LINUX PROGRAMMING: ASSIGNMENT-4:

[Document subtitle]

1.Distinguish between man and whatis commands? Justify with proper example.

- grep -o '^[^:]+' /etc/passwd | tee usernames.txt

2.A binary isn't found in $PATH. How would you use commands (which,find, locate) to troubleshoot and fix the issue?

- which mybinary || find / -name mybinary
- 2>/dev/null || locate mybinary
- export PATH=$PATH:/path/to/binary

3.Write a command pipeline that finds all .log files modified in the last 24 hours in /var/log and saves results into log_report.txt.

- find /var/log -name "*.log" -mtime -1 | tee log_report.txt

4.What is the difference between shutdown -r now and reboot?

- shutdown -r now → Gracefully stops all processes, notifies users, unmounts filesystems, then reboots.
- reboot → Immediately reboots (may skip user warning and some shutdown scripts).
- shutdown -r now is safer, while reboot is faster.

5.How can you use the tee command to debug a script that generates both standard output and error messages?

- ./myscript.sh 2>&1 | tee debug.log

6.Explain any three real-world applications of Linux in industries.

Three real-world applications of Linux in industries:

1. Web & Cloud Servers – Most web servers (like Apache, Nginx) and cloud platforms (AWS, Google Cloud, Azure) run on Linux for stability and scalability.

2. Embedded Systems – Linux powers devices like smart TVs, routers, automotive infotainment, and IoT gadgets due to its lightweight and customizable nature.

3. Cybersecurity & Networking – Tools like Kali Linux, firewalls, and penetration testing environments rely on Linux for monitoring, securing, and troubleshooting networks.

7.Differentiate application, system and utility software in the context of Linux environment.

1. Application Software

Purpose: Helps users perform specific tasks.

Examples in Linux: LibreOffice (documents), Firefox (browsing), GIMP (image editing).

2. System Software

Purpose: Manages hardware and provides a platform for applications.

In Linux: The Linux kernel + operating system components that control CPU, memory, file system.

3. Utility Software

Purpose: Provides system maintenance and support tools.

Examples in Linux: tar, grep, top, fdisk, rsync (backup, monitoring, compression, etc.).

8.What are the key differences between open-source and proprietary operating systems?

Key Differences between Open-Source and Proprietary Operating Systems:

1. Source Code Access

Open-Source: Source code is freely available (e.g., Linux).

Proprietary: Source code is hidden and controlled by the vendor (e.g., Windows, macOS).

2. Cost

Open-Source: Usually free to use and modify.

Proprietary: Requires paid licenses.

3. Customization

Open-Source: Highly customizable by users and developers.

Proprietary: Limited customization, restricted by vendor.

4. Support & Updates

Open-Source: Community-driven support, frequent collaborative updates.

Proprietary: Official vendor support with structured updates.

5. Security

Open-Source: More transparent, issues fixed quickly by community.

Proprietary: Relies on vendor patches, slower disclosure.

9.Write the command to display the system's kernel version.

uname -r

This displays the system's kernel version.

10.What is the difference between head and tail commands in text processing?

head → Displays the first lines of a file (default: first 10).

head filename.txt

tail → Displays the last lines of a file (default: last 10).

tail filename.txt

So, head = beginning of file, tail = end of file.