LINUX PROGRAMMING

ASSIGNMENT-1

SEPTEMBER 20, 2025

1.What is Linux Operating System (OS)? List three pros and cons of it

Linux is a free, open-source operating system based on Unix. It manages hardware resources, runs applications, and provides essential system services. Unlike proprietary systems (Windows, macOS), Linux is collaboratively developed, and its source code is openly available for anyone to view, modify, and share.

Pros of Linux:

- Fast learning curve. You can learn to use linux quite fast and become a decent average user in about a month or two.
- Linux gives users the power of control and the power of customization, since it's geared more to users with more "technical knowledge". You want to update to a new Linux security update? Entirely up to you. You want to move to a new kernel? Entirely up to you. I really like this aspect as I have always had control.
- Linux is a free, open-source operating system based on Unix. It manages hardware resources, runs applications, and provides essential system services. Unlike proprietary systems (Windows, macOS), Linux is collaboratively developed, and its source code is openly available for anyone to view, modify, and share.

Cons of Linux:

- Complexity can be overwhelming to non-developers—steep learning curve to achieve full system potential
- Unsupported hardware: vendors reluctant to provide open-source device drivers. Non-open, binary-only packages, if provided, must be installed separately from the distribution install. Generic drivers may not provide access to all features of the device.
- Limited apps. Since linux isn't as popular as windows or the mac, and due to linux being massively fragmented, it has led developers to shy away and develop for more prominent Operating systems like windows, leading to a massive gap in apps, especially games. Wine and playOnLinux do help to some extent, but not every app can be played on an emulator.

2 . Differentiate between Linux, Mac, Android, and Windows OS with at least six unique features.

Windows

Proprietary & Paid: Developed and owned by Microsoft, Windows requires a paid license.

Broad Hardware Compatibility: Runs on a wide range of PCs from many different manufacturers.

Large Software Ecosystem: Boasts a vast collection of both paid and free software.

User-Friendly Interface: Known for its generally easy-to-use and familiar interface, even for users without technical expertise.

Case-Insensitive File Names: Treats File.txt and file.txt as the same.

Widespread Gaming Support: Has the most extensive support for video games among the four, making it the preferred platform for PC gaming.

macOS

Apple Exclusive: Exclusively designed for and runs on Apple's hardware (MacBooks, iMacs, etc.).

Optimized Hardware & Performance: Its tight integration with Apple's hardware allows for high optimization and strong performance.

Sleek, Polished Interface: Known for its user-friendly, sleek, and aesthetically pleasing interface.

Proprietary & Paid: Like Windows, it is a proprietary, closed-source operating system.

Unix-Based: Shares Unix roots with Linux, contributing to its stability and power**.**

Strong Ecosystem Integration: Seamlessly integrates with other Apple devices and services.

Android

Mobile-First Design: Specifically designed for smartphones and tablets.

Linux-Based Kernel: Built on the Linux kernel, providing a robust and adaptable foundation.

Extensive App Ecosystem (Mobile): Relies on Google Play Store for mobile applications.

Touch-Centric Interface: Features a user interface optimized for touch input.

Open-Source Components: Uses open-source licenses like Apache 2.0 and GNU GPLv2 for its core.

High Usage in Mobile: Dominates the market for smartphones and tablet computers worldwide.

Linux

Open-Source & Free: The source code is freely available, and the OS itself is free to use.

Highly Customizable: Offers extensive control and customization options for users to tailor the system to their needs.

Command-Line Interface (CLI): Many interactions are through the command line (terminal), providing powerful control.

Case-Sensitive File Names: File.txt and file.txt are treated as distinct files.

Wide-Ranging Device Support: Can be used on a variety of hardware, from servers and embedded systems to desktops.

Security Focus: Its open nature and fewer malware attacks (due to market share) are often cited as advantages for security.

3.Why is Linux preferred for Mainframe Servers for legacy application? Give three out-of-the-box technical reasons.

