**Advance Devops-Linux Interview Author : Rohan Sen Version – 3**

**Date-17 Aug 2023**

**This is all I have experienced from professional Interviews Mock and Training, Will put more on version-4 releases on 7 month after Version 3 releases**

**------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

**1.What is Sticky Bits? How to apply Sticky Bits in Linux? [ITC 2023]**

The sticky bit is a special permission bit that can be set on files and directories in Linux. When the sticky bit is set on a directory, it prevents users from deleting or renaming files that they do not own. This is useful for directories that are shared by multiple users, such as /tmp.

To set the sticky bit on a file or directory, you can use the following commands:

**chmod +t <file or directory>**

To unset the sticky bit, you can use the following command:

**chmod -t <file or directory>**

**2.How to add user to a group in Linux? How to add multiple users to a Group in Linux? [TCS 2023]**

**sudo usermod -aG groupname username**

for example, to add the user johndoe to the group wheel, you would type the following command:

**sudo usermod -aG wheel johndoe**

The user will now be added to the group.

To add multiple users to a group, you can use the following command:

**sudo usermod -aG groupname username1 username2 username3 ...**

For example, to add the users johndoe, janedoe, and marydoe to the group wheel, you would type the following command:

**sudo usermod -aG wheel johndoe janedoe marydoe a for appending user to a group**

If you want to add a new user to a group use useradd command :

**useradd -aG wheel johndoe**

**3.What is ACL in Linux? [ITC 2023]**

Access Control Lists (ACLs) in Linux are a way to control who has access to files and directories. ACLs provide more granular control than the standard file permissions, which only allow you to control access by user, group, and everyone else.

With ACLs, you can grant or deny access to individual users or groups, even if they are not the owner of the file or directory. You can also grant specific permissions, such as read, write, or execute, to individual users or groups.

Here are some of the commands you can use to manage ACLs in Linux:

**getfacl: This command shows you the ACL for a file or directory.**

**setfacl: This command sets the ACL for a file or directory**.

To set ACLs, you can use the setfacl command. For example, to give the user johndoe read-only access to the file /etc/passwd, you would use the following command:

**sudo setfacl -m u:johndoe:rw /etc/passwd**

To view the ACL for a file or directory, you can use the getfacl command. For example, to view the ACL for the file /etc/passwd, you would use the following command:

**sudo getfacl /etc/passwd**

**4.How to change ownership of a file or Directory in Linux? [TCS 2023]**

**sudo chown -R johndoe /home/johndoe - R – recursively**

**sudo chown -R new\_owner:new\_group file\_or\_directory 🡪for user and group**

**5.How to see the current Shell using in Linux? [TCS 2023]**

**echo $SHELL**

**6.How to change Shell for a current user in Linux? and what is the file used to change shell in Linux? [TCS 2023]**

**chsh -s new\_shell username**

For example, to change the shell for the user johndoe to the /bin/bash shell, you would type the following command:

**chsh -s /bin/bash johndoe**

To change the shell for a user in the /etc/passwd file, you can use a text editor like vi or nano. Find the line that contains the user's username, and then change the shell field to the path to the new shell you want to use.For example, to change the shell for the user johndoe to the /bin/bash shell, you would change the line that looks like this:

**johndoe:x:1000:1000:John Doe:/home/johndoe:/bin/sh**

to this:

**johndoe:x:1000:1000:John Doe:/home/johndoe:/bin/bash.**

**7.How to add a user in Linux by changing the Shell?**

For example, to add a user named new\_user with the /bin/bash shell and the password password, you would type the following command:

sudo useradd -m -s /bin/bash new\_user

**8.How to add a user by changing the Home Directory?**

To add a user named new\_user with the home directory /home/new\_user and the password password, you would type the following command:

sudo useradd -m -d /home/new\_user new\_user

**9.How to change the Home directory of a existing User?**

For example, to change the home directory for the user johndoe to /home/new\_johndoe, you would type the following command:

sudo usermod -d /home/new\_johndoe johndoe

**10.What are the options available for useradd and usermod command? [Collection Moc]**

Here are some example commands that you can use with the useradd and usermod commands:

