**Index page**

**Name: liza bhor**

**Enroll no:0827CI221083**

**Class:Csit 2**

**Sub:linux lab**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **labs** | **Experiment date** | **Submission date** | **Page number** | **remark** |
| **Lab1** | **19/09/24** |  | **1 to 17** |  |
| **Lab2** | **20/09/24** |  | **17 to 32** |  |
| **Lab3** | **20/09/24** |  | **32 to 38** |  |
| **Lab4** | **20/09/24** |  | **38 to 47** |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Lab 1:Theory of linux

**# introduction of OS**

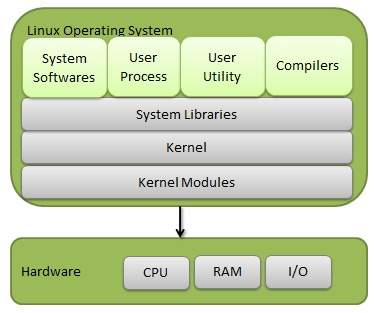
The Linux Operating System is a type of operating system that is similar to Unix, and it is built upon the Linux Kernel. The Linux Kernel is like the brain of the operating system because it manages how the computer interacts with its hardware and resources. It makes sure everything works smoothly and efficiently. But the Linux Kernel alone is not enough to make a complete operating system. To create a full and functional system, the Linux Kernel is combined with a collection of software packages and utilities, which are together called Linux distributions. These distributions make the Linux Operating System ready for users to run their applications and perform tasks on their computers securely and effectively. Linux distributions come in different flavors, each tailored to suit the specific needs and preferences of users.

Developed by Linus Torvalds in 1991, the Linux operating system is a powerful and flexible open-source software platform. It acts as the basis for a variety of devices, such embedded systems, cell phones, servers, and personal computers. Linux, that’s well-known for its reliability, safety, and flexibility, allows users to customize and improve their environment to suit specific needs. With an extensive and active community supporting it, Linux is an appealing choice for people as well as companies due to its wealth of resources and constant developments.

**Components of Linux System:**

Linux Operating System has primarily three components

* **Kernel** − Kernel is the core part of Linux. It is responsible for all major activities of this operating system. It consists of various modules and it interacts directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.
* **System Library** − System libraries are special functions or programs using which application programs or system utilities accesses Kernel's features. These libraries implement most of the functionalities of the operating system and do not requires kernel module's code access rights.
* **System Utility** − System Utility programs are responsible to do specialized, individual level tasks.



**#Services of OS**

An operating system is software that acts as an intermediary between the user and computer hardware. It is a program with the help of which we are able to run various applications. It is the one program that is running all the time. Every computer must have an operating system to smoothly execute other programs.

The OS coordinates the use of the hardware and application programs for various users. It provides a platform for other application programs to work. The operating system is a set of special programs that run on a computer system that allows it to work properly. It controls input-output devices, execution of programs, managing files, etc.

**Services of Operating System**

* Program execution
* Input Output Operations
* Communication between Process
* File Management
* Memory Management
* Process Management
* Security and Privacy
* Resource Management
* User Interface
* Networking
* Error handling
* Time Management

**Program Execution**

It is the Operating System that manages how a program is going to be executed. It loads the program into the memory after which it is executed. The order in which they are executed depends on the CPU Scheduling Algorithms. A few are [FCFS](https://www.geeksforgeeks.org/first-come-first-serve-cpu-scheduling-non-preemptive/), [SJF](https://www.geeksforgeeks.org/sjf-full-form/), etc. When the program is in execution, the Operating System also handles deadlock i.e. no two processes come for execution at the same time. The Operating System is responsible for the smooth execution of both user and system programs. The Operating System utilizes various resources available for the efficient running of all types of functionalities.

**Input Output Operations**

Operating System manages the input-output operations and establishes communication between the user and [device drivers](https://www.geeksforgeeks.org/device-drivers-in-linux/). Device drivers are software that is associated with hardware that is being managed by the OS so that the sync between the devices works properly. It also provides access to input-output devices to a program when needed.

**Communication Between Processes**

The Operating system manages the communication between processes. Communication between processes includes data transfer among them. If the processes are not on the same computer but connected through a computer network, then also their communication is managed by the Operating System itself.

**File Management**

The operating system helps in managing files also. If a program needs access to a file, it is the operating system that grants access. These permissions include read-only, read-write, etc. It also provides a platform for the user to create, and delete files. The Operating System is responsible for making decisions regarding the storage of all types of data or files, i.e, [floppy disk](https://www.geeksforgeeks.org/what-is-a-floppy-disk/)/hard disk/pen drive, etc. The Operating System decides how the data should be manipulated and stored.

**Memory Management**

Let’s understand memory management by OS in simple way. Imagine a cricket team with limited number of player . The team manager (OS) decide whether the upcoming player will be in playing 11 ,playing 15 or will not be included in team , based on his performance . In the same way, OS first check whether the upcoming program fulfil all requirement to get memory space or not ,if all things good, it checks how much memory space will be sufficient for program and then load the program into memory at certain location. And thus , it prevents program from using unnecessary memory.

**Process Management**

Let’s understand the process management in unique way. Imagine, our kitchen stove as the (CPU) where all cooking(execution) is really happen and chef as the (OS) who uses kitchen-stove(CPU) to cook different dishes(program). The chef(OS) has to cook different dishes(programs) so he ensure that any particular dish(program) does not take long time(unnecessary time) and all dishes(programs) gets a chance to cooked(execution) .The chef(OS) basically scheduled time for all dishes(programs) to run kitchen(all the system) smoothly and thus cooked(execute) all the different dishes(programs) efficiently.

**Security and Privacy**

* **Security :**OS keep our computer safe from an unauthorized user by adding security layer to it. Basically, Security is nothing but just a layer of protection which protect computer from bad guys like [viruses](https://www.geeksforgeeks.org/what-is-a-computer-virus/) and hackers. OS provide us defenses like [firewalls](https://www.geeksforgeeks.org/introduction-of-firewall-in-computer-network/) and anti-virus software and ensure good safety of computer and personal information.
* **Privacy :**OS give us facility to keep our essential information hidden like having a lock on our door, where only you can enter and other are not allowed . Basically , it respect our secrets and provide us facility to keep it safe.

**Resource Management**

System resources are shared between various processes. It is the Operating system that manages resource sharing. It also manages the CPU time among processes using [CPU Scheduling Algorithms](https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/). It also helps in the [memory management](https://www.geeksforgeeks.org/memory-management-in-operating-system/) of the system. It also controls input-output devices. The OS also ensures the proper use of all the resources available by deciding which resource to be used by whom.

**User Interface**

User interface is essential and all operating systems provide it. Users either interacts with the operating system through the [command-line interface](https://www.geeksforgeeks.org/linux-operating-system-cli-command-line-interface-and-gui-graphic-user-interface/) or graphical user interface or GUI. The command interpreter executes the next user-specified command.

A GUI offers the user a mouse-based window and menu system as an interface.

**Networking**

This service enables communication between devices on a network, such as connecting to the internet, sending and receiving data packets, and managing network connections.

**Error Handling**

The Operating System also handles the error occurring in the [CPU](https://www.geeksforgeeks.org/difference-between-cpu-and-gpu/), in Input-Output devices, etc. It also ensures that an error does not occur frequently and fixes the errors. It also prevents the process from coming to a deadlock. It also looks for any type of error or bugs that can occur while any task. The well-secured OS sometimes also acts as a countermeasure for preventing any sort of breach of the Computer System from any external source and probably handling them.

**Time Management**

Imagine traffic light as (OS), which indicates all the cars(programs) whether it should be stop(red)=>(simple queue), start(yellow)=>(ready queue),move(green)=>(under execution) and this light (control) changes after a certain interval of time at each side of the road(computer system) so that the cars(program) from all side of road move smoothly without traffic.



**#Need of operating system**



An operating system (OS) is a critical layer of software that manages computer hardware and software resources and provides common services for computer programs. In the context of Linux, an open-source operating system, its necessity and benefits can be summarized as follows:

**1. Resource Management**

* **Hardware Management:** The OS acts as an intermediary between applications and the hardware, managing resources such as CPU, memory, disk space, and I/O devices.
* **Concurrency Control**: It handles the execution of multiple processes, ensuring that they operate independently while sharing the system's resources.

**2. User Interface**

* **Command-Line and GUI**: Linux provides various user interfaces, including command-line interfaces (CLI) and graphical user interfaces (GUI), allowing users to interact with the system easily.
* **Customization:** Users can choose from various desktop environments (e.g., GNOME, KDE) to customize the look and feel of their Linux OS.

**3.File System Management**

* **Hierarchical File System:** Linux uses a structured file system that organizes files and directories hierarchically, making it easier to store and retrieve information.
* **File Permissions:** The OS manages file permissions and security, ensuring that only authorized users can access or modify files.

**4. Process Management**

* **Task Scheduling:** The OS controls the execution of processes, scheduling them efficiently to maximize CPU usage and responsiveness.
* **Multitasking**: Linux supports multitasking, allowing multiple applications to run simultaneously without interference.

**5.Networking**

* **Networking Capabilities**: Linux provides robust networking capabilities, enabling communication between devices over local and wide area networks.
* **Protocols Support**: It supports various networking protocols (e.g., TCP/IP), facilitating internet connectivity and services.

**6.Security**

* **User Accounts and Permissions**: Linux implements a multi-user environment, where each user has specific permissions, enhancing security and preventing unauthorized access.
* **Firewall and Security Feature**: The OS includes built-in security tools and configurations (such as iptables) to protect against unauthorized access and attacks.

**7.Software Management**

* **Package Management:** Linux distributions come with package managers (e.g., APT for Debian/Ubuntu, YUM/DNF for Red Hat/Fedora) to install, update, and manage software easily.
* O**pen Source Software**: Users have access to thousands of open-source applications, promoting collaboration and innovation.

**8. Performance and Efficiency**

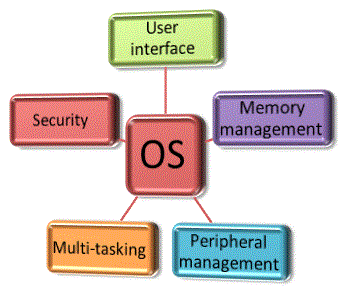
* **Lightweight:** Many Linux distributions can run on older hardware efficiently, making it suitable for a wide range of devices.
* **Customization:** Users can optimize their environments by choosing lightweight window managers or compiling their kernels for specific hardware.

**9.Development and Community Support**

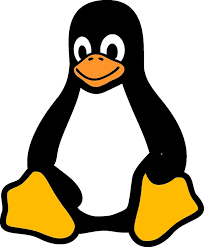
* **Developer-Friendly:** Linux is widely used by developers due to its flexibility, support for various programming languages, and extensive tools and libraries.
* **Active Community:** A large, active community contributes to constant improvements, security updates, and support for users and developers alike.

**10.Conclusion**

In summary, an operating system is essential for managing infrastructure, providing user interfaces, facilitating processes, enabling networking, maintaining security, managing software, and improving system performance. Linux, as an open-source OS, offers these functionalities while promoting flexibility, customization, and collaboration within a supportive community.