1. **Hardware Virtualization & Workload Consolidation (z/VM + KVM on z Systems)**

   Linux on IBM Z (mainframes) can run thousands of isolated Linux instances on a single box through z/VM or KVM, with near-native performance.
   This allows legacy applications (often spread across many aging mid-range servers) to be consolidated onto one mainframe while still maintaining isolation.
   Fewer physical servers → less complexity, lower cooling/power costs, and simplified management.

2. **Binary Compatibility & Open Source Ecosystem**

   Linux provides broad backward compatibility layers and open-source libraries that make it easier to wrap or emulate older runtime environments.
   Legacy COBOL, PL/I, and Fortran apps can interoperate with modern middleware (Java, Python, containers) on Linux without rewriting the core business logic.
   This hybrid capability makes Linux an "integration hub" for both legacy and modern workloads

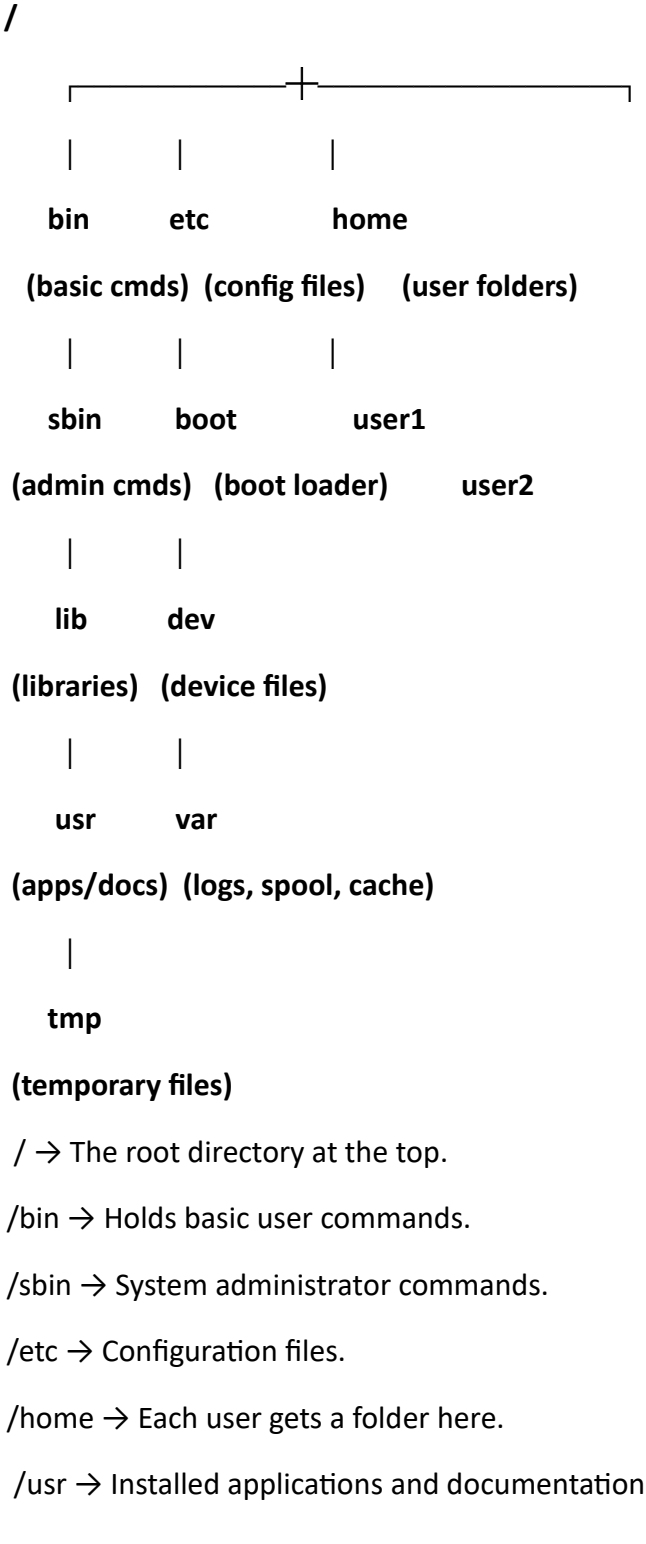
3. **Reliability, Availability, and Serviceability (RAS) at Scale**

   Mainframe hardware already has unmatched RAS features (e.g., fault-tolerant memory, hot-swap CPUs, dynamic reconfiguration).
   Linux on Z is optimized to exploit these hardware features out of the box, so legacy applications inherit enterprise-grade uptime without changes.
   This is critical for industries (banking, insurance, government) where legacy apps cannot tolerate downtime.