**Create a user with the home directory** /home/new\_user **and the shell** /bin/bash**:**

sudo useradd -m -d /home/new\_user -s /bin/bash new\_user

**Change the home directory for the user** johndoe **to** /home/new\_johndoe**:**

sudo usermod -d /home/new\_johndoe johndoe

**Set the comment field for the user** janedoe **to** Jane Doe**:**

sudo usermod -c "Jane Doe" janedoe

**Set the user ID for the user** marydoe **to 1000:**

sudo usermod -u 1000 marydoe

**Set the group ID for the user** peterdoe **to the group** wheel**:**

sudo usermod -g wheel peterdoe

**Set the shell for the user** susan **to** /bin/zsh**:**

sudo usermod -s /bin/zsh susan

**Set the expiration date for the user** thomas **to 2023-03-08:**

sudo usermod -e 2023-03-08 thomas

**Set the inactive time for the user** william **to 90 days:**

sudo usermod -f 90 william

**11.What is YUM and RPM in Linux? What is YUM repository? [X-R Tech 2021]**

YUM stands for Yellowdog Updater Modified. It is a package manager for RPM packages, which are the standard package format for Linux distributions such as Red Hat Enterprise Linux, CentOS, and Fedora. YUM can be used to install, update, remove, and search for RPM packages.

RPM stands for Red Hat Package Manager. It is a package management system that is used to install, update, remove, and query software packages on Linux systems. RPM packages are compressed archives that contain all of the files and dependencies that are needed to install a piece of software.

YUM repository is a collection of RPM packages that are available for installation on a Linux system. YUM repositories are typically hosted on web servers and can be accessed using the yum command.

Conclusion: RPM is the single package for a software if we want to install with all dependency packages YUM is used it is just a collections of RPM.

a YUM repository is a file or a directory that contains all the software packages that are available for installation on a Linux system. It is typically hosted on a web server and can be accessed using the yum command.

A YUM repository is not a single file, but rather a collection of files that are organized in a specific way. The files in a YUM repository typically have the .rpm extension, which stands for Red Hat Package Manager. RPM packages are compressed archives that contain all of the files and dependencies that are needed to install a piece of software.

YUM repositories are connected to the internet because they are hosted on web servers. When you use the yum command to install a software package, YUM will first download the RPM package from the YUM repository to your local system. Once the RPM package has been downloaded, YUM will install the package on your system.

Here are some of the benefits of using YUM repositories:

* They are a central location where all of the software packages for a Linux distribution are available.
* They are easy to use.
* They are updated regularly.

Here are some of the limitations of using YUM repositories:

* They can be slow to download, especially if you have a slow internet connection.
* They can be difficult to manage.
* They are not as widely supported as other package managers, such as apt-get.

Overall, YUM repositories are a convenient way to install software packages on Linux systems. They are easy to use and they provide a central location where all of the software packages for a Linux distribution are available. However, they can be slow to download and they can be difficult to manage.

**12.What is update and upgrade in Linux? How this can be done?**

The update command in Linux does not mean installing new software. Instead, it means replacing the old version of a software package with the latest version that is available in the repository. This can include bug fixes, security patches, and new features.

The upgrade command, on the other hand, does mean installing new software. It will install the latest version of a software package, even if it is not already installed on your system. This can include new features, bug fixes, and security patches.

So, if you want to keep your system up-to-date with the latest security patches and bug fixes, you should use the update command. If you want to install new software or get new features for software that you already have, you should use the upgrade command.

Here is a table that summarizes the differences between the update and upgrade commands:

|  |  |
| --- | --- |
| **Command** | **Description** |
| update | Replaces the old version of a software package with the latest version that is available in the repository. |
| upgrade | Installs the latest version of a software package, even if it is not already installed on your system. |

In Linux, update and upgrade are two different commands that are used to keep your system up-to-date.

Update refers to the process of installing the latest security patches and bug fixes for the software that is already installed on your system.

Upgrade refers to the process of installing the latest version of a software package. This may include new features, bug fixes, and security patches.

How to update and upgrade in Linux:

To update your system, you can use the following command:

**sudo yum update**

This command will check for the latest security patches and bug fixes for all of the software packages that are installed on your system. If there are any updates available, the command will install them.

To upgrade your system, you can use the following command:

**sudo yum upgrade**

**13.What is Patching in Linux? How to Initiate OS patching in Linux Step by Step in Production Server? What are the things need to be in Pre-requisites to initiate OS patching? [TCS 2023]**

To initiate OS patching in Linux step by step in a production server, you will need to follow these general steps:

Identify the Linux systems and versions that need to be patched. This can be done by using a patch management tool or by manually checking the vendor websites for available patches.

1. Scan the Linux systems for missing patches. This can be done using a patch management tool or by manually running the **apt-get update** or **yum update** command.
2. Download the patches to the Linux systems. This can be done using a patch management tool or by manually downloading the patches from the vendor websites.
3. Install the patches on the Linux systems. This can be done using a patch management tool or by manually running the apt-get install or yum install command.
4. Verify that the patches have been installed correctly. This can be done by running the **apt-get check** or **yum check** command.

