LAB CYCLE 3

EXPERIMENT NO: 5 **DATE:**

FILE SYSTEM HEIRARCHY

AIM

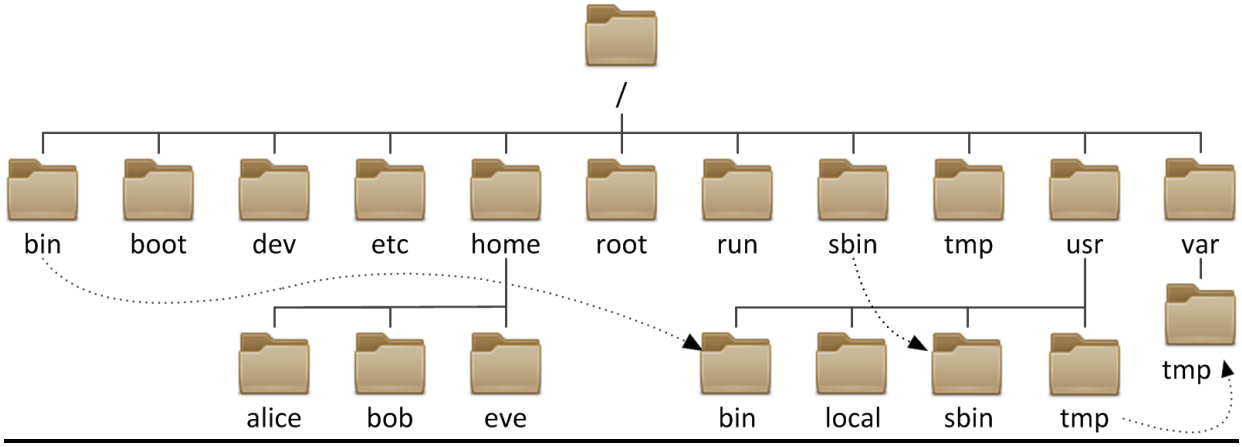
a) File system hierarchy in a common Linux distribution.

b) File and device permissions,

c) Study of system configuration files in /etc,

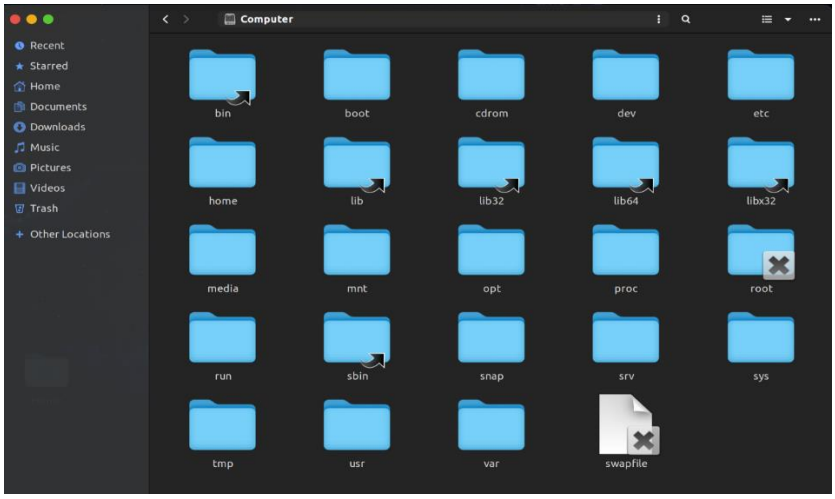
d) Familiarizing log files for system events, user activity, network events.

FILE HEIRARCHY



1. / (Root):
Primary hierarchy root and root directory of the entire file system hierarchy.

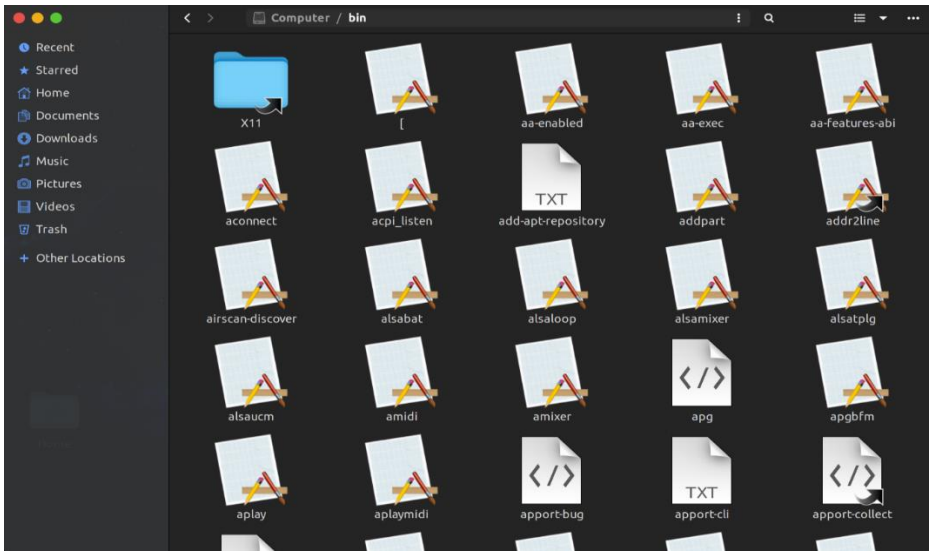
- Every single file and directory start from the root directory.
- The only root user has the right to write under this directory.
- /root is the root user's home directory, which is not the same as /



