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

   Root Directory

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

   a. /bin, /sbin, /usr/bin and /usr/sbin

   b. /etc

   c. /home

   d. /var

   e. /tmp

a.

- /bin: contains essential user binaries such as command-line interpreters (like bash), system administration utilities (like su), and other basic commands that are used by all users.
- /sbin: contains essential system binaries, like the binaries for the system daemons, such as init, syslogd, and so on. These binaries are usually used only by the system administrator or during system maintenance.
- /usr/bin: contains most user commands and utilities, including compilers, text editors, and other programming tools.
- /usr/sbin: contains system administration binaries that are used by the system administrator, such as networking tools and daemons.

b. /etc: contains system configuration files that are used by the operating system and many applications. These files include configuration files for the system startup, network configuration, user management, and more.

c. /home: is the directory where user home directories are located. Each user on the system has their own subdirectory within /home where they can store their own files and configurations.

d. /var: contains variable data files, such as log files, spool directories for printing and mail, and other files that change frequently during normal system operation.

e. /tmp: is a directory where temporary files are stored by various applications and processes on the system. These files are typically deleted automatically when they are no longer needed.

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

The /tmp directory is a special directory in that it is designated for temporary file storage. It differs from other directories in that the files stored in /tmp are typically expected to be deleted automatically by the system or application that created them.

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

The /proc directory in Linux contains information about running processes and system resources. Here are some examples of the kind of information that can be found in /proc:

Process information

System information

Kernel configuration

File system information

5. What makes /proc different from other filesystems?

It is a virtual file system

It provides information about running processes

It is read-only

It provides access to kernel data structures

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

False

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

BOOT_IMAGE=/boot/vmlinuz-5.11.0-34-generic root=/dev/sda1 ro quiet splash

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

/dev

9. Permissions

10. How to change the permissions of a file?

    chmod

11. What does the following permissions mean?:

    a. 777 => read,write and execute for owner group and users

    b. 644 => read+write->owner,readonly->group,readonly->users

    c. 750 => r+w+e->owner,read+execute->group,none->users.

13. What this command does? chmod +x some_file

    executable by the user, group, and others.

14. Explain what is setgid and setuid

    Setgid: Permission to inherit group ownership of parent directory

    Setuid: Permission to execute program with owner's privileges (usually root)

15. What is the purpose of sticky bit?

    The purpose of the sticky bit is to restrict the deletion of files within a directory.

16. What the following commands do?

    a. chmod->change permissions

    b. chown->change ownership in a file or directory

    c. Chgrp->change grp ownership in a file or directory

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

Allows a user to run commands with the privileges of another user, usually the superuser or root.

Make sure that the sudo package is installed on your system. You can check if it is installed by running the command sudo -v. If sudo is not installed, you can install it using your

system's package manager. For example, on Ubuntu or Debian, you can run the command apt-get install sudo.

Add your user account to the sudo group. This can be done using the usermod command. For example, to add a user named "john" to the sudo group, you can run the command sudo usermod -aG sudo john.

Configure the sudo settings by editing the /etc/sudoers file. This file contains the rules that determine who can run sudo commands and what commands they can run. You should never edit this file directly, as it can easily break your system. Instead, you should use the visudo command, which opens the /etc/sudoers file in a safe editor and performs syntax checking before saving any changes.

Once you have configured sudo, you can use it to run commands with elevated privileges. To run a command with sudo, simply prefix the command with the sudo keyword. For example, to install a package using apt-get with elevated privileges, you can run the command sudo apt-get install package-name.

Note that sudo is a powerful tool that can be misused if not used carefully. Always be careful when running commands with sudo, and make sure you understand what the command will do before running it.

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

True

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

ACL stands for Access Control List. It is a set of permissions attached to a file or directory that defines who can access the file or directory and what actions they can perform on it. ACLs provide a more granular and flexible way of controlling access to files and directories compared to the traditional Unix-style permissions that only have owner, group, and other categories.