The following are some of the things that need to be in prerequisites to initiate OS patching:

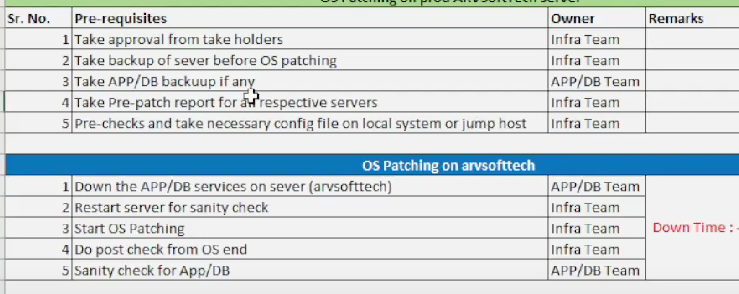
1. A patch management tool or a process for manually managing patches.
2. A way to scan the Linux systems for missing patches.
3. A way to download the patches to the Linux systems.
4. A way to install the patches on the Linux systems.
5. A way to verify that the patches have been installed correctly.

In addition to these prerequisites, it is important to have a plan for how to handle any potential problems that may occur during the patching process. This plan should include steps for backing up the Linux systems, rolling back the patches, and notifying the affected users.

Here are some additional tips for patching Linux systems:

1. Patch regularly. The frequency of patching will depend on the specific Linux distribution and the applications that are running on the systems.
2. Test patches before installing them on production systems. This can be done by installing the patches on a test system or by using a virtualization environment.
3. Back up the Linux systems before patching them. This will help to minimize the impact of any problems that may occur during the patching process.
4. Monitor the Linux systems after patching them. This will help to ensure that the patches have been installed correctly and that they are not causing any problems.

Patching Linux systems is an important part of maintaining the security and stability of the Linux operating system. By following the steps outlined above, you can help to ensure that your Linux systems are properly patched and protected from security vulnerabilities.



**13.What is Patch command in Linux? To patch a single piece of software in Linux?**

The patch command in Linux is used to apply patches to files. Patch files are typically created by the Linux distribution vendor or by the software vendor for a particular application. Patch files contain the difference between the original file and the patched file.

The patch command has a number of options that can be used to control the patching process. Some of the most commonly used options are:

**-p : This option specifies the patch level. The patch level is the number of lines that are skipped before the patch is applied.**

**-i : This option specifies the patch file to use.**

**-R : This option reverses the patch. This is useful if you want to undo a patch that you have already applied.**

**-b : This option creates a backup of the original file before the patch is applied.**

Here is an example of how to use the patch command to apply a patch file :

**patch -p1 -i patch.txt**

This command will apply the patch file patch.txt to the current file. The -p1 option specifies that the patch level is 1, which means that the first line of the patch file will be skipped.

To see a list of all the available options for the patch command, you can run the following command:

**man patch**

Here are some additional tips for using the patch command:

* Always make a backup of the original file before applying a patch. This will help you to undo the patch if it causes any problems.
* Test patches before applying them to production systems. This can be done by applying the patches to a test system or by using a virtualization environment.
* Identify the software that needs to be patched. This can be done by checking the vendor website for available patches or by using a patch management tool.
* Download the patch file for the software. The patch file can be downloaded from the vendor website or from a mirror site.
* Extract the patch file. The patch file is typically a compressed file, so you will need to extract it using a tool like unzip or tar.

Apply the patch file. To apply the patch file, you can use the patch command. For example, to apply a patch file named patch.txt to the file /usr/bin/ls, you would run the following command:

**patch -p1 -i patch.txt /usr/bin/ls**

Verify that the patch was applied correctly. You can verify that the patch was applied correctly by running the ls command. If the patch was applied correctly, the output of the**ls** command should be the same as before the patch was applied.

you can patch an operating system using a patch file. A patch file is a text file that contains the differences between two versions of a file. When you apply a patch file to a file, the patch file will be used to update the file to the newer version.

To patch an operating system using a patch file, you can use the patch command. The patch command takes a patch file as input and applies the differences to the specified file. For example, to patch the operating system file /etc/passwd using the patch file patch.txt, you would run the following command:

**patch -p1 -i patch.txt /etc/passwd**

The -p1 option specifies that the patch level is 1, which means that the first line of the patch file will be skipped.

If you are patching a large operating system file, you may want to use the **-b** option to create a backup of the file before the patch is applied. This will allow you to restore the file if the patch process fails.

**14.Can I apply patch in a command? Like top command?**

Yes, you can apply a patch to the top command. The top command is a utility that displays a live view of the processes running on a Linux system. Patch files can be used to update the top command with new features or to fix bugs.

