1.In Linux FHS (Filesystem Hierarchy Standard) what is the /?

In the Linux Filesystem Hierarchy Standard (FHS), the / is the root directory. It is the top-level directory in the file system hierarchy and all other directories and files are located beneath it.

2.What is stored in each of the following paths?

/bin, /sbin, /usr/bin and /usr/sbin

/etc

/home

/var

/tmp

/bin: This directory contains essential system binaries that are required for the system to boot and operate properly, such as the bash shell, ls, cp, and rm commands.

/sbin: This directory contains system binaries that are used for system administration tasks, such as mounting filesystems, setting up network interfaces, and configuring system services.

/usr/bin: This directory contains user binaries, which are executable programs that are installed by the system administrator or by the user. These programs are typically not required for system boot and operation, but are commonly used by users.

/usr/sbin: This directory contains system administration binaries that are not essential for system boot and operation, but are used for advanced system administration tasks.

/etc: This directory contains configuration files for the system and applications installed on the system. These files are usually plain text files and are edited by system administrators to configure the system and applications.

/home: This directory contains the home directories of users on the system. Each user has a subdirectory under /home that contains their personal files, settings, and configurations.

/var: This directory contains variable files that are generated by the system and applications. These files can include log files, databases, and other temporary or changing data.

/tmp: This directory contains temporary files that are used by the system and applications. These files are typically deleted automatically when the system reboots or periodically by a system process.

3.What is special about the /tmp directory when compared to other directories?

Web servers have a directory named /tmp used to store temporary files. Many programs use this /tmp directory for writing temporary data and generally remove the data when it is no longer needed.

4.What kind of information one can find in /proc?

Within the /proc/ directory, one can find a wealth of information detailing the system hardware and any processes currently running

5.What makes /proc different from other filesystems?

/proc is very special in that it is also a virtual filesystem. It's sometimes referred to as a process information pseudo-file system. It doesn't contain 'real' files but runtime system information (e.g. system memory, devices mounted, hardware configuration, etc).

6.True or False? only root can create files in /proc

False

7.What can be found in /proc/cmdline?

The /proc/cmdline file contains the command line arguments that were passed to the Linux kernel when it was started. This file is readable by any user and can be viewed using a text editor or the cat command.

8.In which path can you find the system devices (e.g. block storage)?

In Linux, the system devices such as block storage devices can be found in the /dev directory.

Permissions:

9.How to change the permissions of a file?

In Linux, you can change the permissions of a file using the chmod command, which stands for "change mode". The chmod command can be used to modify the permissions for the owner, group, and others on the file.

Syntax:- chmod [options] mode filename

Set read and write permissions for the owner, and read-only permissions for group and others

chmod 644 myfile.txt

Add execute permission for the owner and group

chmod u+x,g+x myfile.sh

Remove write permission for others

chmod o-w myfile.txt

10.What does the following permissions mean?

777: This means that the owner, group, and others have read, write, and execute permissions on the file or directory. This is the most permissive set of permissions and should be used with caution.

644: This means that the owner has read and write permissions, while the group and others have only read permissions. This is a common set of permissions for files.

750: This means that the owner has read, write, and execute permissions, the group has read and execute permissions, and others have no permissions. This is a common set of permissions for directories that should be accessible only to a specific group of users.

11.What this command does? chmod +x some_file

The command chmod +x some_file sets the execute permission for the owner, group, and others on the file named some_file.

12.Explain what is setgid and setuid

Setuid and setgid are a way for users to run an executable with the permissions of the user (setuid) or group (setgid) who owns the file.

Setuid: When the setuid permission is set on an executable file, it allows the file to run with the permissions of the user who owns the file, rather than the permissions of the user who is running the file.

Setgid: When the setgid permission is set on a directory, it makes all files and subdirectories created within that directory to inherit the group ownership of the parent directory instead of the user's primary group.

13.What is the purpose of sticky bit?

The sticky bit is a special permission bit that can be set on a directory in Linux and Unix-based operating systems. When the sticky bit is set on a directory, it means that only the owner of a file can delete or rename that file within the directory, even if other users have write permissions on the directory.

The sticky bit is commonly used on directories that are shared among multiple users, such as the /tmp directory. When the sticky bit is set on /tmp, it means that any user can create files in the directory, but they can only delete or modify files that they own.