2. /bin :

Essential command binaries that need to be available in single-user mode; for all users, e.g., cat, ls, cp.

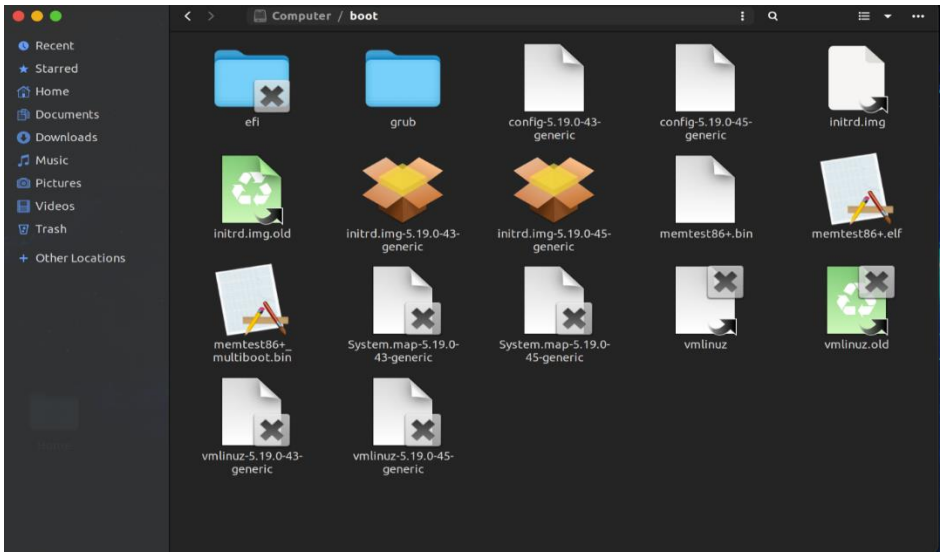
- Contains binary executables.
- Common linux commands you need to use in single-user modes are located under this directory.
- Commands used by all the users of the system are located here e.g. ps, ls, ping, grep, cp



3. /boot :

Boot loader files, e.g., kernels, initrd.

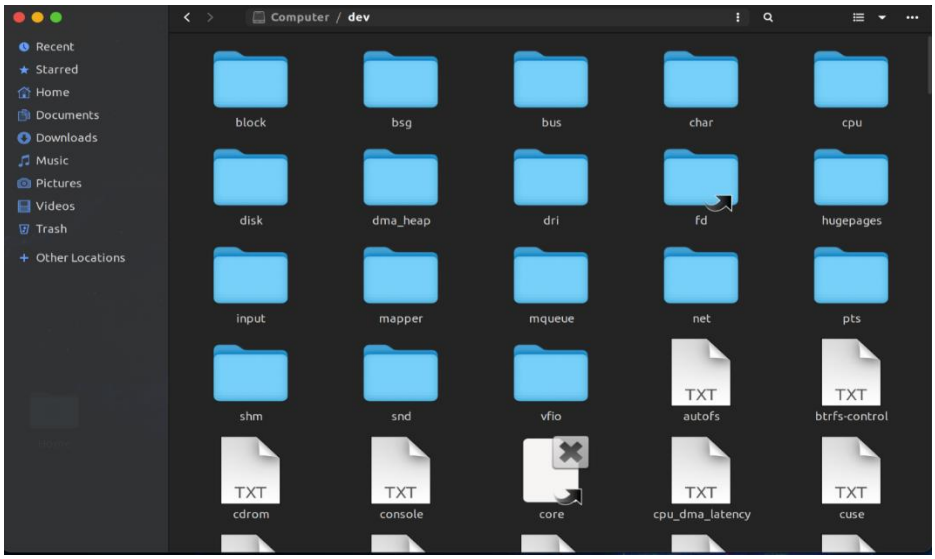
- Kernel initrd, vmlinux, grub files are located under /boot
- Example: initrd.img-5.19.0-43-generic, vmlinuz-5.19.0-43-generic



4. /dev :

Essential device files, e.g., /dev/null.