To apply a patch to the top command, you can use the patch command. The patch command takes a patch file as input and applies the differences to the specified file. For example, to apply the patch file patch.txt to the top command, you would run the following command:

**patch -p1 -i patch.txt /usr/bin/top**

The **-p1** option specifies that the patch level is 1, which means that the first line of the patch file will be skipped.

If you are patching a large file, you may want to use the -b option to create a backup of the file before the patch is applied. This will allow you to restore the file if the patch process fails.

**15.How can I rollback the Patch that has been applied?**

If you use the **-b** option to create a backup of a file before applying a patch, you can roll back the action from the backup file by using the patch command with the **-R** option. The **-R** option tells the patch command to reverse the patch that was applied.

For example, if you applied the patch file patch.txt to the file /usr/bin/top and then decided to roll back the action, you would run the following command:

**patch -R -i patch.txt /usr/bin/top**

This command will restore the file **/usr/bin/top** to its original state before the patch was applied.

It is important to note that the patch command will only restore the file to its original state if you have a backup file of the file. If you do not have a backup file, you will not be able to roll back the action from the patch.

Here are some additional tips for rolling back patches:

Always make a backup of the file before applying a patch. This will help you to restore the file if you need to roll back the action.

Use the -b option when applying a patch. This will create a backup file that you can use to roll back the action.

* Test patches before applying them to production systems. This can be done by applying the patches to a test system or by using a virtualization environment.

**16.After Patching Linux is unable to boot, how can you resolve the issue?**

Boot your server into single-user mode. This can be done by pressing a specific key during the boot process. The key that you need to press will vary depending on your server's manufacturer.

Here are some of the most common keys that you can press:

* Ctrl+X: This is the most common key to press to enter single-user mode.
* e: This key will allow you to edit the GRUB boot options.
* c: This key will boot your server into a command-line prompt.
* Once you have pressed the correct key, your server will boot into single-user mode. You will then be able to access the command line.
* Once you are in single-user mode, you can take the appropriate steps to resolve the issue. For example, if the GRUB configuration file is corrupted, you will need to recreate it.
* Here are some of the commands that you can use to troubleshoot the issue:
* sudo cat /etc/grub.conf: This will show you the contents of the GRUB configuration file. If the file is corrupted, you will see errors.
* sudo cat /var/log/kern.log: This will show you the kernel log. The kernel log will contain information about any errors that occurred during the boot process.
* sudo grub-install /dev/sda: This will reinstall GRUB.
* sudo dracut --force: This will recreate the initramfs image.
* Once you have resolved the issue, you can exit single-user mode by running the following command:
* exit

Your server should now be able to boot normally.

**17.Describe Linux Booting process? Step by step?**

1. **BIOS/UEFI Initialization:** The BIOS or UEFI firmware is executed when you turn on your computer. The BIOS/UEFI performs a Power-On Self Test (POST) to check the hardware components like the CPU, memory, storage, etc., for proper functioning.
2. **Master Boot Record (MBR) or UEFI Firmware Boot Manager**: If your system uses MBR, the BIOS transfers control to the MBR located in the first sector of the boot device. In the case of UEFI, the firmware directly looks for an EFI System Partition (ESP) and reads the bootloader information from it.
3. **Bootloader Execution**: The MBR or UEFI Boot Manager then loads the bootloader into memory. The most common Linux bootloader is GRUB (GRand Unified Bootloader). GRUB displays a menu (if multiple operating systems are installed) and waits for a user selection, typically for a few seconds. Once the timeout expires or the user selects an entry, GRUB proceeds to load the Linux kernel.
4. **Linux Kernel Loading**: GRUB locates the kernel file (usually named "vmlinuz") based on the selected menu entry or default configuration. The kernel is loaded into memory, and GRUB passes control to the kernel.
5. **Kernel Initialization:** The Linux kernel takes control of the system. It performs various tasks, such as setting up memory management, initializing drivers for essential hardware components, and enabling the necessary features. The kernel then mounts the root file system specified in its configuration. The root file system contains critical system files and directories needed for the OS to run.
6. **Init System or Systemd Initialization (SysVinit or Upstart in older systems):** The kernel starts the first userspace process, called "init" (or "systemd" in modern systems), which has process ID 1. The init/systemd process initializes the entire user space, bringing up essential system services, daemons, and background processes. During this stage, various system services are started, network interfaces are initialized, and other user space components are loaded.
7. **Runlevels and Services (SysVinit):** In traditional SysVinit systems, the init process switches to a specific runlevel, which determines the set of services to be started or stopped. Different runlevels correspond to different modes, such as single-user mode, multi-user mode with a graphical interface, or shutdown. Each runlevel has associated scripts in the /etc/init.d directory that control which services are started or stopped.
8. **Login Prompt:** Once the initialization process completes, the system is ready to accept user logins. If the system is configured to start in graphical mode, the login manager (e.g., GDM, LightDM) displays the graphical login screen. If the system starts in text mode, a text-based login prompt appears.
9. **User Session and Desktop Environment (Optional):** After successful authentication, the user's session is started. If a desktop environment is installed (e.g., GNOME, KDE, Xfce), it will be loaded, providing the user with a graphical interface. Once the Linux boot process reaches this stage, the system is fully initialized, and users can interact with the operating system through the terminal or graphical user interface.

