0  0.6 2547568 53084 ?       Sl   20:14   0:03
ceaser    387617  0.0  0.5 3538128 40656 ?       Sl   20:14   0:00
ceaser    387752  0.0  0.5 3538128 40872 ?       Sl   20:15   0:00
ceaser    389181  0.0  0.5 3538128 40672 ?       Sl   20:21   0:00
ceaser    391392  0.0  0.6 2547568 55324 ?       Sl   20:31   0:03
ceaser    391749  0.0  0.6 2547568 55392 ?       Sl   20:33   0:03
root      393575  0.0  0.0       0     0 ?       I    20:45   0:00
root      396525  0.0  0.0       0     0 ?       I    21:02   0:00
ceaser    399320  0.4  0.2 815268 20588 ?        Ssl  21:16   0:08
ceaser    399331  0.0  0.1  13020  8372 pts/5    Ss   21:16   0:01
```

20. **`top`**
    Shows real-time system resource usage. It provides information about running processes, CPU load, memory consumption, and system uptime. It is one of the most important commands for performance monitoring.



```
 383001 ceaser    20   0 3040620  97376  59856 S   0.0   1.2   1:5
 383331 ceaser    20   0 2884072  86480  59068 S   0.0   1.1   0:0
 383407 ceaser    20   0 7147304  74084  57976 S   0.0   0.9   0:1
 383525 root      20   0       0      0      0 I   0.0   0.0   0:0
 383620 ceaser    20   0 4079604  66568  50148 S   0.0   0.8   0:1
 383716 ceaser    20   0   11.0g 107632  50996 S   0.0   1.3   0:5
 384423 ceaser    25   5 1615968  66604  36176 S   0.0   0.8   0:0
 386793 root      20   0       0      0      0 I   0.0   0.0   0:0
 386916 ceaser    20   0 3708908  54368  47152 S   0.0   0.7   0:0
 387128 ceaser    20   0   10.0g 106908  59604 S   0.0   1.3   1:5
 387356 ceaser    20   0 3129856 292976  64820 S   0.0   3.7   9:3
 387427 ceaser    20   0 3149028 110000  63000 S   0.0   1.4   7:4
 387518 ceaser    20   0 2547568  53096  50192 S   0.0   0.7   0:0
 387617 ceaser    20   0 3538128  40660  37632 S   0.0   0.5   0:0
 387752 ceaser    20   0 3538128  40876  37780 S   0.0   0.5   0:0
 389181 ceaser    20   0 3538128  40676  37628 S   0.0   0.5   0:0
 391392 ceaser    20   0 2547568  55336  52448 S   0.0   0.7   0:0
 391749 ceaser    20   0 2547568  55408  52512 S   0.0   0.7   0:0
 393575 root      20   0       0      0      0 I   0.0   0.0   0:0
 396525 root      20   0       0      0      0 I   0.0   0.0   0:0
 399320 ceaser    20   0  815268  20592  17932 S   0.0   0.3   0:0
 399331 ceaser    20   0   13020   8376   6104 S   0.0   0.1   0:0
```

21. **chmod**
    Changes file or directory permissions. Permissions determine who can read, write, or execute a file. chmod is essential when running scripts, protecting sensitive files, or managing user access.