14.What the following commands do?

1.chmod: The chmod command is used to change the permissions of a file or directory.

Syntax:- chmod [options] mode file

2.chown: The chown command is used to change the owner of a file or directory.

Syntax:- chown [options] new_owner file

3.chgrp: The chgrp command is used to change the group ownership of a file or directory.

Syntax:- chgrp [options] new_group file

15.What is sudo? How do you set it up?

sudo is a command in Linux and Unix-based operating systems that allows users to run commands with administrative or "superuser" privileges.

To set up sudo on a Linux system, follow these steps:

Log in as the root user, or a user with administrative privileges.

Install the sudo package if it is not already installed. This can typically be done using the package manager of your Linux distribution. For example, on Ubuntu, you can run the command sudo apt-get install sudo.

Create a new user account if necessary using the adduser command. For example, to create a new user called "john", you can run the command sudo adduser john.

Add the new user to the sudo group using the usermod command. For example, to add the user "john" to the sudo group, you can run the command sudo usermod -aG sudo john.

Optionally, edit the sudoers configuration file using the visudo command to specify custom sudo settings, such as which users or groups are allowed to use sudo, which commands they can run, and which passwords or authentication methods are required.

16.True or False? In order to install packages on the system one must be the root user or use the sudo command.

True

17.Explain what are ACLs. For what use cases would you recommend to use them?

ACLs (Access Control Lists) are a more flexible way to control access to files and directories than the traditional Unix file permissions. While Unix permissions only allow a single owner, group, and set of permissions per file or directory, ACLs allow for more granular control over file permissions and can define permissions for multiple users and groups.

With ACLs, you can specify access permissions for specific users or groups, and you can also assign different levels of access for each user or group.

ACLs are especially useful in situations where you need to grant more fine-grained access to files and directories beyond the standard Unix file permissions.

18.You try to create a file but it fails. Name at least three different reason as to why it could happen

Permission Issues

Insufficient Disk Space

Invalid Filename or File Path

19.A user accidentally executed the following chmod -x $(which chmod). How to fix it?

The command chmod -x $(which chmod) removes the executable permission from the chmod command, which makes it impossible to modify the permissions of files and directories using the chmod command.

To fix this issue, the user can follow these steps:

Reboot the system: The easiest solution is to simply reboot the system. This will reset the permission settings for all system files, including the chmod command.

Use a live CD/USB: If the system cannot be rebooted or the issue persists after rebooting, the user can use a live CD/USB to boot into a separate environment and access the system files. From there, the user can use the chmod command to restore the executable permission to the chmod command. The steps are as follows:

Boot the system from a live CD/USB.

Mount the root partition of the affected system to a temporary directory.

Navigate to the directory where the chmod command is located (usually /bin or /usr/bin).

Use the chmod command to restore the executable permission to the chmod command, for example: sudo chmod +x /path/to/chmod

Unmount the root partition and reboot the system.

Copy the chmod command from another system: If the user has access to another system with the same operating system and version, they can copy the chmod command from that system and replace the non-executable chmod command on the affected system. The steps are as follows:

On the healthy system, navigate to the directory where the chmod command is located (usually /bin or /usr/bin).

Copy the chmod command to a USB drive or any other external storage device.

On the affected system, navigate to the same directory and replace the non-executable chmod command with the copy from the healthy system.

Use the chmod command to restore the executable permission to the chmod command, for example: sudo chmod +x /path/to/chmod

Reboot the system.

20.You would like to copy a file to a remote Linux host. How would you do?

To copy a file to a remote Linux host, you can use the scp command, which stands for secure copy. The scp command uses the SSH protocol to securely transfer files between hosts.

The basic syntax of the scp command is: scp [options] source_file destination_file

To copy a file to a remote Linux host using scp, follow these steps:

1.Open a terminal on your local Linux system.

2.Run the following command to copy the file to the remote host: scp /path/to/local/file username@remote.host:/path/to/destination/directory

3.Press Enter and enter your password when prompted.

4.Wait for the file transfer to complete.