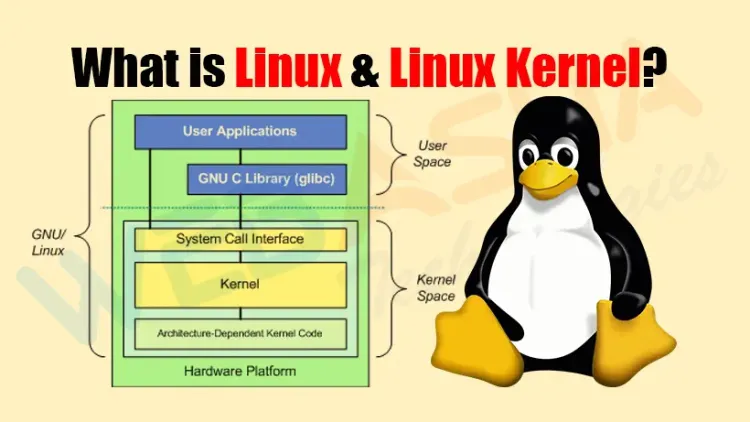
**#what is linux**



Just like Windows, iOS, and Mac OS, Linux is an operating system. In fact, one of the most popular platforms on the planet, Android, is powered by the Linux operating system. An operating system is software that manages all of the hardware resources associated with your desktop or laptop. To put it simply, the operating system manages the communication between your software and your hardware. Without the operating system (OS), the software wouldn’t function.

The Linux operating system comprises several different pieces:

1. **Bootloader** –  The software that manages the boot process of your computer. For most users, this will simply be a splash screen that pops up and eventually goes away to boot into the operating system.
2. **Kernel** – This is the one piece of the whole that is actually called ‘Linux’. The kernel is the core of the system and manages the CPU, memory, and peripheral devices. The kernel is the lowest level of the OS.
3. **Init system** – This is a sub-system that bootstraps the user space and is charged with controlling daemons. One of the most widely used init systems is systemd, which also happens to be one of the most controversial. It is the init system that manages the boot process, once the initial booting is handed over from the bootloader (i.e., GRUB or GRand Unified Bootloader).
4. **Daemons** – These are background services (printing, sound, scheduling, etc.) that either start up during boot or after you log into the desktop.
5. **Graphical server** – This is the sub-system that displays the graphics on your monitor. It is commonly referred to as the X server or just X.
6. **Desktop environment –** This is the piece that the users actually interact with. There are many desktop environments to choose from (GNOME, Cinnamon, Mate, Pantheon, Enlightenment, KDE, Xfce, etc.). Each desktop environment includes built-in applications (such as file managers, configuration tools, web browsers, and games).
7. **Applications –** Desktop environments do not offer the full array of apps. Just like Windows and macOS, Linux offers thousands upon thousands of high-quality software titles that can be easily found and installed. Most modern Linux distributions (more on this below) include App Store-like tools that centralize and simplify application installation. For example, Ubuntu Linux has the Ubuntu Software Center (a rebrand of GNOME Software) which allows you to quickly search among the thousands of apps and install them from one centralized location.



**#Need of linux**

Linux, an open-source operating system, has gained popularity for various reasons, making it an essential choice for many users and organizations. Here are some key needs and benefits of using Linux:

**1. Open Source Nature**: Linux is free and open-source, which means anyone can use, modify, and distribute it. This encourages collaboration and innovation.

**2. Cost-Effective**: Many Linux distributions are available for free, reducing costs for individuals and organizations compared to proprietary operating systems that require purchasing licenses.

**3. Stability and Reliability:** Linux is known for its stability and performance. It can run for extended periods without needing a reboot, making it an ideal choice for servers and critical systems.

**4.Security:** Linux has a robust security model and benefits from a large community that actively patches vulnerabilities. It's less targeted by malware than some other operating systems.

**5.Flexibility and Customization:** Users can choose from various distributions (distros) tailored for different use cases, whether for desktops, servers, or embedded systems. Users can also customize their environments to meet specific needs.

**6.Performance:** Linux can perform well on a wide range of hardware, including older or less powerful machines. This makes it accessible for various users, from casual to professional.

**7.Development Environment:** Linux is favored by developers and system administrators because it provides powerful tools for programming, scripting, and automation. Many programming languages and frameworks are readily available.

**8. Community Support:** A vast community of users and developers offers extensive support through forums, documentation, and tutorials, making it easier for newcomers to learn and troubleshoot.

**9.Compatibility with Servers: Many** web servers run on Linux due to its efficiency, security, and robustness. This makes it a preferred choice for hosting websites and applications.

**10. Privacy:** Unlike some proprietary operating systems, many Linux distributions put a strong emphasis on user privacy and do not collect personal data.

**11.Multiple Desktop Environments**: Linux offers multiple desktop environments (like GNOME, KDE, Xfce), allowing users to choose an interface that suits their preferences and workflow.

**12. Continuous Updates:** The open-source community constantly improves Linux, adding new features, enhancements, and security updates regularly.

In summary, whether you are a casual user, a developer, a system administrator, or running a large enterprise, Linux provides a versatile, secure, and efficient operating system suited to a wide range of tasks. The decision to use Linux often depends on specific needs, preferences, and use cases.

**#linux distribution**



Linux distributions (or "distros") are various versions of the Linux operating system, each tailored for different uses and user experiences. Here are some of the most popular categories and examples of Linux distributions:

**1. General Purpose:**

* Ubuntu: One of the most popular desktop distros, known for its user-friendliness and extensive community support.
* Debian: A stable and versatile distro well-known for its package management and large software repository.
* Fedora: A cutting-edge distro that showcases the latest features of the Linux ecosystem, often used by developers and testers.

**2.Enterprise:**

* Red Hat Enterprise Linux (RHEL): A commercially supported distribution aimed at businesses and organizations.
* CentOS: A free alternative to RHEL, now part of the CentOS Stream project focusing on development and testing.
* SUSE Linux Enterprise: Another enterprise-focused distribution with strong support for cloud and server solutions.

3**.Lightweight Distros:**

* Lubuntu: A lighter version of Ubuntu, using the LXQt desktop environment, ideal for older hardware.
* Xubuntu: Another lightweight Ubuntu variant, this one using the XFCE desktop environment.
* Puppy Linux: Extremely lightweight and designed to run from RAM, making it suitable for very old systems.

**4.Security-Focused:**

* Kali Linux: A Debian-based distro specifically designed for penetration testing and security auditing.
* Tails: A privacy-focused distribution that runs from USB and leaves no trace on the host system.

**5. Rolling Release:**

* Arch Linux: A minimalist and flexible distro that follows a rolling release model, allowing users to have the latest software continuously.
* openSUSE Tumbleweed:The rolling release version of openSUSE, providing the latest stable versions of software.

**6.Educational:**

* Edubuntu: Ubuntu's educational spin, designed for use in classrooms and schools.
* K12Linux: A Linux distribution designed specifically for K-12 schools.

**7.Specialized**

* SteamOS: A Debian-based distro developed by Valve for gaming on Steam.
* Raspberry Pi OS (formerly Raspbian): Based on Debian and optimized for the Raspberry Pi hardware.

**8. Community-Driven**

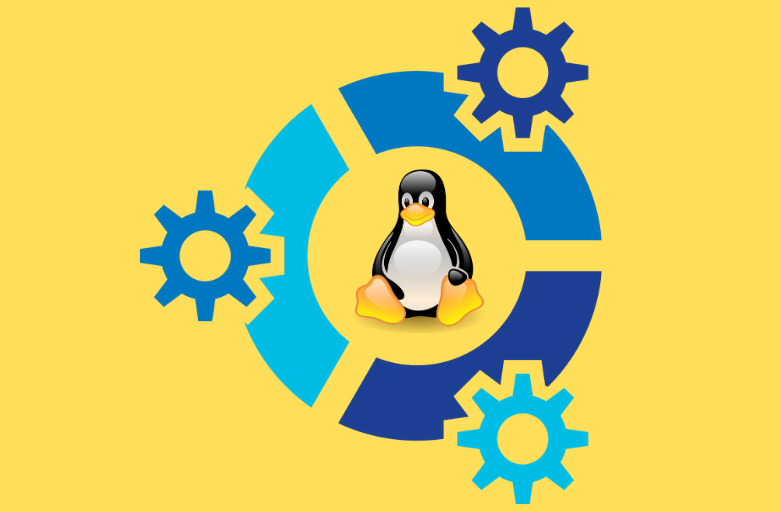
* Manjaro: A user-friendly Arch-based distribution that provides a more accessible entry point for users new to Arch.
* Linux Mint: Based on Ubuntu, it emphasizes usability and provides a familiar interface for users transitioning from Windows.

**Conclusion**

Linux distributions offer a vast range of options tailored to fit various user needs, from desktop use to server environments, from lightweight setups for older computers to comprehensive solutions for security professionals. The choice of a Linux distribution often depends on the user's experience level, needs, and preferences.



**#services of linux**



**Linux Services in brief:**

* [Custom .NET application Development on Linux using Mono](https://www.e-zest.com/mono-web-development/)
* [Linux Multi-Purpose business solution servers](https://www.e-zest.com/linux-multipurpose-servers)
* [Onsite or remote linux system monitoring and administration](https://www.e-zest.com/remote-server-administration)
* [Remote system / Server maintenance](https://www.e-zest.com/linux_server_maintenance)
* Low and high speed Co-Location services for startups and high bandwidth servers
* Upgrading the servers as needed to current patch levels to minimize security risks
* 24x7x365 system admin coverage – full-time, part-time, hourly basis
* Automated server monitoring and notification/paging system
* Administration of the basesystem Configuration and administration of network infrastructure.
* Messaging Solutions
* [Linux Antivirus Services](https://www.e-zest.com/linux-services)
* Mass Mailing Solutions and Administration services

**e-Zest linux Consultants are competent in the following areas of expertise:**

* Firewall solutions on top of OpenSource Operating Systems (as proxyserver for http, mail, ftp,…) , Firewall-ip chains
* Backup-concepts Automation by scripts (Shell, Perl)
* Dial-In solutions with Linux (Modem, ISDN)
* Databases: e.g. MySQL, PostgreSQL, IBM DB/2
* Mailserver: sendmail, Qmail, IMAP, POP3 ,Postfix
* Webserver and Web Technologies: Apache with/without SSL, Samba , PHP3, Servlets and JSP
* Fileserver: Samba, NFS, FTP LDAP: OpenLDAP
* Squid
* Radius
* DNS
* DHCP Server and Client
* File Transfer Protocol (FTP) Server
* Network File System (NFS) Server
* PPP Configuration
* RAID Configuration
* Network Interface Card (NIC) Configuration
* Lightweight Directory Access Protocol (LDAP)

e-Zest provides enterprise quality support for Linux by acting as a resource to match companies seeking Linux support including everything from system administration to software development to integration with other operating systems with qualified technicians who can help them, and all with a simple interface.

We will be expanding our support capabilities, and provide more services to our existing Linux customers. As usual, we will continue our reputation as Linux Specialists, for support, security, networks, and servers, as well as creating new channels to help all Linux users and companies.

If you, or your company is interested in having our team setup your Linux, or maintaining existing installations, or you need us to provide support agreements, for all versions of Linux, please feel free to contact our sales department or mail at [info@e-zest.com](mailto:info@e-zest.com).