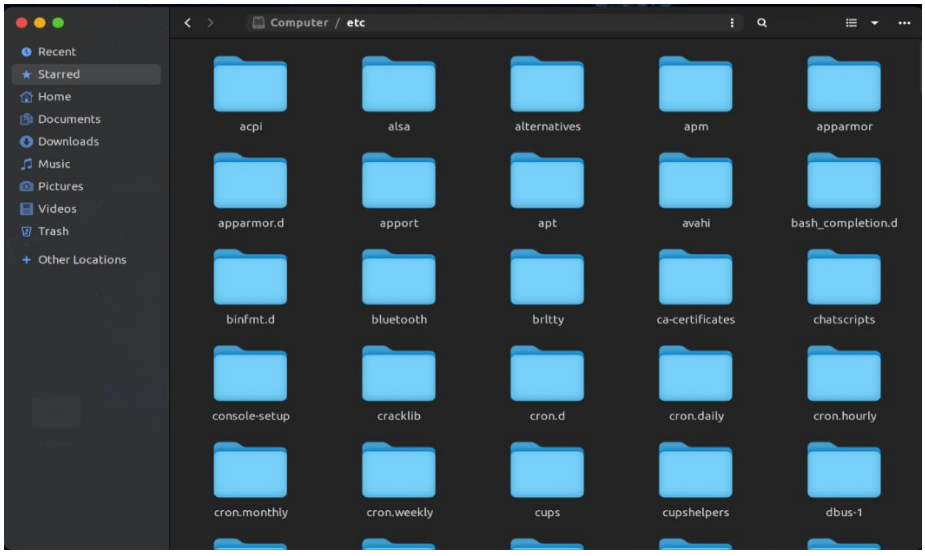
- These include terminal devices, usb, or any device attached to the system.
- Example: /dev/tty1, /dev/usbmon0



5. /etc :

Host-specific system-wide configuration files.

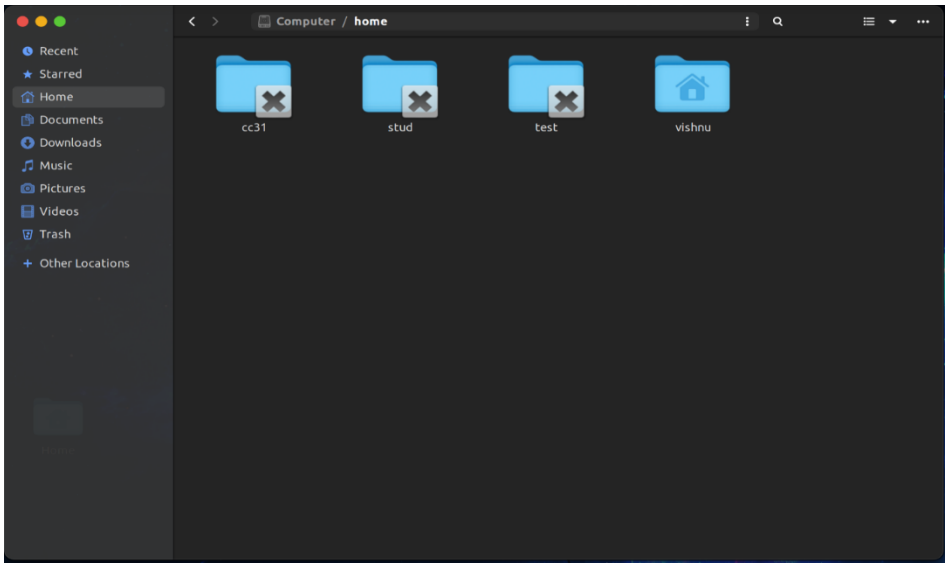
- Contains configuration files required by all programs.
- This also contains startup and shutdown shell scripts used to start/stop individual programs.
- Example: /etc/resolv.conf, /etc/logrotate.conf.



6. /home :

Users' home directories, containing saved files, personal settings, etc.

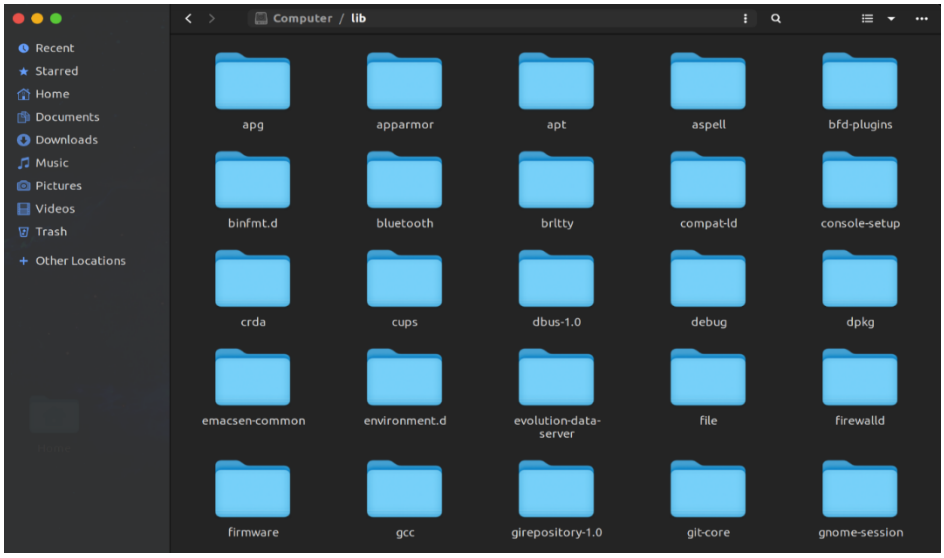
- Home directories for all users to store their personal files.
- example: /home/kishlay, /home/kv