4 .Explain the structure of the Linux File System with proper diagram. Note: you can use the tree command to find it out.

- **Linux File System Structure**

The Linux file system is arranged in the form of a tree. At the very top, we have the root directory /, and everything else comes under it. Unlike Windows (which has drives like C:, D:), Linux has only one root, and all files, folders, and even hardware devices are placed under it.



/ → The root directory at the top.

/bin → Holds basic user commands.

/sbin → System administrator commands.

/etc → Configuration files.

/home → Each user gets a folder here.

/usr → Installed applications and documentation.

/var → Log files and variable data.

/tmp → Temporary storage.

/boot → Boot loader and kernel files.

/dev → Device files for hardware.

/lib → Shared libraries for programs.

5. If Linux OS is open-source, how do companies like Red Hat still making money from it? Do a market study and answer properly

- companies like Red Hat make significant money from open-source Linux by selling value that enterprises will pay for

1) What Red Hat actually sells (revenue streams)

1. Subscriptions for enterprise-grade software (RHEL, OpenShift, middleware)

Red Hat sells subscriptions — not the source code. Subscriptions bundle tested binaries, long-term support, security updates, certified integrations, and lifecycle guarantees. Enterprises pay to avoid risk and to get predictable updates and SLAs.

2. Support, SLAs and security/patch services

Paying customers get 24/7 technical support, urgent security patches, backporting of fixes, and legal/patching guarantees that public/free distributions don't commit to. For mission-critical systems this is worth a premium.

3. Cloud & managed offerings (OpenShift, cloud services, partner-managed stacks)

Red Hat OpenShift (Kubernetes distribution) and managed cloud services are sold as enterprise platforms and through partners/cloud marketplaces — a major growth area because companies want managed, certified Kubernetes. IDC/Red Hat materials and market reports show partners capture service revenue for migrations and managed ops.

4. Professional services, consulting, training, certification

Migration planning, architecture services, custom integration, and training/certification courses are high-margin add-ons around the core stack. These are often required for large deployments.

5. Ecosystem & ISV certifications / marketplace revenue

Vendors pay for certification, support integrations, or to be featured in partner marketplaces — this strengthens lock-in and creates indirect revenue.

2) Why customers pay (the value proposition)

Risk reduction & accountability: Enterprises require vendor accountability (patch timelines, legal assurances, security SLAs). Free distros don't promise this.

Compatibility & certification: Red Hat certifies hardware, software stacks, and cloud providers — reducing integration risk.

Operational efficiency: Consolidated tools (RHEL + OpenShift + Ansible) reduce OPEX and accelerate cloud/native adoption.

Compliance & security: Regulated industries need supported patches, audit trails, and vendor support for compliance.

(Evidence: Red Hat / IBM investor materials and market papers emphasize these enterprise drivers.)

3) Market context & scale (why this is a big, sustainable business)

Red Hat's scale inside IBM: IBM reports Red Hat as a strong growth engine within its software segment (double-digit growth in recent quarters; Red Hat revenue growth cited in IBM earnings). This shows consistent enterprise demand for Red Hat offerings.

Cloud native / Kubernetes demand: Managed Kubernetes and related services are expanding quickly (reports show Managed Kubernetes and services growing strongly), which favors vendors offering enterprise Kubernetes (OpenShift).

4) Business model patterns (how open-source + revenue fit together)

"Open core" / Subscription + services hybrid: Core code is open, but Red Hat packages it, tests it, hardens it, and sells the distribution + subscription that enterprises trust. This is a proven model: open code for innovation; paid services for stability and guarantees.

Platform lock-in (via integration & certifications): Customers who standardize on Red Hat's stack (RHEL + OpenShift + Ansible + middleware) gain operational momentum and vendor relationships that make switching expensive.