**Other related services**

* [Linux Network Servers](https://www.e-zest.com/linux-network-servers)
* [Linux Security Firewall](https://www.e-zest.com/linux-security-firewall)
* [Linux Server Maintenance](https://www.e-zest.com/linux_server_maintenance)
* [Linux Server Antivirus](https://www.e-zest.com/linux-server-antivirus)
* [Workgroup Servers](https://www.e-zest.com/linux-workgroup-servers)

**Lab 2: Command in linux**

**#Use of Linux Commands**

**Linux commands** are a type of Unix command or [**shell**](https://www.geeksforgeeks.org/introduction-linux-shell-shell-scripting/) procedure. They are the basic tools used to interact with Linux on an individual level. Linux commands are used to perform a variety of tasks, including displaying information about files and directories.

[Linux operating system](https://www.geeksforgeeks.org/introduction-to-linux-operating-system/) is used on servers, desktops, and maybe even your smartphone. It has a lot of command line tools that can be used for virtually everything on the system.

All users should be familiar with most of these commands as they are required for most operating system tasks and computer programming.

Here we have put **all Basic** **Linux Commands** that every Linux user (**as a** **beginner**) should know. These are not all that you should know, but these are the basic and most commonly used commands.

# 1. Ls command :-

The ls command line utility lists all the files and directories under a specified directory. By default, ls uses the current directory and lists files and directories in alphabetical order by name. The ls command supports many flags that modify its behavior.

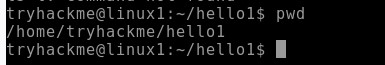
Syntax :- **ls**



# 2. Pwd command :-

Description. The pwd command writes to standard output the full path name of your current directory (from the root directory). All directories are separated by a / (slash). The root directory is represented by the first /, and the last directory named is your current directory.

Syntax :- **pwd**



3**. Mkdir command :-**

The mkdir command in Linux/Unix is a command-line utility that allows users to create new directories. mkdir stands for "make directory." With mkdir , you can also set permissions, create multiple directories at once, and much more.

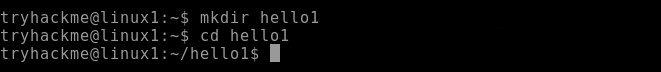
# Syntax :- mkdir <directory name>



**4. Cd command :-**

The [cd command](https://www.geeksforgeeks.org/cd-command-in-linux-with-examples/) is used to navigate between directories.

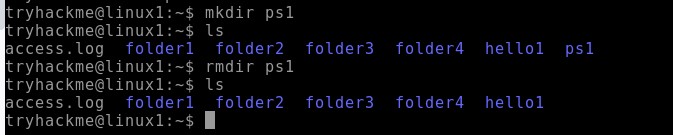
# Syntax :- cd <directory name>



**5. Rmdir command :-**

The [rmdir command](https://www.geeksforgeeks.org/rmdir-command-in-linux-with-examples/) is used to delete permanently an empty directory.

# Syntax :- rmdir <directory name>



**6. Cat command :-**

Display file contents on terminal

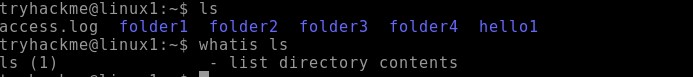
# Syntax :- cat <file name>



**7. Whatis command :-**

whatis command in Linux is used to get a one-line manual page description.

# Syntax :- whatis [option] [command\_name]



# 8. Find command :-

The find command in Linux is a dynamic utility designed for comprehensive file and directory searches within a hierarchical structure.

# Syntax :- find [path] [options] [expression]



**9. Echo command :-**

[echo command](https://www.geeksforgeeks.org/echo-command-in-linux-with-examples/) in Linux is specially used to print something in the terminal

# Syntax :- echo <Text>



# 10. Cal command :-

The [cal command](https://www.geeksforgeeks.org/cal-command-in-linux-with-examples/) is not the most famous command in the terminal but it functions to view the calendar for a particular month in the terminal. Let’s see how this works.

# Syntax :- cal <month> <Year>



# 11. Date command :-

Datecommand is used to display the system date and time. date command is also used to set date and time of the system.

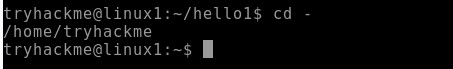
Syntax :- **date**



# 12. Cd - command :-

**Cd -** command is used to go back to previous directory in terminal of the system.

Syntax :- **cd –**



# 13. Cp command :-

The cp command copies files or directories from a source to a destination. It can handle single or multiple files and directories, and it can also overwrite existing files if specified.

# Syntax: cp [options] source destination

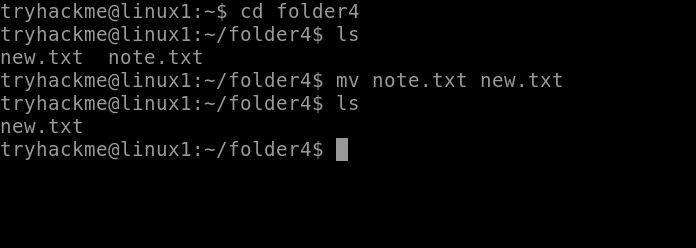
Example:



# 14. mv

The mv command in Linux is used to move or rename files and directories.

Syntax: **mv [options] source destination** Example:

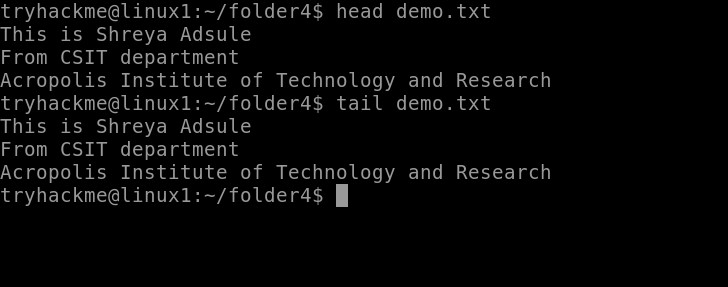


# 15. head

The head command outputs the first part of files or input data. It is commonly used to preview the beginning of a file or stream.

# Syntax: head [options] [file...]

Example:

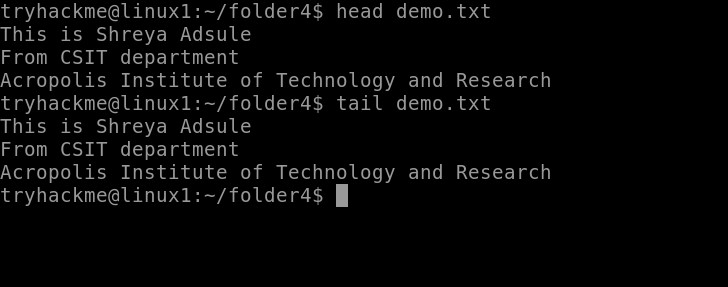


# 16. tail

The tail command outputs the last part of files or input data. It is often used to view the most recent entries in a log file or to monitor the end of a file for changes.

Syntax: **tail [options] [file...]**

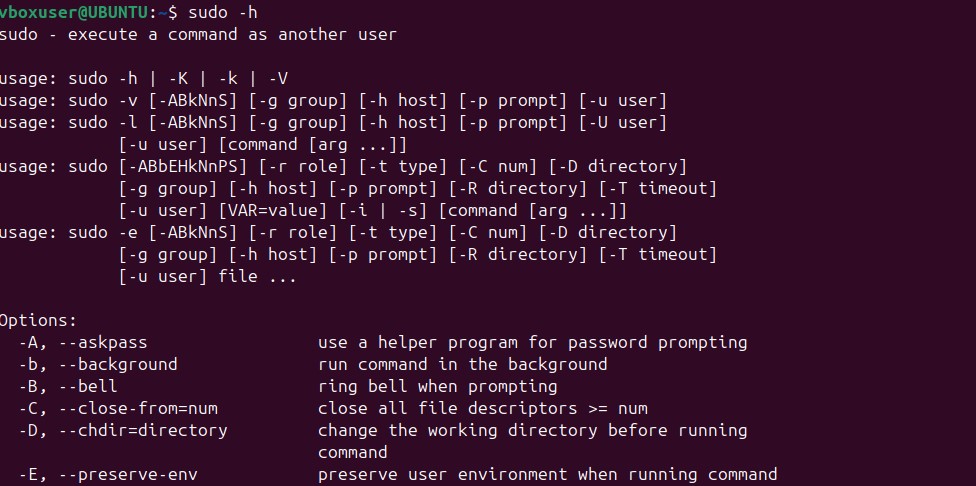
Example :



# 17. sudo

The sudo command grants elevated privileges to run commands that require root or administrative permissions. It's typically used to perform system administration tasks. Syntax: **sudo [options] command**

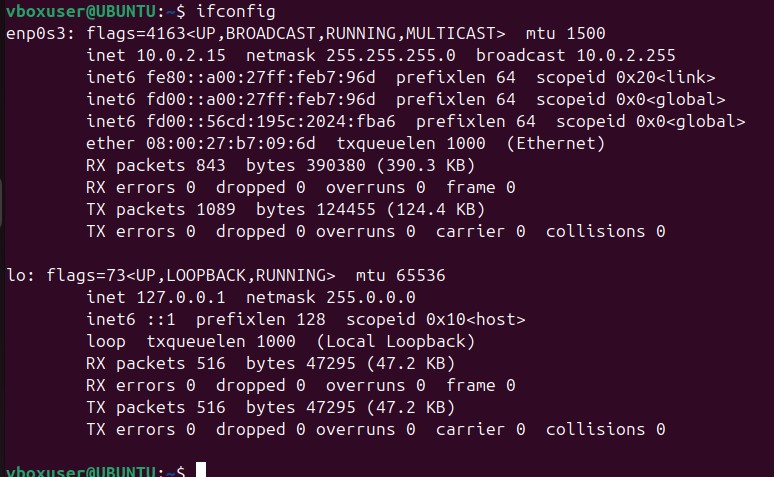
Example:



# 18. ifconfig

The ifconfig (interface configuration) command is used to display or configure a network interface.

Syntax: **ifconfig [interface] [options]** Example:

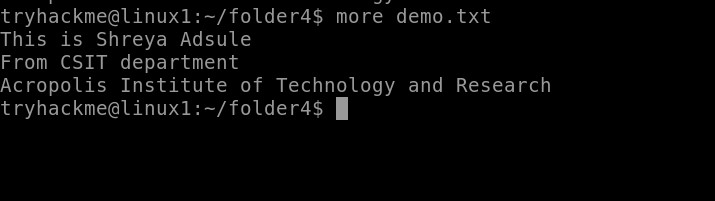


# 19. more

The more command displays the contents of a file, pausing after each screen of text. It is useful for viewing long files that don't fit on a single screen.

# Syntax: more [options] [file]

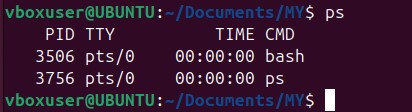
Example:



# 20. ps

The ps command provides a snapshot of current processes, showing details like process IDs (PIDs), terminal associated with the process, CPU and memory usage, and the command that started the process.

Syntax: p**s [options]** Example:

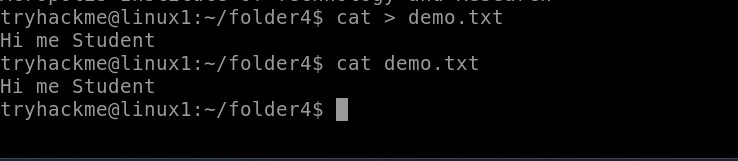


# 21. cat >

Using cat > filename, you can start typing text directly into a new file. This command redirects the terminal input into the specified file until you signal that you're done.