7. /lib:

Libraries essential for the binaries in /bin/ and /sbin/.

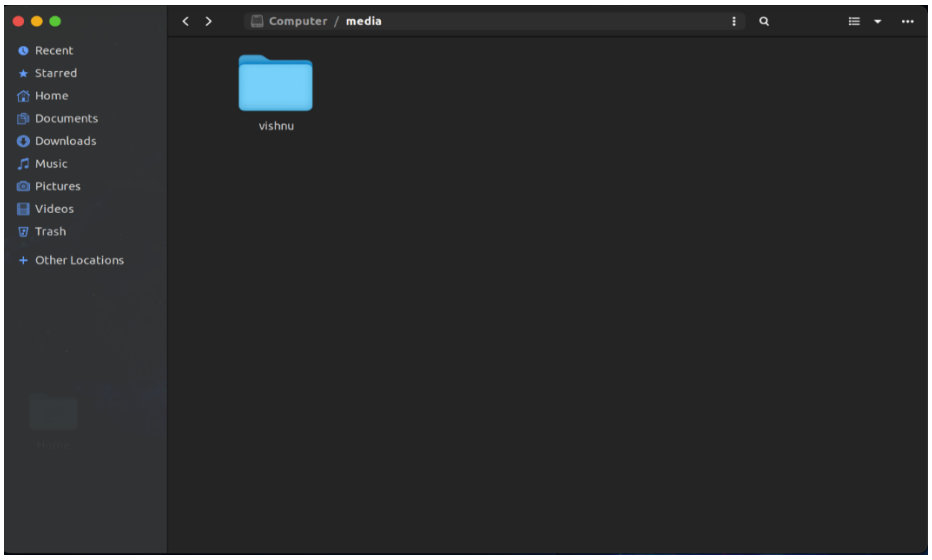
- Library filenames are either ld* or lib*.so.*
- Example: ld-2.11.1.so, libncurses.so.5.7



8. /media:

Mount points for removable media such as CD-ROMs (appeared in FHS-2.3).

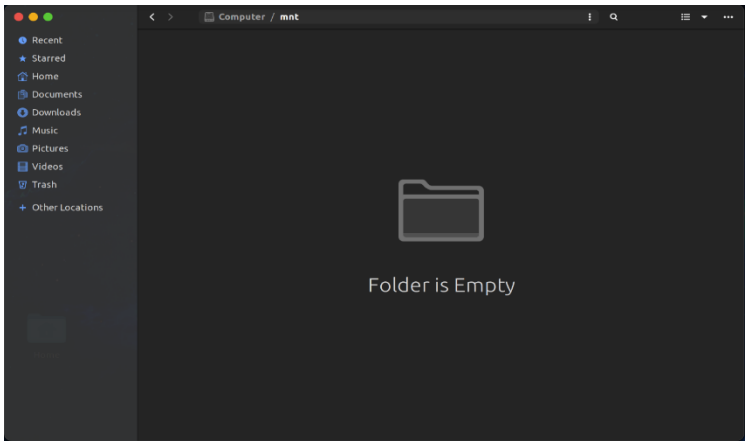
- Temporary mount directory for removable devices.
- Examples, /media/cdrom for CD-ROM; /media/floppy for floppy drives; /media/cdrecorder for CD writer



9. /mnt :

Temporarily mounted filesystems.