ACLs allow you to specify access permissions for specific users or groups, even if they are not the owner or a member of the owning group of the file or directory. This means that you can grant permissions to individual users or groups without having to create new user accounts or change the ownership or group membership of the file or directory.

ACLs are useful in situations where you need to provide more fine-grained access control to files and directories. For example, in a shared folder where multiple users or groups need to access different files or directories with different levels of permission, ACLs can be used to control access on a per-file or per-directory basis. They can also be used to grant temporary access to a file or directory to a specific user or group without changing the ownership or group membership of the file or directory.

Overall, ACLs provide a more flexible and powerful way of controlling access to files and directories than the traditional Unix-style permissions. They are recommended for use cases where you need to provide fine-grained access control to files and directories, especially in situations where multiple users or groups need to access the same files or directories with different levels of permission.

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

a. File already exists.

b. Permission Issues

c. Disk space Issues.

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

/bin/chmod +x $(which chmod)

Scenarios

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

scp [options] /path/to/local/file username@remote:/path/to/destination

23. How to generate a random string?

openssl rand -base64 12(Using Openssl)

uuidgen | tr -d '\n' | tr -d '-'(Using uidgen)

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

openssl rand -hex 4 | tr -d '\n'

Systemd

25. What is systemd?

systemd is a system and service manager for Linux operating systems. It is designed to replace the traditional SysVinit system initialization and startup scripts, providing a more modern and flexible way of managing system services and processes.

26. How to start or stop a service?

Start:sudo systemctl start servicename

Stop:sudo systemctl stop servicename

27. How to check the status of a service?

sudo systemctl status servicename

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

sudo journalctl

For example, to display logs for a specific service, you can use the -u option followed by the service name, like this:

sudo journalctl -u servicename

Using -p to filter the priorities;

sudo journalctl -p warning

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

Create a new service file in the /etc/systemd/system directory using a text editor. For example:

sudo nano /etc/systemd/system/myapp.service

In the new file, add the following lines:

[Unit]

Description=My App

After=network.target

[Service]

ExecStart=/path/to/myapp

Restart=on-failure

[Install]

WantedBy=multi-user.target

Save and close the file.

Reload the systemd configuration to pick up the new service file:

sudo systemctl daemon-reload

Start the new service using systemctl:

sudo systemctl start myapp

30. Troubleshooting and Debugging

Troubleshooting is the process of identifying the source of a problem or error in a system or application. It involves analyzing symptoms, gathering information, and testing various solutions to isolate the cause of the issue.

Debugging is the process of finding and fixing errors or bugs in software code. It involves using specialized tools and techniques to identify and analyze problems in the code, and then modifying the code to correct the issues.

31. Where system logs are located?

/var/log/journal

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

tail -f /var/log/syslog

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

Netstat

Ping

Tcdump

TraceRoute

Nmap

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

fdisk

Isblk

df

fsck

mount

dmesg

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

ps

top

strace

lsof

gdb

htop

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

top

vmstat

perf

strace

htop

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

Ask the user to provide more information

Check system resource usage

Check for processes using excessive resources

Check system logs

Check disk health

Check network connectivity

38. Explain iostat output

The iostat command in Linux provides information about the input/output (I/O) statistics of your system's disks and partitions. The output is divided into several columns, including:

Device: The name of the disk or partition being monitored

tps (transactions per second): The number of I/O operations completed per second

kB_read/s (kilobytes read per second): The amount of data read from the disk per second

kB_wrtn/s (kilobytes written per second): The amount of data written to the disk per second

kB_dscd/s (kilobytes discarded per second): The amount of data discarded from the disk per second (this value is only shown if using a version of iostat that supports it)

%util (percent utilization): The percentage of time the device was busy servicing I/O requests (i.e. how much it was being used

39. How to debug binaries?

Compile the binary with debugging symbols

Identify the issue

Choose a debugger

Start the debugger

Set breakpoints

Step through the code

Examine variables and memory

Identify and fix the issue

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

CPU Load:

CPU load refers to the amount of work that the CPU is currently processing or has queued up. This can include running processes, waiting processes, and processes that are blocked on input/output operations. A high CPU load can indicate that the system is heavily utilized and may be experiencing performance issues.