Syntax: **cat > filename**

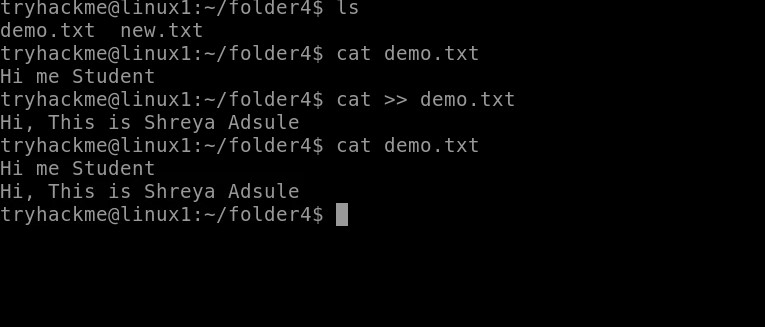
Example:



# 22. cat >>

Using cat >> filename, you can add new content to the end of a specified file. This command allows you to continue writing to the file without overwriting its current contents.

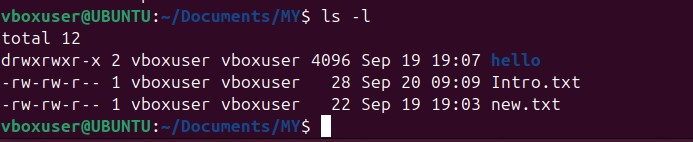
Syntax: **cat >> filename**



# 23. ls -l

The ls -l command lists files and directories in a long format, showing detailed attributes for each item, including permissions, number of links, owner, group, size, and modification date.

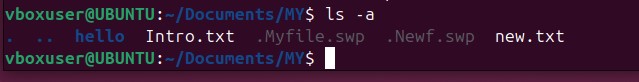
Syntax: **ls -l [directory]** Example:



# 24. ls -a

The ls -a command displays all entries in a directory, including those that begin with a dot (.), which are considered hidden files in Unix-like systems.

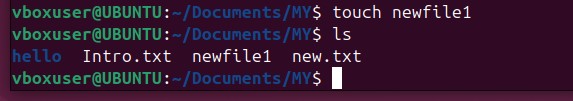
Syntax: **ls -a [directory]** Example:



# 25. touch

The touch command creates a new, empty file if the specified file does not exist. If the file already exists, it updates the access and modification timestamps to the current time without modifying the file's content.

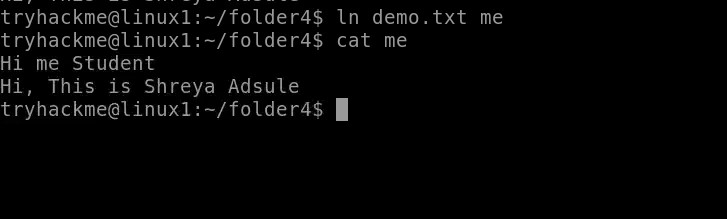
# Syntax: touch [options] filename



# 26. ln

The ln command in Linux is used to create links between files. There are two types of links: hard links and symbolic (soft) links.

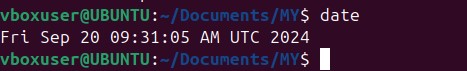
Syntax: **ln [options] target [link\_name]** Example:



# 27. date

The date command in Linux is used to display or set the system date and time.

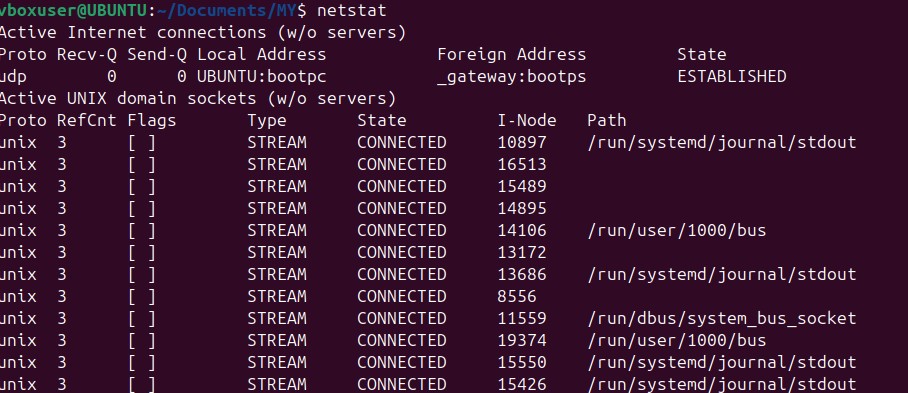
Syntax: **date [options] [+format]** Example:



# 28. netstat

The netstat command provides information about active network connections and network interface statistics, helping users monitor and troubleshoot network issues.

Syntax: **netstat [options]** Example:

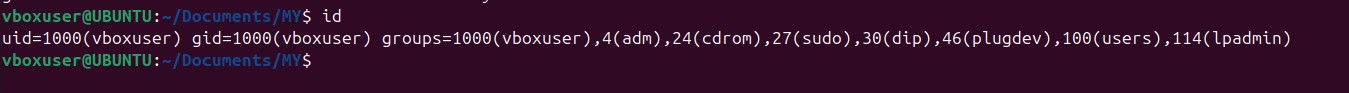


# 29. gid

In Linux, GID stands for Group Identifier. It is a numeric value used to identify a specific group on the system. Each user in Linux can belong to one or more groups, and each group is assigned a unique GID.

Syntax: **id username**

Example:



# 30. chmod

The chmod command allows users to specify who can read, write, or execute a file. Permissions can be set for three categories: the file owner, the group, and others.

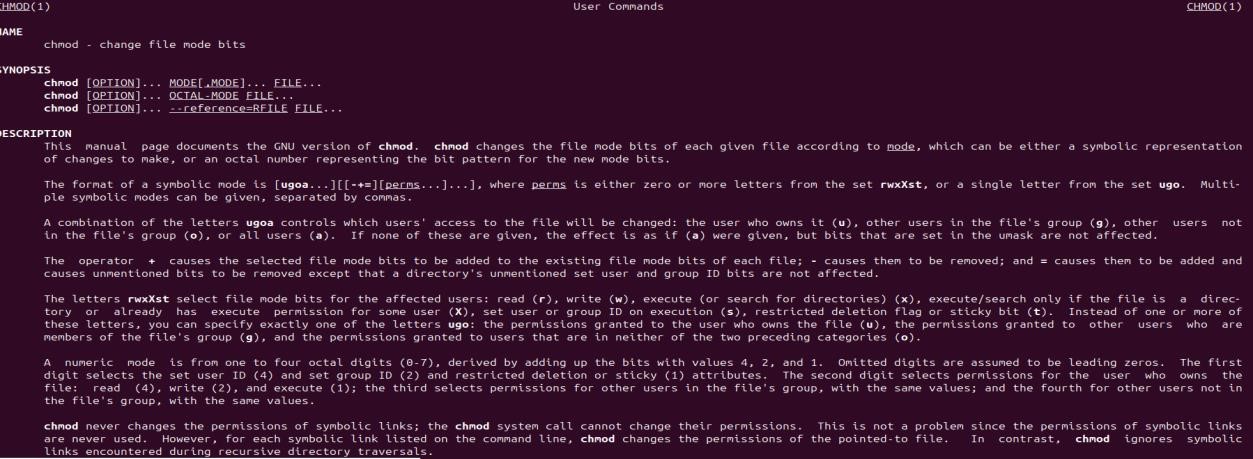
Syntax: **chmod [options] mode file** Example:



# 31. man

The man command is a built-in command that allows users to access the manual documentation for commands, functions, system calls, and other components in Linux.

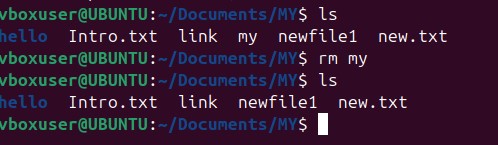
Syntax: **man [options] command** Example :



# 32. rm

The rm command allows users to delete files and directories from the filesystem. It is a powerful command that permanently removes files without placing them in a recycle bin or trash.

Syntax: **rm [options] file** Example:



# 33. rmdir

The rmdir command allows users to delete directories, but it can only remove those that are empty. If the directory contains files or other directories, the command will fail.

Syntax: **rmdir [options] directory** Example:



# 34. less

The less command provides a convenient way to scroll through text files, allowing both forward and backward navigation.

# Syntax: less [options] file

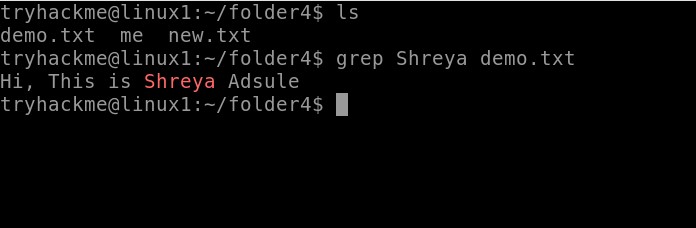
Example:



# 35. grep

The grep command searches through the input (files or standard input) and prints lines that match a specified pattern. It's commonly used for text processing and searching logs.

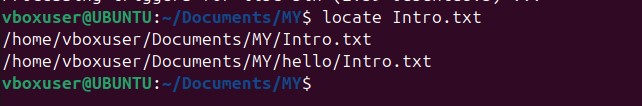
Syntax: **grep [options] pattern [file...]** Example:



# 36. locate

The locate command searches for files and directories in a database that contains the paths of all files on the system. This database is typically updated daily by a background service (updatedb), allowing for fast searches.

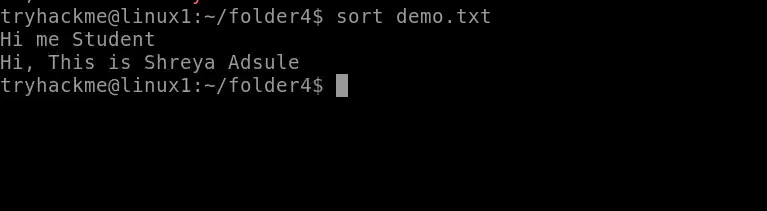
Syntax: **locate [options] pattern** Example:



# 37. sort

The sort command arranges the lines of a file or input in a specified order (ascending or descending). By default, it sorts in ascending order based on the ASCII values of characters.

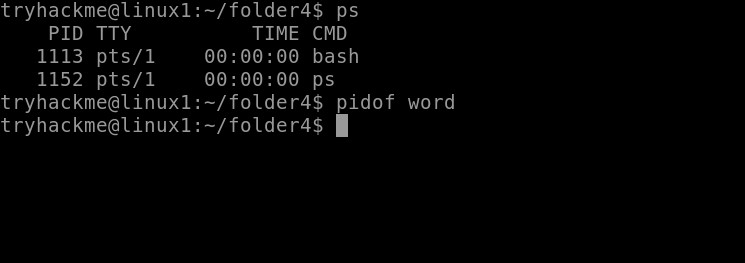
Syntax**: sort [options] [file...]** Example:



# 38. pid

The PID (process identification number) is a serial number (starting from 1) given by the operating system to uniquely identify the process. Every process started either by the operating system or by the user gets a PID in order of their invocation by the kernel.