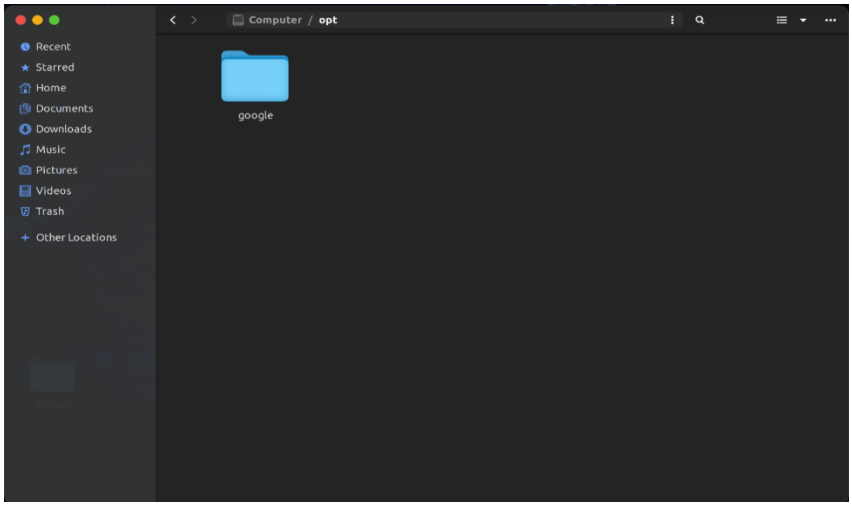
- Temporary mount directory where sysadmins can mount filesystems.



10. /opt :

Optional application software packages.

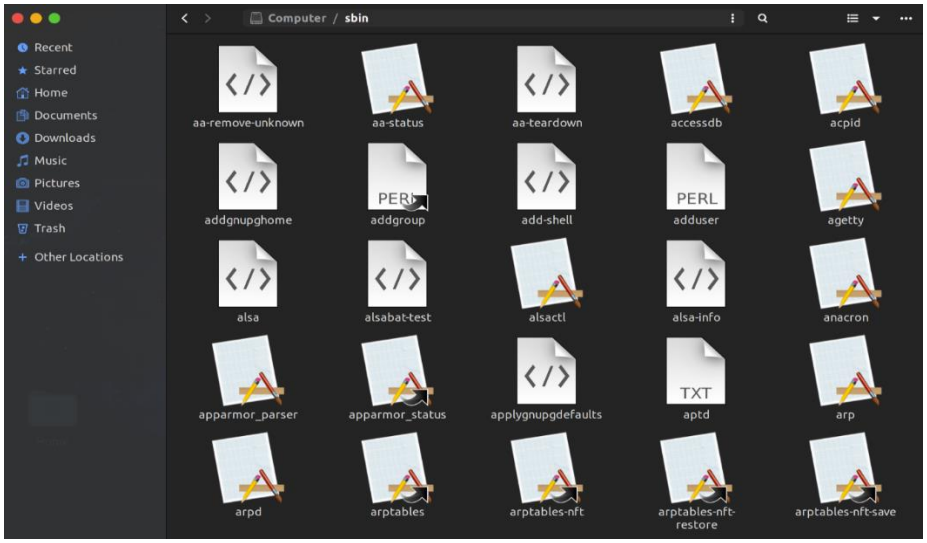
- Contains add-on applications from individual vendors.
- Add-on applications should be installed under either /opt/ or /opt/ sub-directory.



11. /sbin :

Essential system binaries, e.g., fsck, init, route.

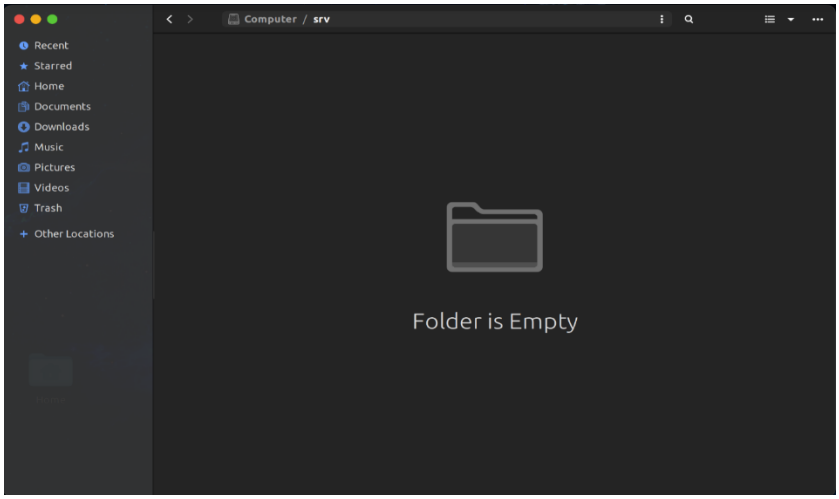
- Just like /bin, /sbin also contains binary executables.
- The linux commands located under this directory are used typically by system administrators, for system maintenance purposes.
- Example: iptables, reboot, fdisk, ifconfig, swapon



12. /srv :

Site-specific data served by this system, such as data and scripts for web servers, data offered by FTP servers, and repositories for version control systems.

