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

- Root directory

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 programs that are required to boot the system and run basic commands.
- /sbin- This directory contains system binaries that are used for system administration tasks.
- /usr/bin- This directory contains non-essential binaries that are installed by the system or by individual users.
- /usr/sbin- This directory contains non-essential system binaries that are used for system administration tasks.
- /etc- This directory contains configuration files for the system and for individual programs.
- /home- This directory contains user home directories. Each user on the system has their own subdirectory in /home, which is typically named after their username.
- /var- This directory contains variable data files for the system and for individual programs.
- /tmp- This directory contains temporary files that are created by the system and by individual programs.

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

directories?

- The files stored in /tmp are volatile and are typically deleted when the system is rebooted, or when they are no longer needed.

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

- The /proc directory in Linux is a virtual file system that provides a way for users and programs to access information about the system and its processes. The files in the /proc directory are not real files, but rather virtual files that are generated by the kernel on the fly as they are accessed.

5. What makes /proc different from other filesystems?

- As mentioned, the /proc filesystem is virtual, meaning that the files it provides don't actually exist on disk. Instead, they are generated by the kernel on the fly as they are accessed.

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

- false

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

- The information contained in /proc/cmdline can be useful for troubleshooting boot problems, identifying system parameters and options, and understanding how the system was configured at boot time.

BOOT_IMAGE=/vmlinuz-5.4.0-73-generic root=/dev/mapper/ubuntu--vg-root ro quiet splash

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

-> In Linux, system devices like block storage devices can be found in the /dev directory.

9. How to change the permissions of a file?

- chmod [options] mode file

10. What does the following permissions mean?:

- 777- This means the owner, group, and others have full read, write, and execute permissions on the file or directory.
- 644- This means the owner has read and write permissions, and the group and others have only read permissions.
- 750- This means the owner has read, write, and execute permissions, the group has read and execute permissions, and others have no permissions.

11.What this command does? chmod +x some_file

- The command chmod +x some_file sets the executable permission (x) for the owner of the file (+) named some_file.

12. Explain what is setgid and setuid

- setgid (Set Group ID) is a permission bit that can be set on a directory. When set on a directory, any new file or directory created within it inherits the group ownership of the parent directory, rather than the group ownership of the user who created the file or directory.
- setuid (Set User ID) is a permission bit that can be set on an executable file. When set on an executable file, the process that is created when the file is executed runs with the privileges of the file's owner, rather than the privileges of the user who executed the file.

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. When the sticky bit is set on a directory, it means that only the owner of a file or directory can delete or rename the file or directory within that directory, even if other users have write permission to the directory.

14. What the following commands do?

- chmod- The chmod command is used to change the permissions of a file or directory. It can be used to add or remove read, write, or execute permissions for the owner, group, or other users.
- chown- The chown command is used to change the owner of a file or directory. It can be used to change the user or group ownership of a file or directory to another user or group
- Chgrp- -The chgrp command is used to change the group ownership of a file or directory. It can be used to change the group ownership of a file or directory to another group.

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

- sudo is a command in Linux that allows authorized users to run commands with the privileges of another user, typically the root user. This is useful when performing tasks that require elevated privileges, such as system maintenance or installation of software.
- sudo usermod -aG sudo your_username

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?

- Access Control Lists (ACLs) are a type of security mechanism in Linux that provide more granular control over file and directory permissions.
- ACLs are useful in scenarios where you need to grant specific access to a subset of users or groups for a file or directory, without changing the existing file permissions. For example, if you have a directory shared by multiple users, you may want to give one user read-only access, while giving another user full read-write access. With ACLs, you can do this without changing the permissions of the entire directory.

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

as to why it could happen

- Permission issues: If the user does not have the necessary permissions to create a file in the directory, the file creation will fail. The user may need write permission to the directory or may not be the owner of the directory.
- File system full: If the file system is full, the creation of new files will fail. This can occur when there is not enough free space on the disk to store the new file.
- Invalid file name: If the file name is invalid, the creation of the file will fail. This could happen if the file name contains special characters or exceeds the maximum file name length allowed by the file system.

19. A user accidentally executed the following chmod -x $(which chmod).

How to fix it?

- Reboot the system in single-user mode, if possible, to avoid any issues related to permissions.
- Log in as root or as a user with sudo privileges.
- sudo chmod +x /bin/chmod
- -rwxr-xr-x 1 root root 106408 Jun 23 12:25 /bin/chmod
- chmod +x /path/to/some/executable/file
- This should restore the execute permission to the file.