Syntax: **pidof <exact\_process\_name>** Example:

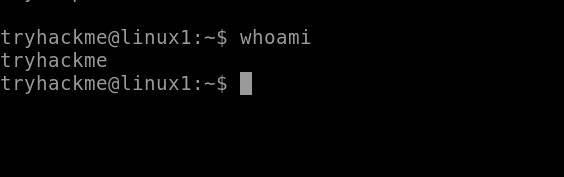


# 39. whoami

The command allows Linux users to see the currently logged-in user. The output displays the username of the effective user in the current shell. Additionally, whoami is useful in bash scripting to show who runs the script

# Syntax : whoami [OPTION]

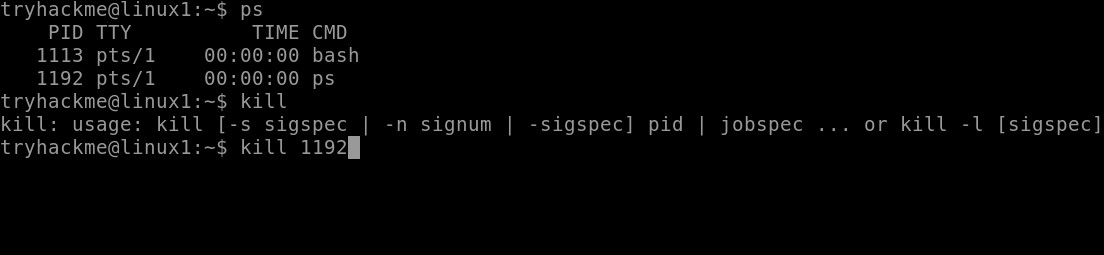
Example :



**40. kill**: kill command in Linux (located in /bin/kill), is a built-in command which is used to terminate processes manually

# Syntax : kill [signal] PID

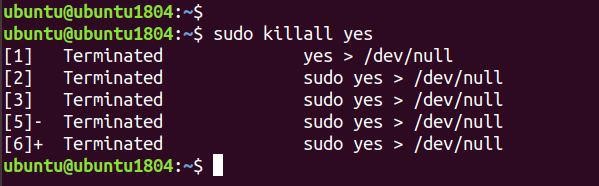
Example :



# 41. kill all

The killall command in Linux is a utility that terminates running processes based on their name. It can be useful when you need to kill multiple instances of a process or when you don't know the process ID (PID).

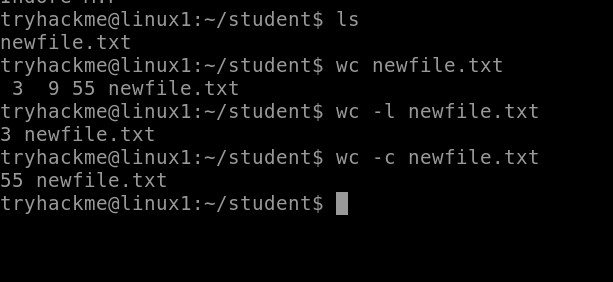
Syntax : **killall** [ **-** ] [ **-signal** ] Example :



**42. wc:** wc stands for **word count**. As the name implies, it is mainly used for counting purpose.It is used to find out **number of lines**, **word count**, **byte and characters count** in the files specified in the file arguments.

Syntax : **wc [option]... [file]...**

Example :

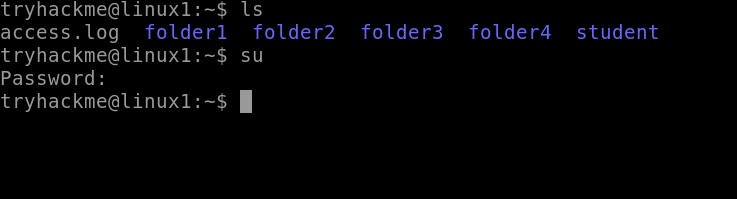


# 43. su :

The su command in Linux switches users or executes commands as a different user. It's useful for administrative tasks that require elevated privileges.

# Syntax : su [options] [username]

Example :

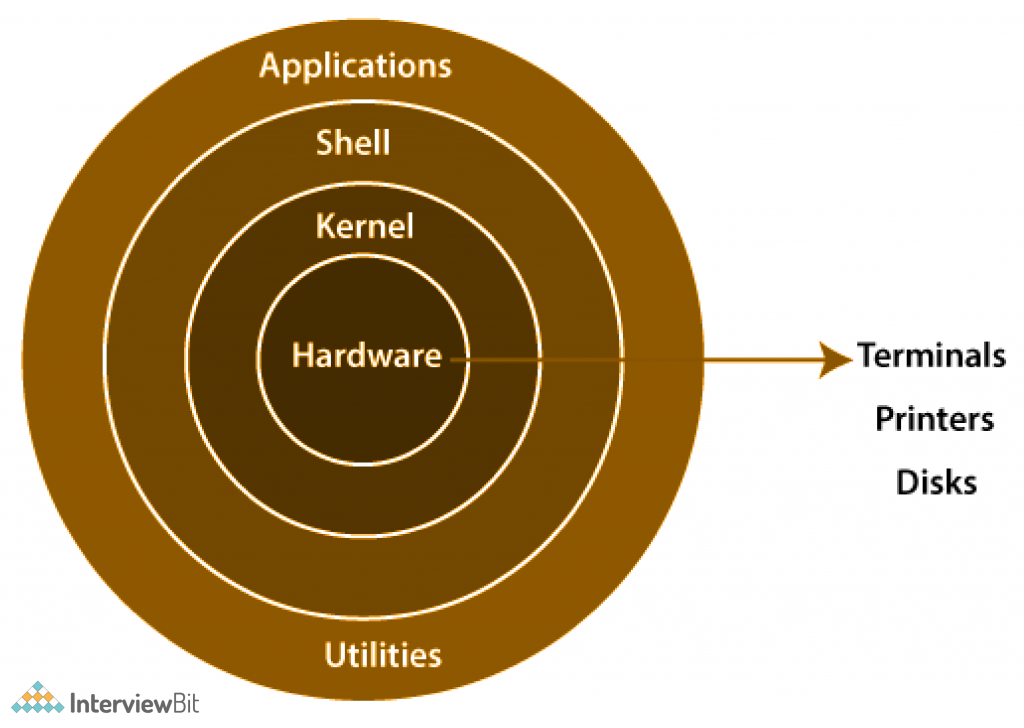


Lab 3:Linux basics and Architecture

A computer’s operating system interface to the hardware is referred to as a software application. A number of software applications are run on operating systems to manage hardware resources on a computer.

The diagram illustrates the structure of the Linux system, according to the layers concept.

Diagram of architecture:



* **Hardware layer** − Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).
* **Kernel** − It is the core component of Operating System, interacts directly with hardware, provides low level services to upper layer components.
* **Shell** − An interface to kernel, hiding complexity of kernel's functions from users. The shell takes commands from the user and executes kernel's functions.
* **Utilities** − Utility programs that provide the user most of the functionalities of an operating systems

**Kernal:**

* The Linux kernel is a monolithic kernel, which means that device drivers are components of the kernel. The kernel is the core of an operating system and is responsible for managing memory, resources, and devices.

**The five main types of kernels are:**

* Monolithic Kernel.
* Microkernel.
* Hybrid Kernel.
* Nano Kernel.
* Exo Kernel.

## **1. Monolithic Kernel**

It is one of the types of kernel where all operating system services operate in kernel space. It has dependencies between systems components. It has huge lines of code which is complex.

**Example:**

Unix, Linux, Open VMS, XTS-400 etc.

**2. Micro Kernel**

It is kernel types which has minimalist approach. It has virtual memory and thread scheduling. It is more stable with less services in kernel space. It puts rest in user space. It is use in small os.  
**Example :**

Mach, L4, AmigaOS, Minix, K42 etc.

**3. Hybrid Kernel**

It is the combination of both monolithic kernel and microkernel. It has speed and design of monolithic kernel and modularity and stability of microkernel.   
**Example :**

Windows NT, Netware, BeOS etc.

**4. Exo Kernel**

It is the type of kernel which follows end-to-end principle. It has fewest hardware abstractions as possible. It allocates physical resources to applications.

**Example :**

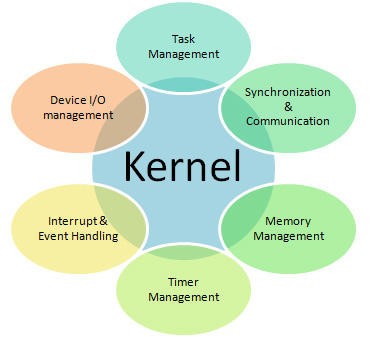
Nemesis, ExOS etc.

**5. Nano Kernel**

It is the type of kernel that offers hardware abstraction but without system services. Micro Kernel also does not have system services therefore the Micro Kernel and Nano Kernel have become analogous.

**Example :**

EROS etc.



**Shell:**

A shell is a program that acts as an interface between a user and the kernel. It allows a user to give commands to the kernel and receive responses from it. Through a shell, we can execute programs and utilities on the kernel. Hence, at its core, a shell is a program used to execute other programs on our system.

**Types of shell:**

* **The C Shell –**

Denoted as **csh**

* Bill Joy created it at the University of California at Berkeley. It incorporated features such as aliases and command history. It includes helpful programming features like built-in arithmetic and C-like expression syntax. In C shell:

Command full-path name is /bin/csh,  
Non-root user default prompt is hostname %,  
Root user default prompt is hostname #.

* **The Bourne Shell –**

Denoted as **sh**

* It was written by Steve Bourne at AT&T Bell Labs. It is the original UNIX shell. It is faster and more preferred. It lacks features for interactive use like the ability to recall previous commands. It also lacks built-in arithmetic and logical expression handling. It is default shell for Solaris OS. For the Bourne shell the:

Command full-path name is /bin/sh and /sbin/sh,  
Non-root user default prompt is $,  
Root user default prompt is #.

* **The Korn Shell**

It is denoted as **ksh**

* It was written by David Korn at AT&T Bell Labs. It is a superset of the Bourne shell. So it supports everything in the Bourne shell.It has interactive features. It includes features like built-in arithmetic and C-like arrays, functions, and string-manipulation facilities. It is faster than C shell. It is compatible with script written for C shell. For the Korn shell the:

Command full-path name is /bin/ksh,  
Non-root user default prompt is $,  
Root user default prompt is #.

* **GNU Bourne-Again Shell –**

Denoted as **bash**

* It is compatible to the Bourne shell. It includes features from Korn and Bourne shell. For the GNU Bourne-Again shell the:

Command full-path name is /bin/bash,  
Default prompt for a non-root user is bash-g.gg$   
(g.ggindicates the shell version number like bash-3.50$),  
Root user default prompt is bash-g.gg#.

* It is compatible to the Bourne shell. It includes features from Korn and Bourne shell. For the GNU Bourne-Again shell the:

Command full-path name is /bin/bash,  
Default prompt for a non-root user is bash-g.gg$   
(g.ggindicates the shell version number like bash-3.50$),  
Root user default prompt is bash-g.gg#.