5) Evidence & numbers (select citations)

IBM/Red Hat financial reporting and investor materials highlight Red Hat's contribution to IBM software revenue and show growth in the Red Hat business.

IDC/Red Hat partner whitepapers stress that OpenShift and managed cloud services create new partner-service revenue streams.

Market reports on Kubernetes and managed services show large, fast-growing demand for managed/container services — a market that Red Hat targets with OpenShift and cloud offerings.

6) Short case example (how a sale typically works)

1. Enterprise picks RHEL + OpenShift subscriptions for production workloads.

2. Red Hat (or partners) provides certified images, migration tools, 24/7 support, and training.

3. Customer pays recurring subscription fees (annual), plus one-time migration and consulting fees.

4. Over time, the customer buys add-ons (Ansible automation, middleware, additional support tiers). This creates recurring revenue and services revenue.

7) Risks & competitive pressures

Commoditization risk: Basic Linux is free, and some customers will use community distros. Red Hat counters this with enterprise features, certifications, and managed services.

Competitors & cloud vendors: Public clouds (AWS, Azure, Google) and other vendors offer managed Kubernetes and proprietary enhancements — Red Hat competes by focusing on multi-cloud compatibility and partner channels.

8) Bottom line (one-line summary)

Red Hat converts open-source software into a large, sustainable business by selling enterprise subscriptions, certified distributions, managed cloud platforms, and professional services — essentially monetizing reliability, accountability, and integration that large organizations require. That's why open source and profitable enterprise business models can — and do — coexist.

6.Write the command to display today's date and time (i.e., current System time).

- Date

7. Which command is used to check how long the system has been running

- uptime

8. What is the difference between shutdown -h now and halt?

shutdown -h now

Tells systemd (or SysV init) to bring the system down safely.

Steps it performs:

1. Sends warning messages to all logged-in users.

2. Stops/terminates all running processes gracefully.

3. Unmounts or remounts filesystems as read-only.

4. Halts the system (and often powers it off if supported).

- Safer because it ensures data isn't lost and services shut down cleanly.

halt

Directly instructs the kernel to stop all CPU functions.

On modern Linux (systemd), it usually just calls the same low-level halt routine, but:

It may not notify users.

It may not stop services cleanly.

Depending on configuration, it may not power off the system, only halt CPU activity.

9. Compare init 0 and shutdown -h. Which is safer? Why?

- **init 0**

  Sends the system into runlevel 0, which is defined as "halt."

  This effectively powers off the system.

  It immediately transitions runlevels, bypassing some of the higher-level shutdown scripts.

  Not all services get a clean shutdown (depends on distro/config).

  Historically intended for administrative use (changing runlevels), not general system shutdowns.

- **shutdown -h**

  Tells the system to halt safely (the -h stands for halt).

  Runs through the shutdown sequence:

  Notifies logged-in users.

  Stops services gracefully.

  Unmounts filesystems properly.

Syncs disks before halting.

Specifically designed for safe power-off.

Which is safer?

- shutdown -h is safer because:

It ensures services are stopped cleanly.

Prevents data loss by syncing/unmounting disks properly.

Is the standard tool for shutting down in Unix/Linux.

init 0 can shut down a system, but it's riskier since it skips the orderly shutdown process in some setups, making shutdown -h the recommended command.

10. A system administrator accidentally powers off a Server machine without shutting it down properly. What problems can occur to the said Server?

If a server is switched off suddenly (like pulling the power cable) instead of shutting it down properly, a few bad things can happen:

1**. Data loss** – Anything still in memory and not yet written to disk can disappear. For example, files you were saving or transactions in a database.

2**. Corrupted files** – The filesystem may get damaged, so the server might need repairs before it works normally.

3. **Services crash badly** – Programs like databases, mail servers, or websites don't get a chance to close safely, so they may not start up cleanly next time.

4**. Hardware stress** – On older spinning hard drives, the read/write head can slam down when power cuts, which can physically damage the disk.

5. **Boot problems** – If important system files were being updated at that moment, the server may fail to boot or drop into recovery mode.

6. **Incomplete logs** – Security and activity logs may be missing entries, making troubleshooting harder.