**BIOS🡪MBR🡪GRUB🡪KERNEL🡪INIT**

**18.What is the difference between systemd and init? [TCS 2023]**

**1.Fast Startup :**

* SysVinit: Follows a sequential startup method, where services are started one after another, often leading to longer boot times, especially on systems with numerous services.
* Systemd: Adopts a parallel startup method, which allows services to start simultaneously when possible. This approach leverages modern hardware capabilities and multiple CPU cores to reduce boot times significantly.

**2. Service Management:**

* SysVinit: Uses shell scripts located in the /etc/init.d directory to manage services. Each runlevel has its set of scripts to control the services to be started or stopped.
* Systemd: Employs unit files (usually stored in /etc/systemd/system and /lib/systemd/system directories) to define services, targets (equivalent to runlevels), and other system resources. The unit files are typically written in a structured INI-like syntax and provide more advanced service management features.

**3.Logging and Journaling:**

* systemd introduced the concept of the systemd Journal (journalctl), which replaces traditional syslog. The journal captures more structured log data, making it easier to analyze and retrieve information about system events and service status.

**4.Service Management:**

* systemd provides robust tools for managing services, including easy enable/disable operations, restarting, monitoring, and dependency tracking.

**5. Socket Activation:**

* systemd supports socket activation, allowing services to be started only when needed. This can further improve performance by avoiding the unnecessary startup of services that aren't immediately required.

Overall, systemd is more feature-rich and offers significant improvements over the traditional init system, making it the preferred choice for many modern Linux distributions. However, it is worth noting that some users and distributions still prefer init for its simplicity and compatibility with older systems.

**19.How to see Installed packages in Linux? Check httpd is installed or not?**

* **yum install <package name>**
* **yum install httpd**

If you want to see more information about a specific package, you can use the rpm -q command:

* **rpm -qa httpd**

**20.How to List the file installed by the particular package in Linux?**

* **rpm -ql httpd**

**21.How to see which rpm is responsible for creating a particular file example /etc/httpd?**

* **rpm -qf /etc/httpd**

**22.What is the difference between /bin and /sbin? [Zilo-tech Mock]**

The /bin and /sbin directories in Linux are both used to store executable files, but they have different purposes.

The /bin directory contains executable files that are needed for the basic operation of the system. These files are typically used by users and administrators, and they do not require root privileges to run. Some examples of files in the /bin directory include:

* ls: Lists the contents of a directory
* cat: Displays the contents of a file
* rm: Removes a file
* mkdir: Creates a new directory

The /sbin directory contains executable files that are needed for system administration tasks. These files require root privileges to run, and they are typically not used by regular users. Some examples of files in the /sbin directory include:

* fdisk: Partitions a hard drive
* mount: Mounts a filesystem
* reboot: Reboots the system
* shutdown: Shuts down the system

In general, you should only use files in the /sbin directory if you are a system administrator or if you are troubleshooting a problem with your system. If you are a regular user, you should only use files in the /bin directory.

**23.What is umask and how umask is calculated?**

In computing, umask is a value that determines the default permissions of newly created files and directories. It is a mask used to subtract certain permissions from the default permission set, effectively limiting the permissions granted to new files and directories.

The umask value is subtracted from the maximum possible permission (usually 777 for directories and 666 for files) to calculate the default permission of newly created files and directories.

The umask value is represented in octal notation, and it consists of three digits that represent the permissions to be masked for the owner, group, and others, respectively. Each digit corresponds to read (4), write (2), and execute (1) permissions. Setting a bit to 0 in the umask means that permission will be allowed, while setting it to 1 means it will be denied.

Here's how umask is calculated:

Start with the maximum possible permission for files (666) and directories (777).

Convert the umask value to octal. For example, if the umask is set to 022, its octal representation is 022.