21.How to generate a random string?

openssl rand -hex 16

This command generates a random string of 32 hexadecimal characters, which is 16 bytes of random data. You can adjust the length of the string by changing the number after the -hex option. For example, to generate a random string of 8 hexadecimal characters (4 bytes), use -hex 8.

22.How to generate a random string of 7 characters?

openssl rand -base64 5 | cut -c 1-7

23.What is systemd?

systemd is a system and service manager for Linux-based operating systems. It is designed to start up and manage system services and daemons, handle system logging and journaling, and provide an interface for system management and configuration.

---

24. How to start or stop a service?

In a Linux-based operating system using systemd, you can start or stop a service using the systemctl command.

1.Starting a service:sudo systemctl start servicename.service

2.Stopping a service:sudo systemctl stop servicename.service

---

25. How to check the status of a service?

sudo systemctl status servicename.service

---

26. On a system which uses systemd, how would you display the logs?

Displaying all system logs:sudo journalctl

Displaying logs for a specific unit:sudo journalctl -u unitname.service

Displaying logs for a specific time range:sudo journalctl --since "yesterday" --until "now"

---

27. Describe how to make a certain process/app a service

1.Create a systemd unit file:

You can create a new unit file in the /etc/systemd/system directory with a descriptive name, such as myapp.service. The unit file should have the following sections:

[Unit]: Describes the service and its dependencies.

[Service]: Specifies how to start and stop the service, and its runtime options.

[Install]: Specifies how the service should be enabled or disabled.

[Unit]

Example:

Description=My Script

After=network.target

[Service]

Type=simple

ExecStart=/usr/bin/python3 /path/to/myscript.py

Restart=always

[Install]

WantedBy=multi-user.target

2.Reload the systemd daemon:sudo systemctl daemon-reload

3.Start and enable the service:sudo systemctl start myapp.service

sudo systemctl enable myapp.service

---

28. Troubleshooting and Debugging

Understand the problem

Check the logs

Reproduce the issue

Break it down

Use tools and techniques

Keep records

Collaborate and seek help

---

29. Where system logs are located?

System logs in Linux are usually located in the /var/log directory

---

30. How to follow file's content as it being appended without opening the file every time?

To follow a file's content as it's being appended without opening the file every time, you can use the tail command with the -f option. The tail command displays the last few lines of a file by default, but the -f option makes it "follow" the file and display new lines as they are added.

tail -f mylog.txt

---

31. What are you using for troubleshooting and debugging network issues?

Ping: A utility that sends ICMP packets to a target host to check if it is reachable and measure its response time.

Traceroute: A tool that traces the path taken by packets from a source host to a destination host, and identifies any network hops or routers that may be causing issues.

Netstat: A command-line tool that displays active network connections, open ports, and network statistics on a host.

Wireshark: A network protocol analyzer that captures and displays packet-level traffic on a network interface, and allows detailed analysis of network protocols and traffic patterns.

TCPdump: A command-line packet sniffer that captures and displays network traffic on a specific interface, and allows filtering and analysis of captured packets.

Nmap: A network scanner that performs host discovery, port scanning, and OS detection on a target network, and generates reports and maps of network topology.

---

32. What are you using for troubleshooting and debugging disk & file system issues?

df: A command-line utility that displays information about file systems, including disk space usage and file system types.

fsck: A tool that checks and repairs file system integrity issues, such as inconsistencies, corruption, and bad blocks.

lsof: A command-line utility that lists open files and the processes that are accessing them, which can help identify which processes are holding locks on a file system or preventing unmounting.

du: A command-line utility that estimates file space usage, and can help identify large or unexpected files that are consuming disk space.

mount: A command-line utility that shows mounted file systems and their options, and can be used to mount or unmount file systems manually.

smartctl: A tool that provides information about the health and status of hard disk drives and solid-state drives, and can be used to diagnose disk failures and predict disk failures before they occur.

---

33. What are you using for troubleshooting and debugging process issues?

ps: A command-line utility that displays information about active processes, including their process ID (PID), CPU and memory usage, and status.

top: A real-time process monitoring tool that displays information about active processes, including their resource usage and system impact.

strace: A tool that intercepts and logs system calls and signals made by a process, which can help identify issues related to system calls or program execution.

ltrace: A tool that intercepts and logs library calls made by a process, which can help identify issues related to libraries or dynamic linking.

gdb: A command-line debugger that allows interactive debugging of a process, including setting breakpoints, examining memory, and modifying variables.