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

- scp /path/to/local/file username@remote:/path/to/remote/directory
- If the remote machine is using a non-standard SSH port (i.e. not port 22), add the -P option followed by the port number.
- scp -P 2222 /path/to/local/file username@remote:/path/to/remote/directory
- Enter the password for the remote user when prompted.
- Wait for the transfer to complete. You can monitor the progress of the transfer in the terminal.
- Once the transfer is complete, you can verify that the file was copied by logging into the remote machine and navigating to the destination directory.ame@remote:/path/to/remote/directory

21. How to generate a random string?

- openssl rand -hex 12

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

- tr -dc 'a-zA-Z0-9' < /dev/urandom | head -c 7

23. What is systemd?

- Systemd is a system and service manager for Linux operating systems. It is responsible for managing the core system processes and services, as well as providing a variety of other features such as logging, device management, and network configuration.

24. How to start or stop a service?

- sudo systemctl start service_name
- sudo systemctl stop service_name

25. How to check the status of a service?

- sudo systemctl status service_name

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

- sudo journalctl

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

- To make a process or application a service, you need to create a service file in the /etc/systemd/system/ directory. The service file should include information such as the service name, the command to start the process or application, the user to run the service as, and any other relevant configuration options. Once you have created the service file, you can use the systemctl command to manage the service, such as starting, stopping, enabling it at boot time, and more.
- sudo systemctl start myapp.service
- sudo systemctl status myapp.service
- sudo systemctl enable myapp.service

28. Troubleshooting and Debugging

- Troubleshooting involves systematically identifying the root cause of a problem and determining possible solutions
- debugging is the process of finding and fixing errors or bugs in code or software.

29. Where system logs are located?

- On Windows systems, logs are typically stored in the Event Viewer or in the C:\Windows\System32\winevt\Logs directory.

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

every time?

- You can use the tail command with the -f option to follow a file's content as it is being appended. The -f option stands for "follow", and it causes tail to output the last 10 lines of the specified file and then wait for additional lines to be added to the file
- tail -f /var/log/syslog

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

- Ping
- Traceroute
- Netstat
- Dig

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

system issues?

- fsck
- smartctl
- fdisk
- fdisk
- rsync

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

- ps
- top
- htop
- isof
- journalctl

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

- mpstat
- perf
- strace
- ltrace
- gdb

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

- Check system resources
- Check for running processes
- Check for disk space
- Check for hardware issues
- Check for malware or viruses
- Check network performance

36. Explain iostat output

- iostat is a command-line utility that provides statistics about the input/output (I/O) operations of a Linux system. The output of iostat can help you understand how well the system's disk subsystem is performing, and it can also help you diagnose performance issues related to disk I/O.

1.Device

2.tps

3.kB_read/s

4.kB_wrtn/

5.kB_read

6.kB_wrtn

7.await

8.svctm

9.%util

37. How to debug binaries?

- Compile the binary with debugging symbols using the -g flag.
- Start GDB with the binary as an argument: gdb binary.
- Set any necessary breakpoints with the break command.
- Start the program with the run command.
- Use various GDB commands to examine the program state, including the print command to print variable values and the step command to step through the program line-by-line.
- Use the continue command to resume program execution until the next breakpoint is hit, or until the program exits.

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

- CPU utilization is a measure of the percentage of time the CPU is busy executing instructions. It reflects the amount of work the CPU is doing at any given time. High CPU utilization can be an indication of a CPU bottleneck or an application that is consuming excessive resources.
- CPU load, on the other hand, is a measure of the average number of processes that are using or waiting for CPU resources over a specific period of time. It reflects the overall demand on the CPU and takes into account both the number of processes running and the amount of time they spend waiting for CPU resources.

39. How you measure time execution of a program?

- Using the time command
- Using a profiler
- Using code instrumentation

- Using performance counters

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?

- fuser -v /path/to/file
- kill PID

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

- A kernel is a critical component of an operating system that serves as a bridge between the computer hardware and software. It is the core of the operating system and is responsible for managing system resources, such as memory, CPU time, and input/output (I/O) operations. The kernel is responsible for providing low-level services to other parts of the operating system, such as device drivers, process management, file system management, and security. It also serves as a mediator between software applications and the hardware, enabling applications to interact with the hardware in a controlled and secure manner.