* **T Shell –**

Denoted as **tsh**

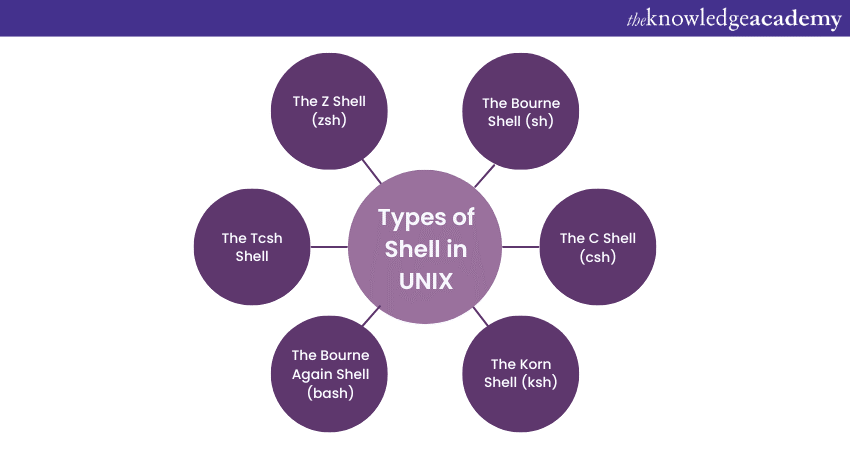
* It was originally developed for the Plan 9 operating system, but has since been ported to other systems, including Linux, FreeBSD, and macOS.

Command full-path name is /bin/tcsh,  
Default prompt for a non-root user is abhishekaslk(user):~>  
Root user default prompt is root@abhishekaslk(user):~#.

* **Z Shell –**

Denoted by **zsh**

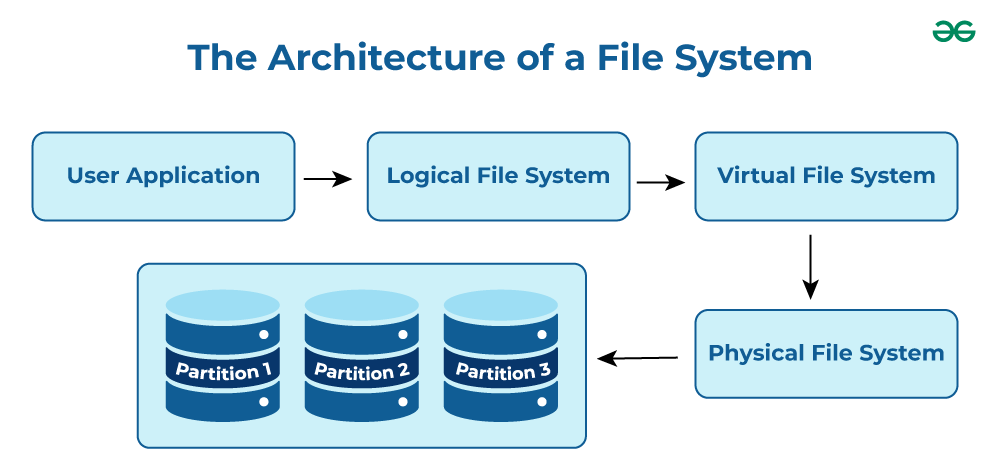
* Z Shell (zsh) was created by Paul Falstad in 1990 while he was a student at Princeton University. Z Shell is an extended version of the Bourne-Again Shell (bash), with additional features and capabilities.



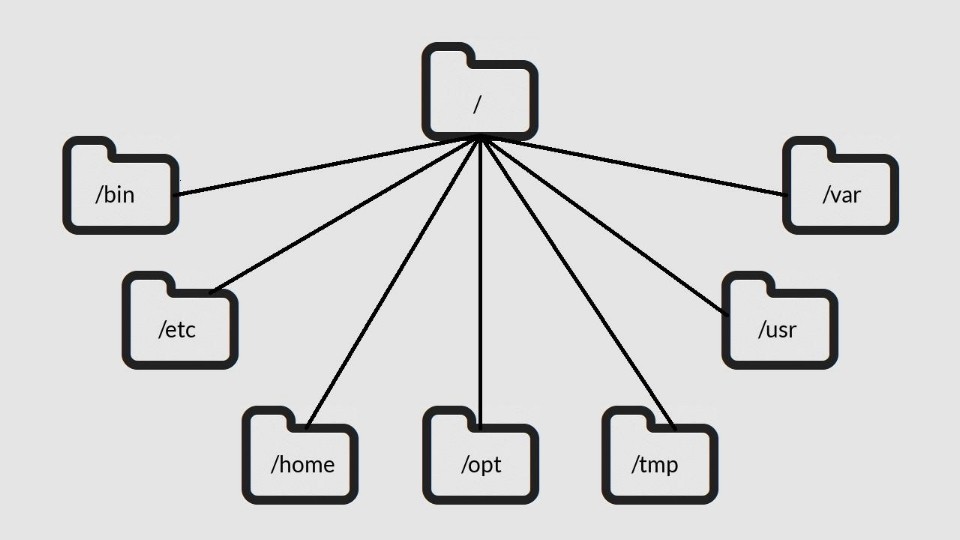
Lab 4: linux file system

**Managing the file system:**

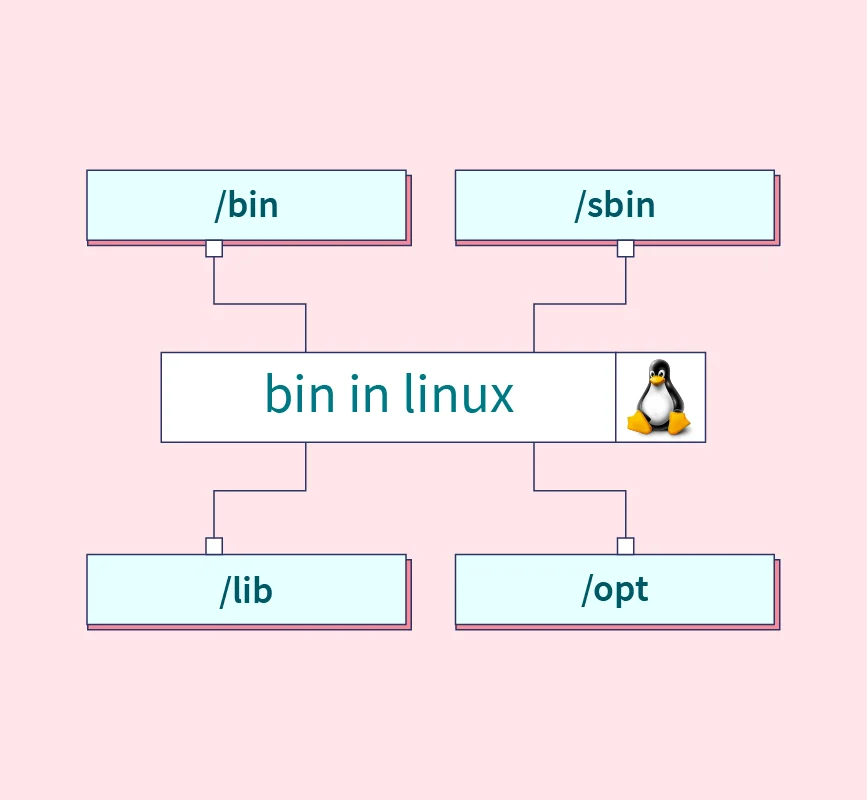
Linux file management involves organizing, accessing, and manipulating files and directories within the Linux operating system. Users interact with files using commands like ls (list), cp (copy), mv (move), and rm (remove). Directories, or folders, provide a hierarchical structure for organizing files.