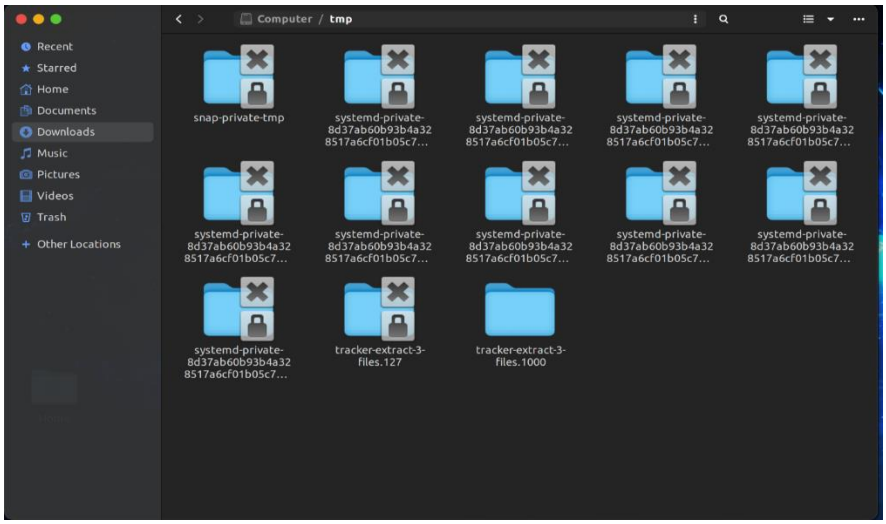
- srv stands for service.
- Contains server specific services related data.
- Example, /srv/cvs contains CVS related data.



13. /tmp :

Temporary files. Often not preserved between system reboots and may be severely size restricted.

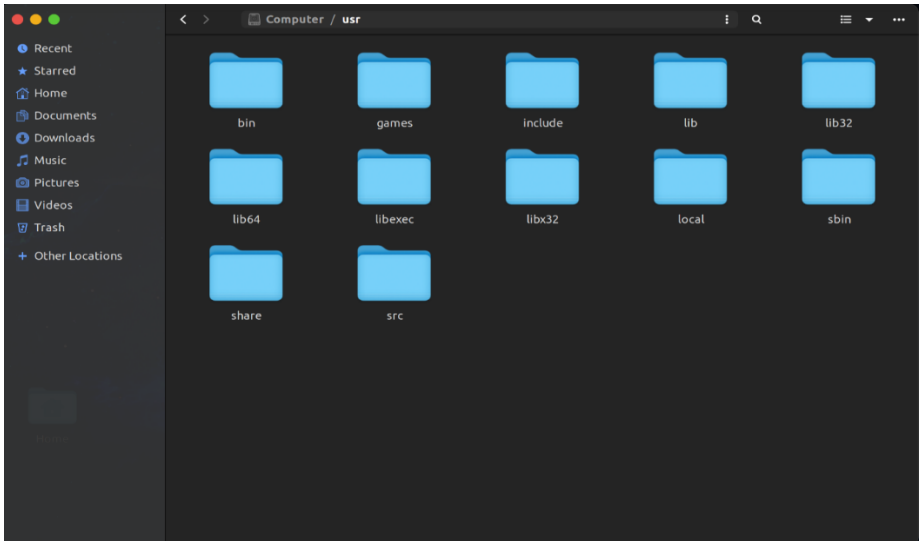
- Directory that contains temporary files created by system and users.
- Files under this directory are deleted when the system is rebooted.



14. /usr :

Secondary hierarchy for read-only user data; contains the majority of (multi-)user utilities and applications.

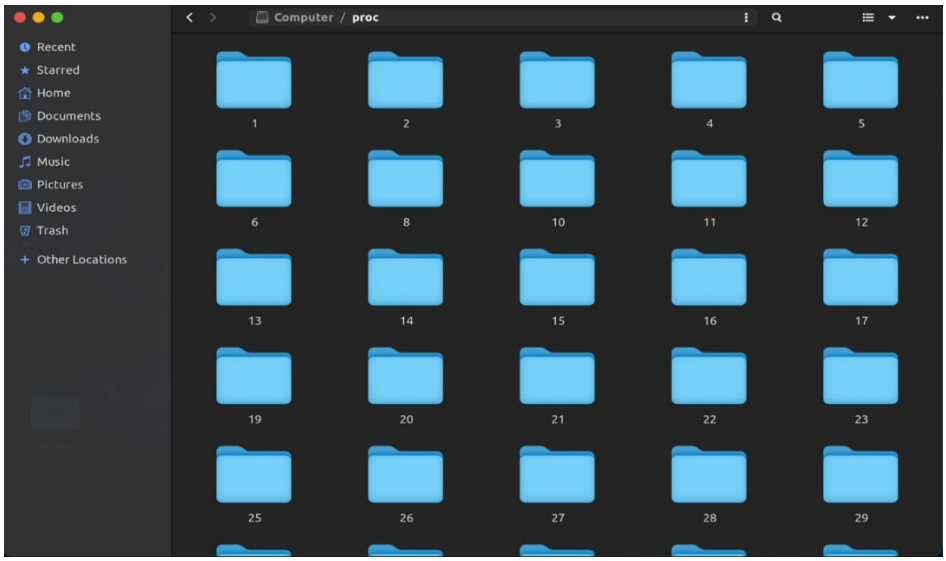
- Contains binaries, libraries, documentation, and source-code for second level programs.
- /usr/bin contains binary files for user programs. If you can't find a user binary under /bin, look under /usr/bin. For example: at, awk, cc, less, scp
- /usr/sbin contains binary files for system administrators. If you can't find a system binary under /sbin, look under /usr/sbin. For example: atd, cron, sshd, useradd, userdel