Subtract the octal umask value from the maximum permissions:

For files, subtract the umask value (022) from 666: 666 - 022 = 644

For directories, subtract the umask value (022) from 777: 777 - 022 = 755

So, if the umask is set to 022, any new files will have default permissions of 644 (rw-r--r--) and new directories will have default permissions of 755 (rwxr-xr-x).

The umask is typically set in the shell configuration files like ~/.bashrc, ~/.bash\_profile, or system-wide configuration files like /etc/profile, /etc/login.defs, or /etc/bashrc.

**24.What is Run level in Linux? How to change run level in Linux? [Red-Hat System-Admin 2019]**

In Linux, runlevels are predefined operating states that determine which services and daemons are active on the system. Each runlevel represents a specific mode of operation, and the services running in each runlevel can vary depending on the distribution and configuration.

Traditionally, Linux systems used runlevels from 0 to 6, each with a different purpose:

0: Halt - System shutdown.

1: Single User Mode - Used for maintenance and administrative tasks.

2: Multi-User Mode without Networking - A basic multi-user mode without network services.

3: Multi-User Mode with Networking - The standard multi-user mode with networking.

4: Not used (custom usage on some systems).

5: Graphical User Interface (GUI) Mode - Multi-user mode with a graphical desktop environment.

6: Reboot - System restart.

Modern Linux distributions may use a slightly different set of runlevels or use **systemd** targets instead, but the concept remains similar.

**To see the current runlevel in Linux, you can use the runlevel command or check the /run/utmp file using the who -r command:**

Using runlevel command:

**Switch to runlevel 0 (Halt - System shutdown):**

sudo init 0 # For systems using SysV init

sudo telinit 0 # For systems using SysV init

sudo systemctl poweroff # For systems using system

**Switch to runlevel 1 (Single User Mode):**

sudo init 1 # For systems using SysV init

sudo telinit 1 # For systems using SysV init

sudo systemctl rescue # For systems using system

**Switch to runlevel 2 (Multi-User Mode without Networking):**

sudo init 2 # For systems using SysV init

sudo telinit 2 # For systems using SysV init

sudo systemctl isolate multi-user.target # For systems using system

**Switch to runlevel 3 (Multi-User Mode with Networking):**

sudo init 3 # For systems using SysV init

sudo telinit 3 # For systems using SysV init

sudo systemctl isolate graphical.target # For systems using systemd, if you want to switch to a text-based multi-user mode, use: sudo systemctl isolate multi-user.target

**Switch to runlevel 5 (Graphical User Interface - GUI Mode):**

sudo init 5 # For systems using SysV init

sudo telinit 5 # For systems using SysV init

sudo systemctl isolate graphical.target # For systems using system

**Switch to runlevel 6 (Reboot - System restart):**

sudo init 6 # For systems using SysV init

sudo telinit 6 # For systems using SysV init

sudo systemctl reboot # For systems using systemd

Please note that the availability of specific runlevels and their purposes might differ based on the Linux distribution and version you are using. Also, in modern systems using systemd, runlevels are often replaced with target units.

**25. How In Linux, switching to a different run level without disturbing the user?**

Switching to a different run level without disturbing the user in Linux means changing the system's operating mode without interrupting the currently logged-in users or terminating critical services.

To achieve this, modern Linux systems use the systemd init system. With systemd, run levels are replaced by "targets." Each target corresponds to a specific system state, and switching between targets should maintain user sessions and essential services.

Here's how you can switch between targets (run levels) using systemd in an easy way:

**Check the Current Target (Run Level):**

To see the current target, use the following command:

systemctl get-default

It will show you the currently active target, which could be multi-user.target (equivalent to run level 3) or graphical.target (equivalent to run level 5).

**Switch to a Different Target (Run Level):**

To switch to a different target, use the following command with sudo (administrative privileges):

**To switch to a text-based multi-user mode (equivalent to run level 3):**

sudo systemctl isolate multi-user.target

**To switch to a graphical user interface (GUI) mode (equivalent to run level 5):**

sudo systemctl isolate graphical.target

That's it! The system will transition to the new target without disturbing the active user sessions or abruptly stopping essential services.

Switching run levels (targets) using systemd is designed to be non-disruptive, but as a good practice, it's still recommended to save your work and close important applications before making significant changes to the system's state.

By using these simple commands with administrative privileges, you can easily switch between different run levels (targets) in Linux without causing any disruptions to the users or services running on the system.

Using **isolate** is a clean way to switch between different targets (run levels) without causing disruptions, as it ensures a smooth transition to the new target without starting or stopping unrelated services.