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 kernel. It allows the kernel to add new functionality or device support without requiring a reboot.
- sudo modprobe nvidia

44. Explain user space vs. kernel space

- User space is where user applications run. When a user starts an application, it runs in user space. The user space is protected from accessing critical system resources directly. Applications running in user space can only access hardware and resources through system calls.
- Kernel space is where the kernel and kernel modules run. The kernel is the core component of the operating system, and it manages the hardware and software resources of the system. Kernel space has direct access to the hardware and resources of the system. When an application in user space requires a resource, it sends a system call to the kernel.

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

- During the build time, you can configure the kernel by running the make menuconfig command or any other make target that allows you to configure the kernel options before building the kernel. This includes enabling or disabling certain features, modules, or drivers, setting the scheduler, and more.
- At runtime, you can modify kernel configuration through the /proc filesystem, which allows you to access and change various system parameters such as network settings, memory management, and file system tuning

46. Where can you find kernel's configuration?

- The kernel configuration can be found in the /proc/config.gz file or in the /boot directory with the name config-*. It can also be accessed through the make menuconfig, make xconfig, or make config commands if the kernel source code is available on the system.

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 typically located in the /boot/grub/grub.cfg file. This file contains the configuration information for the GRUB boot loader, which is used to select the operating system to boot at startup. However, it's important to note that this file should not be edited manually, as changes may be overwritten by system updates or other configuration changes. Instead, the grub2-mkconfig command should be used to generate the configuration file based on the current system settings.

48. How to list kernel's runtime parameters?

- You can list the kernel's runtime parameters by checking the contents of the /proc/cmdline file. This file contains the command line arguments passed to the kernel by the bootloader during boot time.
- cat /proc/cmdline

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

- Yes, running **sysctl -a** as a regular user will produce a different result than running it as root. The **sysctl** command allows users to view and modify kernel parameters at runtime. However, certain parameters can only be viewed or modified by the root user. When running **sysctl -a** as a regular user, only the parameters that are accessible to that user will be displayed. When running the same command as root, all parameters, including those that require root privileges, will be displayed.

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

- Open a terminal or console on the system.
- Log in as root or a user with sudo privileges.
- sysctl net.ipv4.ip_forward=1

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

- When a user runs the sysctl command, it reads the new values specified by the user in the command line arguments or configuration file, and then updates the kernel's runtime parameters accordingly. sysctl interacts with the kernel through the proc file system, specifically the /proc/sys directory. When sysctl modifies a parameter, it writes the new value to the corresponding file under /proc/sys. The kernel then reads the new value from this file and applies the change. This means that the changes take effect immediately and persist until the next system reboot unless they are changed again.

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

- to make a change persistent, you can create or edit a .conf file and add the appropriate sysctl settings, using the same syntax as the sysctl command. For example, to make IPv4 forwarding enabled by default, you could create a file named /etc/sysctl.d/99-enable-ipv4-forwarding.
- net.ipv4.ip_forward = 1

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 made to kernel parameters in a container do not affect the kernel parameters of the host on which the container runs. Containers have their own isolated namespaces for various system resources, including kernel parameters

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

- SSH stands for Secure Shell, a protocol for secure communication between two computers. It is commonly used for remote login to a server or for transferring files between two machines.
- systemctl status sshd

55. Why SSH is considered better than telnet?

- Security
- Authentication

- Portability
- Remote access

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

- The ~/.ssh/known_hosts file is used by the SSH (Secure Shell) client to store the public keys of known hosts.

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

- This error message means that the SSH client has detected a potential security threat. It could mean that someone is attempting to perform a man-in-the-middle attack, intercepting your connection and redirecting it to a different server. It could also mean that the server's host key has changed since the last time you connected, which could indicate a security issue or a server configuration change.

58. What is the difference between SSH and SSL?

- SSH is primarily used for secure remote login and file transfer, allowing users to securely access a remote system over an insecure network.
- SSL, on the other hand, is primarily used for secure communication between a web server and a web browser, protecting sensitive data such as passwords, credit card information, and other personal data

59. What ssh-keygen is used for?

- ssh-keygen is a command-line utility used to generate, manage, and convert authentication keys for SSH (Secure Shell) protocol.

60. What is SSH port forwarding?

- SSH port forwarding, also known as SSH tunneling, is a feature in SSH that allows a user to create a secure and encrypted connection between two networked computers or servers.