15. /proc:

Virtual filesystem providing process and kernel information as files. In Linux, it corresponds to a procs mount. Generally, automatically generated and populated by the system, on the fly.

- Contains information about system process.
- This is a pseudo filesystem that contains information about running processes. For example: /proc/{pid} directory contains information about the process with that particular pid.
- This is a virtual filesystem with text information about system resources. For example: /proc/uptime



Linux File permissions

The ls command along with its -l (for long listing) option will show you metadata about your Linux files, including the permissions set on the file.

```
$ ls -l

drwxr-xr-x. 4 root root    68 Jun 13 20:25 tuned
-rw-r--r--. 1 root root  4017 Feb 24  2022 vimrc
```

The first field of the ls -l output is a group of metadata that includes the permissions on each file. Here are the components of the vimrc listing:

- File type: -
- Permission settings: rw-r--r--
- Extended attributes: dot (.)
- User owner: root
- Group owner: root

This string is an expression of three different sets of permissions:

rw-

r--

r--

The first set of permissions applies to the owner of the file. The second set of permissions applies to the user group that owns the file. The third set of permissions is generally referred to as "others." All Linux files belong to an owner and a group.

When permissions and users are represented by letters, that is called symbolic mode. For users, u stands for user owner, g for group owner, and o for others. For permissions, r stands for read, w for write, and x for execute.

When Linux file permissions are represented by numbers, it is called numeric mode. In numeric mode, a three-digit value represents specific file permissions (for example, 744.) These are called octal values. The first digit is for owner permissions, the second digit is for group permissions, and the third is for other users. Each permission has a numeric value assigned to it:

r (read): 4

w (write): 2

x (execute): 1

In the permission value 744, the first digit corresponds to the user, the second digit to the group, and the third digit to others. By adding up the value of each user classification, you can find the file permissions.

For example, a file might have read, write, and execute permissions for its owner, and only read permission for all other users. That looks like this:

- Owner: rwx = 4+2+1 = 7
- Group: r-- = 4+0+0 = 4
- Others: r-- = 4+0+0 = 4

The results produce the three-digit value 744.

Modifying Linux file permissions

We can modify file and directory permissions with the chmod command, which stands for "change mode." To change file permissions in numeric mode, you enter chmod and the octal value you desire, such as 744, alongside the file name. To change file permissions in symbolic mode, you enter a user class and the permissions you want to grant them next to the file name. For example:

```
$ chmod ug+rwx example.txt
$ chmod o+r example2.txt
```

This grants read, write, and execute for the user and group, and only read for others. In symbolic mode, chmod u represents permissions for the user owner, chmod g represents other users in the file's group, chmod o represents other users not in the file's group. For all users, use chmod.

Study of system configuration files in /etc

The '/etc' directory in Linux contains system configuration files that control various aspects of the operating system and installed software.

1. /etc/passwd:
    - The /etc/passwd file stores user account information, including usernames, user IDs, home directories, and default shells.
    - It is readable by all users but can only be modified by the root user or privileged users.
2. /etc/group:
    - The /etc/group file stores information about groups on the system.
    - It lists group names, group IDs, and the users belonging to each group.
    - Similar to /etc/passwd, it is readable by all users but can only be modified by root or privileged users.
3. /etc/shadow:
    - The /etc/shadow file stores encrypted user passwords.
    - It is readable only by the root user, ensuring the security of user passwords.
    - Password hashes and password-related information are stored here.
4. /etc/hosts:
    - The /etc/hosts file contains mappings of IP addresses to hostnames.
    - It is used for local DNS resolution and hostname resolution without relying on DNS servers.
    - Commonly used for defining local network configurations and mapping localhost.
5. /etc/hostname:
    - The /etc/hostname file stores the hostname of the system.
    - It defines the name of the local machine.
6. /etc/resolv.conf:
    - The /etc/resolv.conf file contains the configuration for DNS resolution.
    - It specifies the DNS servers to use for hostname resolution.
7. /etc/fstab:
    - The /etc/fstab file lists the file systems and partitions to be mounted during system boot.
    - It defines mount options, mount points, and file system types.
    - This file is crucial for automounting and managing file system configurations.