**26. How to check CPU usage in Linux? [NIC 2020]**

top command:

Open a terminal and type top to launch the top command. It displays a real-time dynamic view of system processes, including CPU usage, memory usage, running processes, and more.

The CPU usage is shown as a percentage for each core, and the overall CPU usage is displayed at the top of the output.

htop command:

If you have htop installed, you can use it as an alternative to top. htop provides a more user-friendly and colorful interface with additional features.

To use htop, simply type htop in the terminal.

uptime command:

The uptime command provides a quick overview of system load and how long the system has been running. It also shows average CPU load for the last 1, 5, and 15 minutes.

Open a terminal and type uptime to see the information.

sar command:

The sar (System Activity Reporter) command can display historical system performance data, including CPU usage, over a specified period.

To use sar, you might need to install the sysstat package first. Then, you can run sar with specific options, such as sar -u to show CPU usage.

mpstat command:

The mpstat command displays detailed processor-related statistics, including individual core usage, CPU interrupts, context switches, and more.

Open a terminal and type mpstat to see the CPU statistics.

/proc/stat file:

The /proc/stat file contains various system statistics, including CPU usage information.

You can use cat /proc/stat to see the raw content, but it might not be very user-friendly. For a better interpretation, you can parse the data using a script.

Keep in mind that CPU usage can fluctuate rapidly, so using commands like top or htop is more suitable for observing real-time changes. For more detailed historical analysis, sar or custom monitoring solutions are better options.

**27.How to check complete configuration of your system in Linux?**

Here are the commands to check the complete configuration of your system in Linux, briefly:

uname - Shows the name, kernel, and version of your operating system.

lshw - Shows detailed information about your hardware, including your CPU, memory, storage, and network devices.

dmidecode - Shows information about your motherboard, including the BIOS version, serial number, and other hardware details.

cat /proc/cpuinfo - Shows detailed information about your CPU, including the number of cores, clock speed, and cache size.

cat /proc/meminfo - Shows detailed information about your memory, including the amount of RAM installed, the free memory, and the used memory.

cat /etc/fstab - Shows the list of all your filesystems, including the type of filesystem, the mount point, and the UUID.

systemctl status - Shows the status of all the services running on your system.

To run these commands, you will need to open a terminal window and type the command. The command will output a lot of information, so you may need to scroll through it to find what you are looking for.

**28.How to check RAM in Linux?**

To check RAM in Linux –

free -h

vmstat

cat /proc/meminfo

**29.What is Load Average? Describe uptime command?**

The uptime command is a simple and useful tool available in most Linux and Unix-like operating systems. It provides a quick summary of three key pieces of information about the system:

Current Time: The uptime command displays the current time of the system when you execute it. This is helpful for knowing the exact time of the system status report.

System Uptime: The second part of the uptime output indicates how long the system has been running since the last boot. It displays the time in hours and minutes (HH:MM) or days and hours if the system has been running for more than 24 hours. This information gives you an idea of how long the system has been up without any reboots.

Load Average: The last part of the uptime output presents the average system load over the last 1, 5, and 15 minutes, respectively. The load average is a measure of how many processes are waiting to be executed by the CPU. It is a crucial indicator of system workload. The three values represent the load average over the short term (1 minute), medium term (5 minutes), and long term (15 minutes). The values are represented as decimal numbers, where a value below 1.0 typically indicates a lightly loaded system, while values above 1.0 may indicate higher CPU utilization or increased system demand.

Here's an example of uptime output:

**08:32:47 up 1:23, 2 users, load average: 0.25, 0.38, 0.42**

In this example:

Current time: 08:32:47

System uptime: 1 hour and 23 minutes

2 users are currently logged in.

Load average: 0.25 (1-minute), 0.38 (5-minute), 0.42 (15-minute)

The uptime command is particularly useful for quickly checking the system's health, its uptime since the last reboot, and getting a rough estimate of its current workload. It can be beneficial for monitoring server performance or identifying potential issues related to high system loads.

**30. You have installed a new version of Java on your Linux system, but the system is still pointing to the older version. How you resolve this? [Infinite-2023 ] ?**

If you have installed a new version of Java on your Linux system, but the system is still pointing to the older version, it is likely due to the order of directories in the system's PATH environment variable. The PATH variable determines the order in which the system searches for executable files, including Java.

Here's how you can resolve the issue and make your system use the newly installed version of Java:

Check the installed Java versions:

First, verify that the new version of Java is installed on your system and its location. You can use the java -version command to check the default Java version and its location.

java –version

This will display the currently active Java version and its installation path.

Update the PATH variable:

To ensure that the system uses the new Java version, you need to update the PATH variable to include the directory where the new Java executable is located. Make sure to put the directory of the new Java version ahead of the old one in the PATH.