Systemd service logs: In a Systemd-based system, the logs for a service can be viewed using the journalctl command, which displays system logs in a structured and searchable format.

---

34. What are you using for debugging CPU related issues?

top: A real-time process monitoring tool that displays system resource usage, including CPU usage, memory usage, and process activity.

htop: An interactive process viewer and system monitor that provides a graphical representation of system resources, including CPU usage and memory usage.

perf: A tool that can be used to analyze and diagnose CPU-related issues by sampling CPU performance counters and system events, such as cache misses, instruction executions, and page faults.

strace: A tool that can be used to trace system calls and signals made by a process, which can help identify issues related to CPU usage or program execution.

lsof: A tool that can be used to list open files and the processes that are accessing them, which can help identify processes that are consuming CPU resources due to excessive I/O operations.

System logs: System logs can provide useful information about CPU-related issues, such as system crashes, kernel panics, and hardware failures.

---

35. You get a call from someone claiming "my system is SLOW". What do you do?

Ask the user to describe the symptoms in detail: What is slow specifically? Is it the entire system or just certain applications? Is it slow all the time or only under certain conditions? Gathering as much information as possible can help narrow down the possible causes of the issue.

Check the system resources: Use tools like top, htop, or perf to check the CPU, memory, and disk usage of the system. High resource usage can indicate a bottleneck that may be causing the slow performance.

Check for processes using excessive resources: Use tools like ps or top to check for processes that are consuming a lot of CPU or memory resources. Terminating or adjusting these processes may improve system performance.

Check for hardware issues: Slow system performance can be caused by hardware issues like failing hard drives or failing RAM. Use tools like smartctl or memtest86 to diagnose hardware issues.

Check the system logs: Check the system logs for any error messages or warnings that may indicate a problem. For example, disk errors, kernel panics, or system crashes can all cause slow performance.

Consider network issues: If the system relies on network resources, a slow network connection or network congestion can cause slow system performance. Checking network bandwidth and connection speed may be necessary.

Consider system updates: Ensure that the system is up-to-date with the latest security patches and software updates. Outdated software can cause compatibility issues or security vulnerabilities that can impact system performance.

---

36. Explain iostat output

iostat is a system monitoring utility that reports statistics about input/output (I/O) operations for devices and partitions. The iostat output consists of multiple sections that report different types of information about the system's I/O operations.

The first section of iostat output provides information about CPU usage.

The second section provides information about disk utilization, including the number of reads and writes per second, the number of read and write requests currently in progress, and the average response time for each operation.

The third section reports information about the partitions or devices being monitored by iostat, including the amount of data read and written, the number of read and write requests, and the average response time for each operation.

The final section of the iostat output provides information about the network interface card (NIC) utilization.

---

37. How to debug binaries?

Using a debugger

Debugging symbols

Print statements

Profiling

Dynamic tracing

Binary instrumentation

---

38. What is the difference between CPU load and utilization?

CPU load is a measure of how much work the CPU is doing at any given moment. It is usually expressed as a percentage of the total number of CPU cores available on the system. A CPU load of 100% means that all cores are working at maximum capacity, while a load of 50% means that half of the cores are being utilized.

CPU utilization, on the other hand, is a measure of how much of the CPU's capacity is being used over a certain period of time. It is usually expressed as a percentage of the total available CPU time during that period. CPU utilization takes into account the amount of time the CPU spends idle as well as the time it spends processing tasks.

The main difference between CPU load and utilization is that CPU load is a real-time metric, while CPU utilization is a measure over a period of time. CPU load provides a snapshot of how much work the CPU is doing at any given moment, while CPU utilization provides a more accurate picture of how much of the CPU's capacity is being utilized over a longer period of time.

---

39. How you measure time execution of a program?

Built-in functions

Command-line tools

Performance profiling tools

Benchmarking libraries

40.You have a process writing to a file. You don't know which process exactly, you just know the path of the file. You would like to kill the process

as it's no longer needed. How would you achieve it?