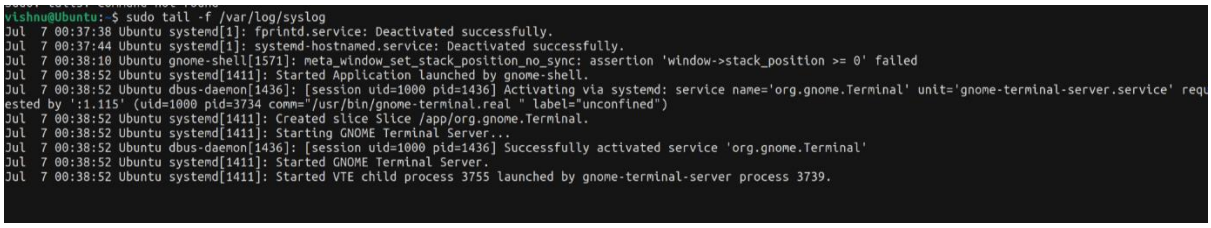
8. /etc/network/interfaces or /etc/sysconfig/network-scripts:
    - These files (location depends on the distribution) contain network interface configuration details.
    - They define network interface settings such as IP addresses, gateways, DNS servers, and more.
9. /etc/sudoers:
    - The /etc/sudoers file controls sudo access for users and groups.
    - It specifies which users or groups can execute commands with administrative privileges using the sudo command.
10. /etc/apt/sources.list:
    - The /etc/apt/sources.list file (specific to Debian-based systems) defines package repositories for package management systems like apt.
    - It lists the repositories from which software packages can be downloaded and installed.

These are just a few examples of system configuration files located in the /etc directory. Each configuration file plays a vital role in controlling system behaviour, security, network settings, and other aspects of the Linux operating system.

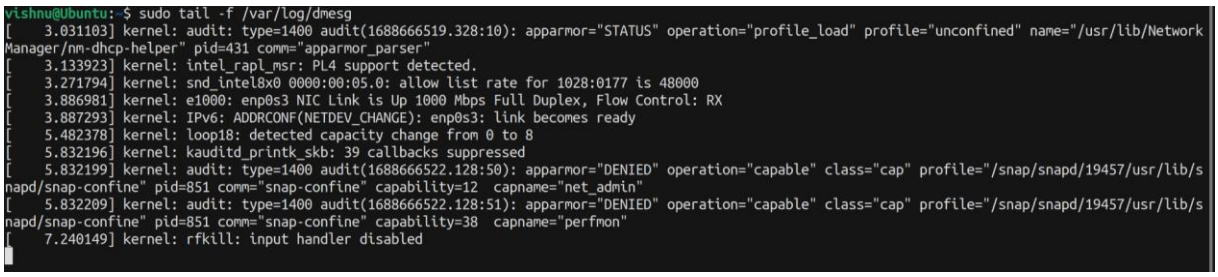
Familiarizing log files for system events, user activity, network events

In Linux, log files are essential for monitoring system events, user activity, and network events. They provide valuable information for troubleshooting, security auditing, and system analysis. Here are some common log files related to system events, user activity, and network events:

1. System Event Logs:
    - /var/log/syslog or /var/log/messages: General system event log capturing various system-related messages, including kernel messages, daemons, and system services.



    - /var/log/dmesg: Kernel ring buffer log displaying boot-time and runtime kernel messages.



    - /var/log/auth.log or /var/log/secure: Authentication logs containing information about user authentication, login attempts, and authentication-related events.

2. User Activity Logs:
    - /var/log/wtmp or /var/log/utmp: User login records tracking user logins, logouts, and system boot/shutdown information.

- /var/log/lastlog: Records the most recent login information for each user.
- /var/log/btmp or /var/log/secure: Records failed login attempts and security-related events.

3. Network Event Logs:
    - /var/log/iptables.log: Logs for the IPTables firewall, capturing network traffic, firewall rules, and related events.
    - /var/log/auth.log or /var/log/secure: Network-related authentication events, such as SSH login attempts.
4. Application-specific Logs:
    - /var/log/apache2/access.log or /var/log/httpd/access_log: Apache web server access logs, recording HTTP requests and client access information.
    - /var/log/apache2/error.log or /var/log/httpd/error_log: Apache web server error logs, capturing server errors, warnings, and debugging information.
    - /var/log/mysql/error.log: MySQL database server error log, containing errors and warnings related to the MySQL service.

RESULT

The familiarization with the Linux file hierarchy provided a comprehensive understanding of the directory structure, including the root directory, system directories, user home directories, and common directories for configuration files, and system resources, enabling efficient navigation, organization, and management of files and directories within the Linux operating system.