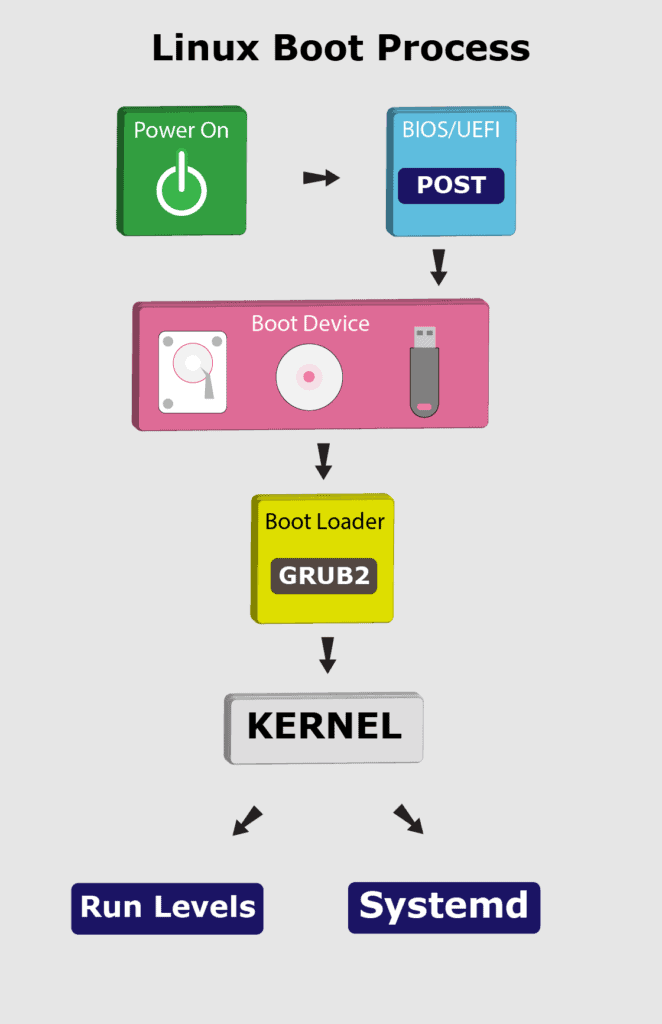


**Types of directories**:

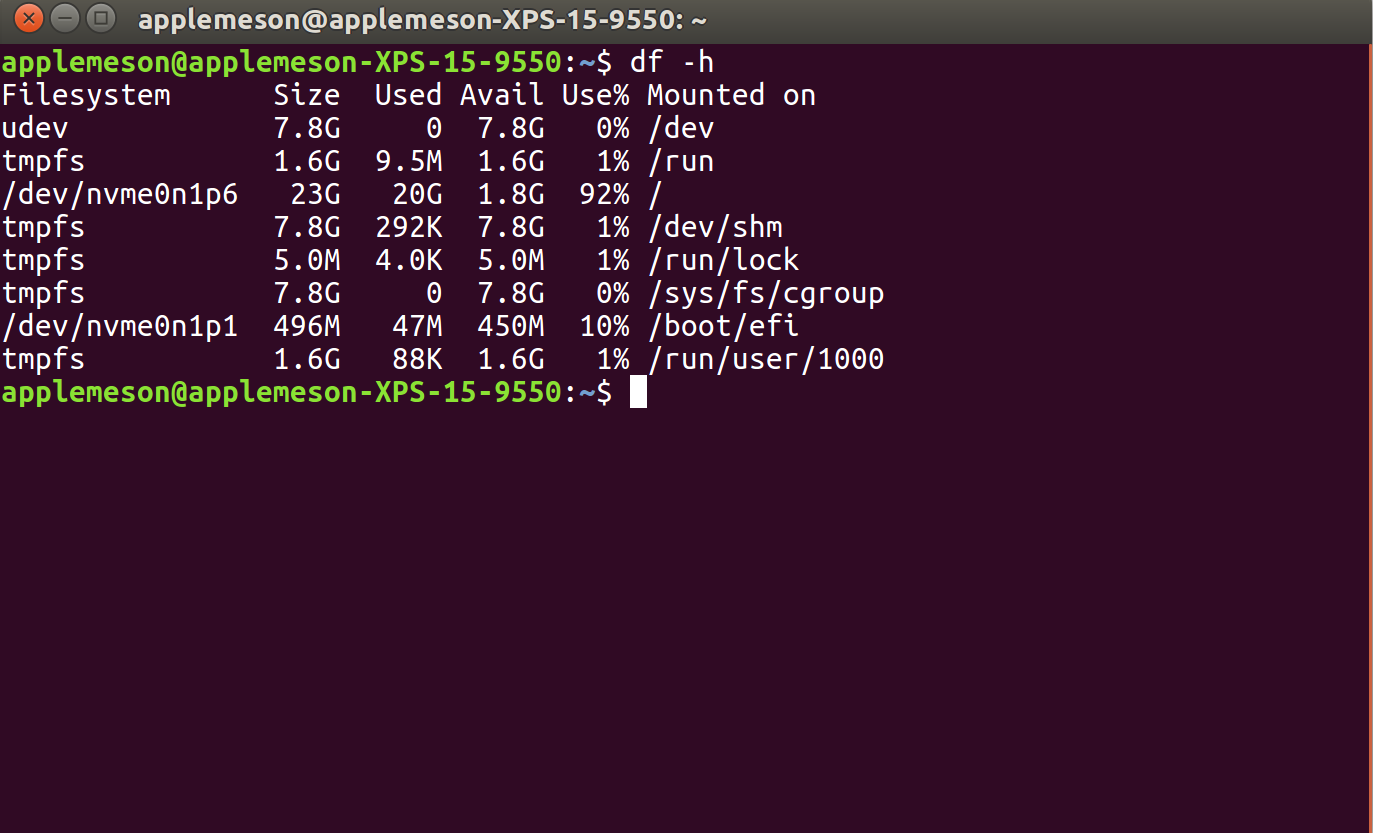


**1./bin**: The /bin directory, which stands for "binary" includes basic binaries that are required for the system's basic function. These binaries are available to all users and are required for the system to function properly. In other words, /bin contains the basic commands that users utilize on a daily basis.

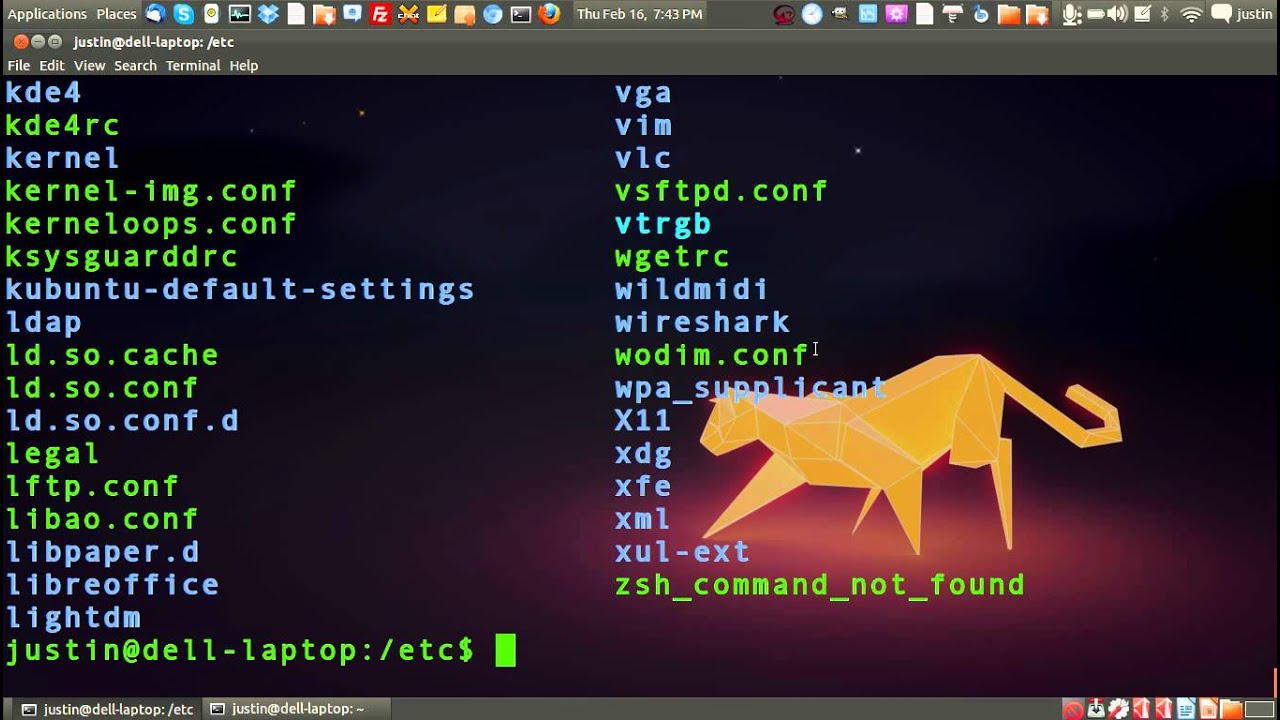


2. This directory contains everything required for the boot process except for configuration files not needed at boot time (the most notable of those being those that belong to the GRUB boot-loader) and the map installer. Thus, the /boot directory stores data that is used before the kernel begins executing user-mode programs. This may include redundant (back-up) master boot records, sector/system map files, the kernel and other important boot files and data that is not directly edited by hand. Programs necessary to arrange for the boot loader to be able to boot a file are placed in /sbin. Configuration files for boot loaders are placed in /etc. The system kernel is located in either / or /boot (or as under Debian in /boot but is actually a symbolically link accordance with the FSSTND).

**3./dev:** The /dev directory contains device files (also sometimes known as device special files and device nodes) that provide access to peripheral devices such as hard disks, to resources on peripheral devices such as disk partitions, and pseudo devices such as a random number generator.



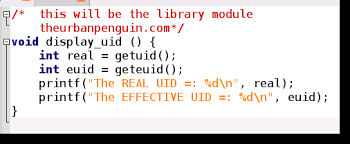
**4./etc:** In the Linux operating system, the '/etc' directory is a crucial part of the file system, standing for "et cetera" and containing configuration files and directories. The '/etc' directory holds various configuration files that are essential for the proper functioning of the system and its installed applications.



**5. /home**: We can use the /home as the default path for Linux Home Directory. It is the primary or starting path for entering the individual user profile. USER NAME: We need to pass the user name in the “/home” path. It will help to enter in the individual user profile path (in terms of the file system).

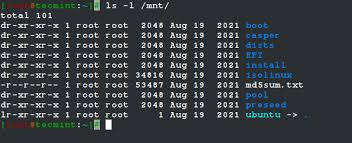


**6.lib:** The "lib" and "lib64" directories are the default locations where shared libraries are stored in Linux systems. The "lib" directory is present on 32-bit systems, while the "lib64" directory is used on 64-bit systems.

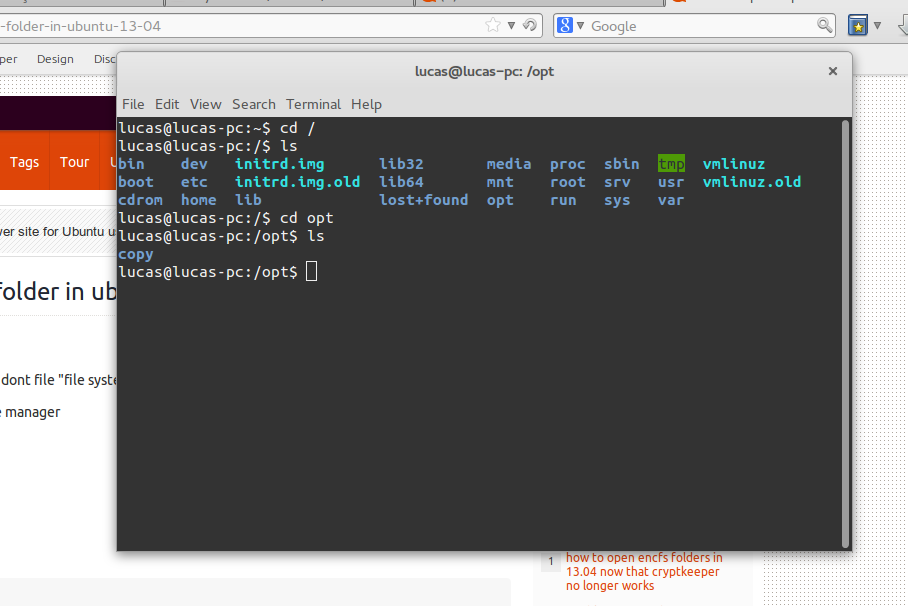


**7./media:** The "/media" directory in the Linux filesystem layout serves a important purpose in managing and accessing removable media devices. It provides a standardized location for mounting and accessing various types of removable media, such as USB drives, optical discs, and external hard drives.

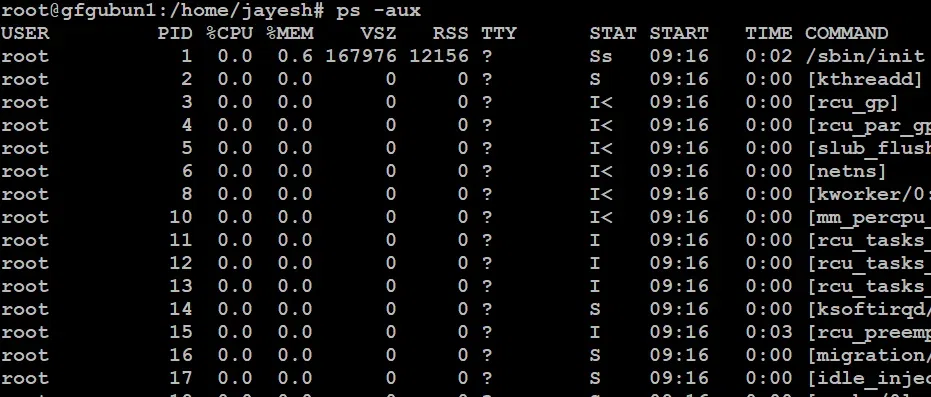
**8./mnt:** Linux has two default mount point locations for removable media: /dev (device) is where all physical devices are first mounted. From there, an additional link to the removable media is made in the /mnt (mount) directory.



**9./opt:** The term "/opt" stands for "optional" and signifies that the directory is intended for the installation of optional or add-on software packages.



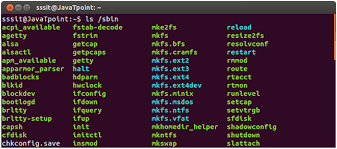
**10./proc:** The /proc directory is present on all Linux systems, regardless of flavor or architecture. The /proc directory is NOT a real file system but a virtual file system that is created dynamically by Linux to provide access to certain types of hardware information and information about the running processes.



**11./root:** On Linux, everything starts at "root." The hard drive has a root partition ( / ), and the default initial user (at least traditionally) is root (often referred to as the "superuser"). Historically, you used the root account to log in, create secondary "normal" users, and then mostly forget about it.