To find the process writing to a specific file, you can use the fuser command in Linux

fuser -v /path/to/file

This will show you the process IDs (PIDs) that are accessing the file. Once you have the PID, you can use the kill command to terminate the process.

kill PID

---

41. What is a kernel, and what does it do?

A kernel is a central component of an operating system that acts as an interface between the software running on a computer and the computer hardware. It is responsible for managing system resources such as the CPU, memory, and I/O devices, and providing an abstraction layer that shields the higher-level software from the details of the underlying hardware.

The kernel performs several essential functions, including:

Memory management

Process management

Device management

Security

System calls

---

42. How do you find out which Kernel version your system is using?

uname -r

---

43. What is a Linux kernel module and how do you load a new module?

A Linux kernel module is a piece of code that can be dynamically loaded and unloaded into the running kernel, extending its functionality without the need to recompile or reboot the entire system.

To load a new module into the kernel, you can use the modprobe command followed by the name of the module.

You can also load modules manually by using the insmod command followed by the name of the module file

---

44. Explain user space vs. kernel space

Kernel space is the privileged mode of operation where the kernel code executes. The kernel code has direct access to the hardware and can perform privileged operations such as controlling hardware devices, allocating and managing memory, and modifying system settings. Kernel space is reserved for the operating system and device driver code, which needs to interact with the hardware directly.

User space, on the other hand, is the mode of operation where applications and user-level processes execute. User space applications cannot access the system hardware directly and must go through the kernel to interact with it. User space processes run in a separate memory space, isolated from the kernel, which provides protection against system crashes and security vulnerabilities.

---

45. In what phases of kernel lifecycle, can you change its configuration?

Compilation: During the kernel's compilation, you can configure its features and options by modifying the kernel's configuration file (usually named ".config"). This file specifies the kernel's configuration options, which determine which features are compiled into the kernel.

Boot Time: During the boot process, you can change kernel configuration settings using the boot loader's configuration files. The boot loader, such as GRUB, reads its configuration file during the boot process, which can include parameters to configure the kernel.

Runtime: While the kernel is running, you can modify its configuration settings using the sysctl command or by modifying the kernel's procfs interface (/proc/sys/). Sysctl is a command-line tool that allows you to view and modify kernel parameters dynamically.

Reboot: Some kernel configurations, such as the kernel's version, cannot be changed at runtime. You need to reboot the system to activate the new kernel version or other kernel configuration settings.

---

46. Where can you find kernel's configuration?

The kernel's configuration file is usually located in the kernel source code directory and is named ".config".

---

47. Where can you find the file that contains the command passed to the boot loader to run the kernel?

The file that contains the command passed to the boot loader to run the kernel is the bootloader configuration file.

---

48. How to list kernel's runtime parameters?

kernel's runtime parameters can be listed by reading the /proc/cmdline file.

---

49. Will running sysctl -a as a regular user vs. root, produce different result?

Yes, running sysctl -a as a regular user versus root will produce different results. This is because many of the kernel parameters exposed through sysctl are only accessible by the root user, due to the sensitive nature of the information they provide.

---

50. You would like to enable IPv4 forwarding in the kernel, how would you do it?

To enable IPv4 forwarding in the kernel, you can use the sysctl command.

1.Open a terminal and log in as root or use sudo command.

2.Check the current value of net.ipv4.ip_forward by running the command:sysctl net.ipv4.ip_forward

This will display the current value of the net.ipv4.ip_forward parameter. A value of 0 indicates that IPv4 forwarding is disabled.

3.To enable IPv4 forwarding, set the value of net.ipv4.ip_forward to 1 by running the command:sysctl -w net.ipv4.ip_forward=1

This will immediately enable IPv4 forwarding in the kernel.

4.To make the change persistent across reboots, edit the /etc/sysctl.conf file and add the following line:

net.ipv4.ip_forward = 1

Save the file and exit.

---

51. How sysctl applies the changes to kernel's runtime parameters the moment you run sysctl command?

When you run the sysctl command, it reads the values of the kernel parameters from the /proc/sys/ virtual filesystem and allows you to modify them. When you change a parameter value, sysctl writes the new value to the corresponding file in the /proc/sys/ directory.

The change in value in /proc/sys/ is propagated to the kernel immediately. In other words, the kernel will immediately start using the new parameter value. So, any changes made using the sysctl command will take effect immediately.