CPU Utilization:

CPU utilization, on the other hand, refers to the percentage of time that the CPU is busy processing instructions. It is a measure of how much of the CPU's processing power is being used at a given time. A high CPU utilization can indicate that the system is working hard, but it doesn't necessarily mean that the system is experiencing performance issues.

41. How you measure time execution of a program?

```
start_time=$(date +%s.%N)

    # program code

    end_time=$(date +%s.%N)

    total_time=$(echo "$end_time - $start_time" | bc)

    echo "Total time: $total_time seconds"
```

Use the "time" command

Use a profiler

Add timing code to the program

Use a benchmarking tool

Scenarios

42. 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 that is writing to a file:

lsof /path/to/file

This will display a list of all the processes that have the file open, including their process ID (PID) and the name of the executable file. Look for the process that is actively writing to the file (indicated by the "W" in the "FD" column), and take note of its PID.

To Kill:

kill <PID>

Kernel

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

A kernel is the core component of an operating system. It is the bridge between the hardware and the software, and it manages the system's resources, such as the CPU, memory, and input/output (I/O) devices.

The kernel is responsible for many important tasks, including:

Process management: The kernel manages the processes running on the system, allocating resources such as memory and CPU time.

Memory management: The kernel allocates and deallocates memory for processes and manages virtual memory, allowing processes to use more memory than is physically available.

Device management: The kernel manages input/output (I/O) devices, such as hard drives, network cards, and USB devices.

File system management: The kernel provides access to the file system and manages file and directory permissions, ownership, and attributes.

Network management: The kernel provides network stack and protocols for communication.

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

uname -r

45. 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 into the kernel at runtime.

To load a new kernel:

sudo modprobe <module-name>

46. Explain user space vs. kernel space

User Space:

User space refers to the memory space where user-level applications run. These applications are programs that are executed by users or processes that are initiated by users. In this space, applications have limited access to the system resources and hardware. User space applications can only interact with the operating system through system calls.

Kernel Space:

kernel space refers to the memory space where the operating system kernel runs. The kernel is the core component of an operating system that provides services and resources to user-level applications. Kernel space has unrestricted access to the system resources and hardware, and it is responsible for managing them. The kernel provides a set of system calls that can be used by user-level applications to interact with the system.

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

During:

Kernel Compilation

Boot Time

Runtime

48. Where can you find kernel's configuration?

In the kernel Source Code Directory:

/usr/src/linux

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

GRUB: The GRUB bootloader configuration file is usually located in the "/boot/grub/grub.cfg" file. However, this file is typically generated automatically and should not be edited directly. Instead, the "/etc/default/grub" file should be edited, and changes will be reflected in the generated configuration file.

LILO: The LILO bootloader configuration file is typically located in the "/etc/lilo.conf" file.

SYSLINUX: The SYSLINUX bootloader configuration file is typically located in the "/boot/syslinux/syslinux.cfg" file.

systemd-boot: The systemd-boot bootloader configuration file is typically located in the "/boot/loader/entries" directory, where each kernel version has its own configuration file.

50. How to list kernel's runtime parameters?

    sysctl -a

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

    Yes

When running sysctl -a as a regular user, only the subset of kernel runtime parameters that are accessible to regular users will be displayed. These are typically parameters that are considered safe for non-privileged users to modify, such as network-related settings.

However, when running sysctl -a as the root user, all kernel runtime parameters will be displayed, including those that are restricted to root access. This allows the root user to modify any kernel runtime parameter, which can have a significant impact on the system's behavior.

52.

Open a terminal window.