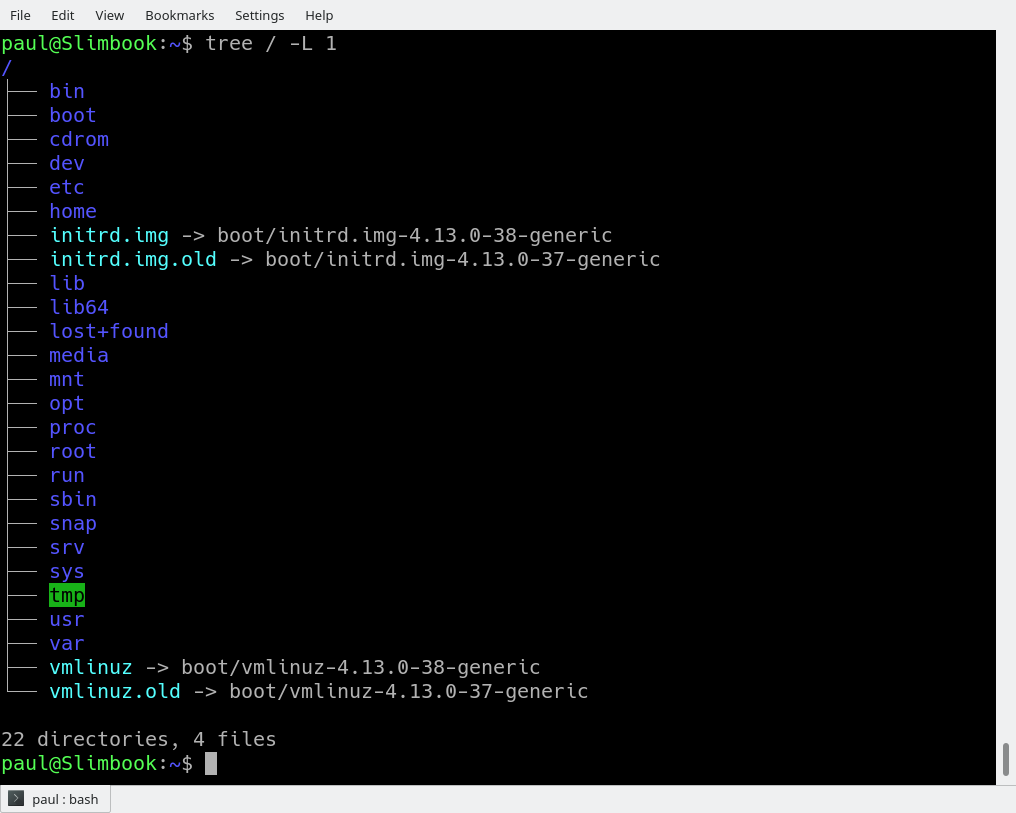
**12./sbin:** The term sbin in Linux stands for "system binaries" or "system command binaries." It is one of the directories in the Linux file system hierarchy, and it includes important system administration commands used primarily by the system administrator (root) to control the operating system and conduct other system-related .



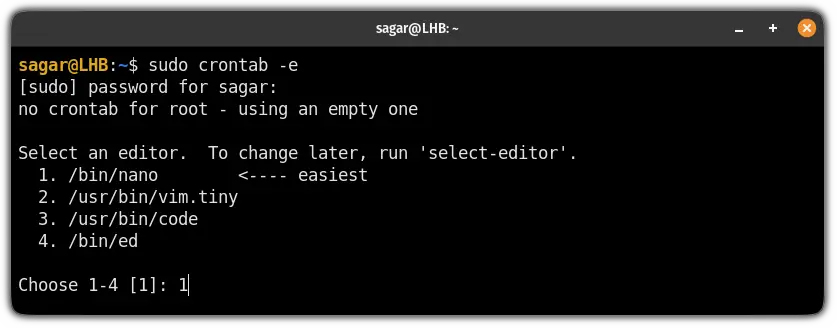
**13./srv:** /srv. The /srv directory contains data for servers. If you are running a web server from your Linux box, your HTML files for your sites would go into /srv/http (or /srv/www). If you were running an FTP server, your files would go into /srv/ftp.



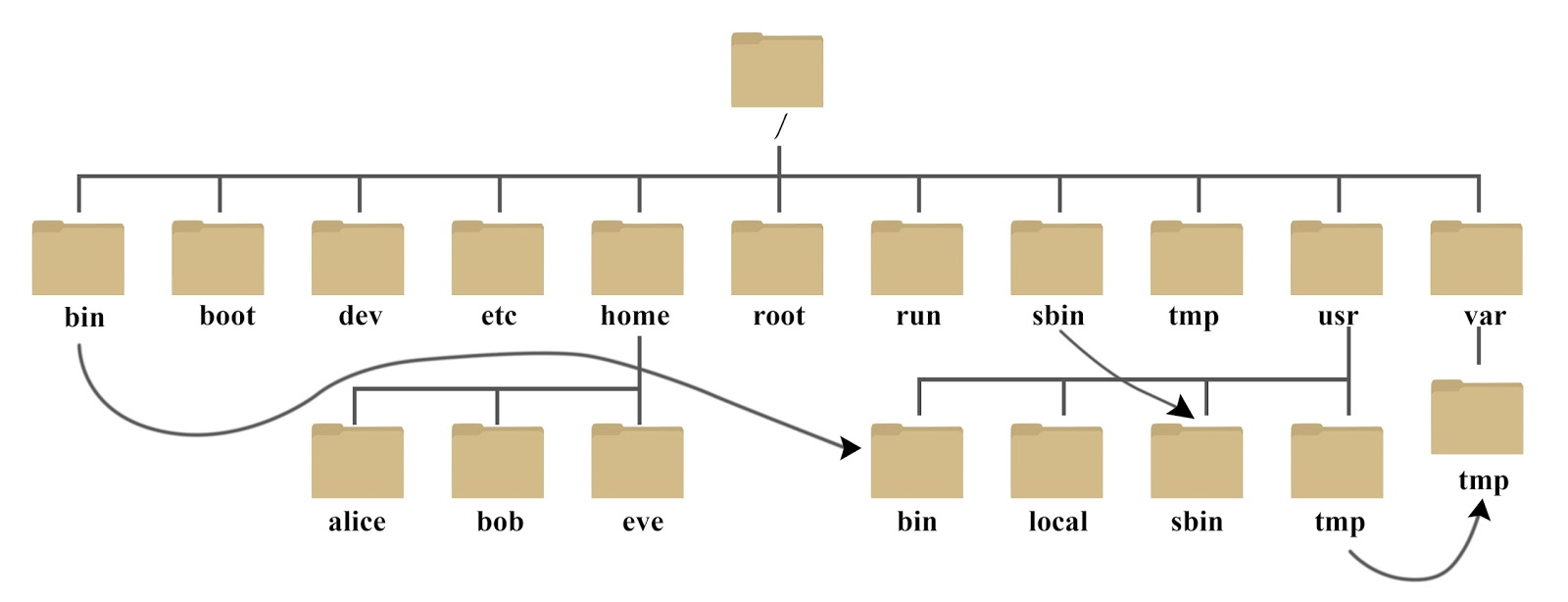
**14./sys:** /sys is another virtual directory like /proc and /dev and also contains information from devices connected to your computer. In some cases you can also manipulate those devices.



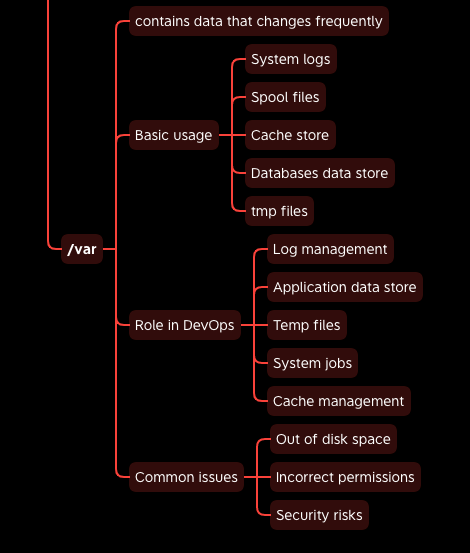
**15./tmp:** What is the /tmp directory in Linux? As the name suggests, the tmp (temporary) directory under root is used to store the data used by the system and user applications to store the data that are needed for a short period of time. Most Linux distributions are preconfigured to empty the tmp directory after each reboot.



**16./usr:** This is one of the most critical directories in the Linux system. The /usr directory is a directory that comprises libraries, binaries, and documentation for installed software applications. System files contained in this directory are shareable among other users.



**17./var:** In Linux, /var is a standard directory that stands for "variable files". As the name suggests, this directory contains data that changes frequently while the system is running. The origins of the /var directory in Unix and Linux systems can be traced back to the Unix Filesystem Hierarchy Standard (FHS).



**#Types of files in linux**

Operating systems, the software that powers your computer, rely on a crucial element known as the file system. Think of it as a virtual organizational tool that manages, stores, and retrieves your data efficiently. In the Linux world, a diverse range of file systems has emerged, each crafted to address specific needs and preferences. This article aims to simplify the intricacies of Linux file systems, guiding beginners through their layers, characteristics, and implementations. By shedding light on these nuances, we empower users to make informed choices in navigating the dynamic landscape of Linux operating systems.



**1) ext (Extended File System):**

Implemented in 1992, it is the first file system specifically designed for Linux. It is the first member of the ext family of file systems.

**2) ext2:**

The second ext was developed in 1993. It is a non-journaling file system that is preferred to be used with flash drives and SSDs. It solved the problems of separate timestamp for access, inode modification and data modification. Due to not being journaled, it is slow to load at boot time.

**3) Xiafs:**

Also developed in 1993, this file system was less powerful and functional than ext2 and is no longer in use anywhere.

**4) ext3:**

The third ext developed in 1999 is a journaling file system. It is reliable and unlike ext2, it prevents long delays at system boot if the file system is in an inconsistent state after an unclean shutdown. Other factors that make it better and different than ext2 are online file system growth and HTree indexing for large directories.

**5) JFS (Journaled File System):**

First created by IBM in 1990, the original JFS was taken to open source to be implemented for Linux in 1999. JFS performs well under different kinds of load but is not commonly used anymore due to the release of ext4 in 2006 which gives better performance.

**6) ReiserFS:**

It is a journal file system developed in 2001. Despite its earlier issues, it has [tail packing](https://en.wikipedia.org/wiki/Block_suballocation#Tail_packing) as a scheme to reduce internal fragmentation. It uses a B+ Tree that gives less than linear time in directory lookups and updates. It was the default file system in SUSE Linux till version 6.4, until switching to ext3 in 2006 for version 10.2.

**7) XFS:**

XFS is a 64-bit journaling file system and was ported to Linux in 2001. It now acts as the default file system for many Linux distributions. It provides features like snapshots, online defragmentation, sparse files, variable block sizes, and excellent capacity. It also excels at parallel I/O operations.

**8) SquashFS:**

Developed in 2002, this file system is read-only and is used only with embedded systems where low overhead is needed.

**9) Reiser4:**

It is an incremental model to ReiserFS. It was developed in 2004. However, it is not widely adapted or supported on many Linux distributions.

**10) ext4:**

The fourth ext developed in 2006, is a journaling file system. It has backward compatibility with ext3 and ext2 and it provides several other features, some of which are persistent pre-allocation, unlimited number of subdirectories, metadata checksumming and large file size. ext4 is the default file system for many Linux distributions and also has compatibility with Windows and Macintosh.

**11) btrfs (Better/Butter/B-tree FS):**

It was developed in 2007. It provides many features such as snapshotting, drive pooling, data scrubbing, self-healing and online defragmentation. It is the default file system for Fedora Workstation.

**12) bcachefs:**

This is a copy-on-write file system that was first announced in 2015 with the goal of performing better than btrfs and ext4. Its features include full filesystem encryption, native compression, snapshots, and 64-bit check summing.

**13) Others:**

 Linux also has support for file systems of operating systems such as NTFS and exFAT, but these do not support standard Unix permission settings. They are mostly used for interoperability with other operating systems.