---

52. How changes to kernel runtime parameters persist? (applied even after reboot to the system for example)

Changes to kernel runtime parameters made using the sysctl command are not persistent by default. They only apply to the current running system and will be lost when the system is rebooted.

To make changes persistent across reboots, you can either edit the /etc/sysctl.conf file or create a new file in the /etc/sysctl.d/ directory. These files contain a list of kernel parameters and their values, and are read by the sysctl command during system startup.

For example, to enable IPv4 forwarding persistently, you can add the following line to /etc/sysctl.conf or a new file in /etc/sysctl.d/:

net.ipv4.ip_forward = 1

After you save the file, the changes will be applied automatically on system startup. You can also apply the changes immediately by running the sysctl -p command. This command reads the configuration files and applies the changes to the kernel runtime parameters.

---

53. Are the changes you make to kernel parameters in a container, affects also the kernel parameters of the host on which the container runs?

No, the changes you make to kernel parameters in a container do not affect the kernel parameters of the host on which the container runs. Containers are isolated environments and have their own set of kernel parameters, separate from the host system. Changes made to kernel parameters inside a container will only affect that particular container, and will not propagate to the host or any other container running on the same host.

---

54. What is SSH? How to check if a Linux server is running SSH?

SSH stands for Secure Shell and is a protocol used for secure remote access to a Linux (or other Unix-like) system. It provides a secure encrypted connection between the client and server, allowing users to securely access and manage the server from a remote location.

To check if a Linux server is running SSH, you can use the following command:systemctl status ssh

---

55. Why SSH is considered better than telnet?

SSH is considered better than Telnet for several reasons, including:

Security: SSH uses encryption to secure the connection between the client and server, whereas Telnet sends all data, including usernames and passwords, in plain text. This makes Telnet vulnerable to eavesdropping and man-in-the-middle attacks.

Authentication: SSH provides a more secure method of user authentication compared to Telnet, which relies on a username and password. SSH uses public-key cryptography, which makes it more difficult for unauthorized users to gain access to the system.

Flexibility: SSH allows for remote command execution, file transfer, and port forwarding, making it a more versatile tool than Telnet.

Platform support: SSH is supported by a wide range of platforms, including Linux, macOS, Windows, and mobile devices. Telnet, on the other hand, is largely limited to legacy systems and is not widely used today.

---

56. What is stored in ~/.ssh/known_hosts?

The ~/.ssh/known_hosts file contains a list of host keys for all remote hosts that the user has accessed via SSH. When a user connects to a remote host via SSH for the first time, the host's public key is added to this file on the user's local machine.

---

57. You try to ssh to a server and you get "Host key verification failed". What does it mean?

When you try to ssh to a server for the first time, your SSH client verifies the host's key fingerprint against the one stored in the known_hosts file. If the fingerprints don't match, you'll get a "Host key verification failed" error, which means that your SSH client has detected a possible man-in-the-middle attack.

---

58. What is the difference between SSH and SSL?

SSH (Secure Shell) and SSL (Secure Sockets Layer) are both cryptographic protocols used for secure communication over a network. However, they are used for different purposes and operate at different layers of the network stack.

SSH is primarily used for secure remote access to a system's command line interface. It encrypts the connection between the client and server and provides secure authentication using public key cryptography.

SSL, on the other hand, is primarily used for secure communication between web browsers and web servers, typically over HTTPS. It provides secure authentication, encryption, and data integrity, and is used to protect sensitive information such as passwords, credit card numbers, and other personal data transmitted over the internet.

---

59. What ssh-keygen is used for?

The ssh-keygen tool is commonly used for:

Creating a new SSH key pair (public and private key).

Converting SSH keys to different formats.

Changing the passphrase or password associated with an SSH key.

Adding an SSH key to an SSH agent, which allows for passwordless authentication to remote servers.

Configuring the settings for SSH key generation and management.

---

60. What is SSH port forwarding?

SSH port forwarding (also known as SSH tunneling) is a technique used to securely forward network traffic from one computer to another through an encrypted SSH connection. It allows users to create a secure connection between two computers over an unsecured network, such as the internet, by encrypting all data that passes through the connection.