Check the current value of the net.ipv4.ip_forward parameter by running the following command:

Copy code

```
sysctl net.ipv4.ip_forward
```

This will display the current value of the net.ipv4.ip_forward parameter. If the value is 0, then IPv4 forwarding is currently disabled.

To enable IPv4 forwarding, run the following command:

Copy code

```
sudo sysctl net.ipv4.ip_forward=1
```

This will set the value of the net.ipv4.ip_forward parameter to 1, which enables IPv4 forwarding.

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

Copy code

```
net.ipv4.ip_forward=1
```

This will cause the net.ipv4.ip_forward parameter to be set to 1 at boot time.

Save the changes to the /etc/sysctl.conf file.

To apply the changes immediately, run the following command:

bash

Copy code

sudo sysctl -p /etc/sysctl.conf

This will reload the /etc/sysctl.conf file and apply the changes to the running kernel.

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

Internally, the sysctl command uses the sysctl() system call to communicate with the kernel and modify the value of the specified runtime parameter. The sysctl() system call is a part of the Linux kernel's system call interface, which allows user-space processes to interact with the kernel.

When the sysctl() system call is made to modify a runtime parameter, the kernel validates the new value and updates its internal data structures to reflect the change. This change is immediately visible to all processes that access the affected runtime parameter, as the kernel's internal data structures are shared among all processes.

For example, if you run the following command to modify the net.ipv4.ip_forward parameter:

sudo sysctl net.ipv4.ip_forward=1

The kernel immediately updates its internal data structures to enable IPv4 forwarding. Any subsequent network traffic that passes through the system will be subject to the new value of the net.ipv4.ip_forward parameter.

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

Changes to kernel runtime parameters can be made persistent across reboots by writing the modified parameter values to configuration files that are read at boot time.

In most Linux distributions, these configuration files are located in the /etc/sysctl.d/ directory and have a .conf extension. Each file contains a list of kernel runtime parameters and their associated values, in the same format as the sysctl command

55. 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 inside a container do not affect the kernel parameters of the host on which the container runs.

Containers are isolated from each other and from the host operating system, which means that any changes made to kernel parameters inside a container only affect that container's view of the kernel. Each container has its own virtualized network stack, filesystem, and other system resources, which are separate from those of the host and other containers.

SSH

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

SSH stands for Secure Shell, which is a protocol for securely connecting to a remote computer and executing commands. It is widely used for remote administration, file transfers, and tunneling network traffic.

Command to check idf the linux server is running on SSH:

systemctl status sshd

57. Why SSH is considered better than telnet?

Encryption

Authentication

Integrity

Portability

Better Control

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

A plain text file that contains a list of public host keys for remote servers that have been previously connected to via SSH.

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

occurs when the SSH client fails to verify the authenticity of the remote server's public key.

60. What is the difference between SSH and SSL?

SSH(Secure SHell):

SSH is a network protocol used for secure remote access to servers and other network devices. It provides encrypted communication between a client and a server and supports various authentication methods for establishing a secure connection, such as password authentication, public key authentication, and Kerberos.

SSL(Secured Socket Layer):

SSL, on the other hand, is a protocol used for securing communication between web clients (such as browsers) and servers over the internet. It is commonly used to secure online transactions, such as online banking, e-commerce, and other web-based applications that require secure communication.

61. What ssh-keygen is used for?

ssh-keygen is a command-line tool used for generating, managing, and converting SSH keys.

The ssh-keygen command is used to generate a public and private key pair. The public key is shared with remote servers to authenticate the client, while the private key is kept securely on the client system and used to decrypt the authentication challenge sent by the server.

62. What is SSH port forwarding?

SSH port forwarding, also known as SSH tunneling, is a feature of SSH that allows users to create secure encrypted connections between a local computer and a remote server or network.

There are two types of port forwarding:

Local port forwarding: This forwards traffic from a local port on the client machine to a remote port on the server or network.

Remote port forwarding: This forwards traffic from a remote port on the server or network to a local port on the client machine.