**Open your shell configuration file (.bashrc, .bash\_profile, .zshrc, etc., depending on your shell) using a text editor:**

nano ~/.bashrc

Add or update the JAVA\_HOME and PATH variables:

In the shell configuration file, set the JAVA\_HOME variable to the installation directory of your new Java version and add the new Java's bin directory to the beginning of the PATH variable. For example:

export JAVA\_HOME=/path/to/your/new/java/installation

export PATH=$JAVA\_HOME/bin:$PATH

Replace /path/to/your/new/java/installation with the actual path to your new Java installation.

Save and apply the changes:

Save the file (Ctrl+O in nano) and exit the text editor (Ctrl+X in nano).

**To apply the changes, you can either restart your terminal or run the following command:**

**Reload the file**

**source ~/.bashrc**

Verify the Java version:

After updating the PATH, check the Java version again using:

java -version

It should now display the new version of Java.

By updating the PATH environment variable and setting JAVA\_HOME correctly, your system should now use the newly installed version of Java instead of the older one.

**31. Different between /bin and /sbin?**

- /bin (short for binary) contains essential user commands that are required by both the system administrator and regular users,

such as ls, cp, rm, mkdir, cat, grep.

- sbin /sbin (short for system binary) also contains essential commands, but they are primarily intended for use by the system administrator commands such as fdisk,iptable etc.

**32.What is crontab in Linux?**

- cronjob is used to schedule task to run in a particular date/time automatically

- crontab -l <To list jobs>

- crontab -e <To edit crontab>

\* \* \* \* \* /path/to/command ./arg2

- - - - -

| | | | |

| | | | ----- Day of the week (0 - 7) (Sunday is both 0 and 7)

| | | ------- Month (1 - 12)

| | --------- Day of the month (1 - 31)

| ----------- Hour (0 - 23)

------------- Minute (0 - 59)

\* \* \* \* \* /path/to/command arg1 arg2 # Runs the command every minute

0 0 \* \* \* /path/to/command arg1 arg2 # Runs the command every day at midnight

0 12 \* \* Mon-Fri /path/to/command arg1 arg2 # Runs the command every weekday at noon

30 8 1-10 \* \* /path/to/command arg1 arg2 # Runs the command at 8:30 am on the first 10 days of every month

0 0 1,15 \* \* /path/to/command arg1 arg2 # Runs the command at midnight on the 1st and 15th of every month.

**33. Discuss about of securities in Linux? [TCS PRO 2018]**

Access control: Linux provides a robust system of access control that can limit user access to specific files, directories, and system resources. This can be implemented using file permissions, ownership, and access control lists (ACLs).

Firewalls: Linux distributions typically include a built-in firewall that can be used to control network traffic and prevent unauthorized access to the system. Firewall rules can be configured using tools such as iptables or firewalld.

Encryption: Linux supports a variety of encryption methods, including full disk encryption, file-level encryption, and network encryption. This can be used to protect sensitive data from unauthorized access.

User management: Effective user management is critical to maintaining the security of a Linux system. This includes practices such as limiting user privileges, enforcing strong passwords, and implementing multi-factor authentication.

Updates and patching: Keeping a Linux system up to date with the latest security patches and updates is essential for maintaining system security and preventing vulnerabilities from being exploited.

**34. What is w and who command define its output? [CTS 2020]**

The w and who commands are used to display information about users who are currently logged in to a Linux system.The w command provides a summary of each user's current activity, including their username, the terminal or pseudo-terminal they are using, their remote hostname or IP address if they are connected via a network, their login time, idle time, and the current process they are running. The output also includes the load average of the system, which provides an indication of how busy the system is.

Here's an example output of the w command:

16:01:11 up 1:43, 2 users, load average: 0.01, 0.09, 0.09

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

user1 pts/0 192.168.1.2 14:35 1:25m 0.23s 0.23s -bash

user2 pts/1 192.168.1.3 15:45 0.00s 0.05s 0.00s w

The who command, on the other hand, provides a more simplified output that displays only the username, terminal or pseudo-terminal, login time, and remote hostname or IP address if applicable.

Here's an example output of the who command:

user1 pts/0 2022-04-01 14:35 (192.168.1.2)

user2 pts/1 2022-04-01 15:45 (192.168.1.3)

Both the w and who commands provide useful information about currently logged in users and their activity, which can be helpful for system administration, troubleshooting, and security monitoring purposes.

**35.What is SElinux security what is setenforce and getenforce? SElinux configuration file? [TCS Projects 2023]**

SELinux (Security-Enhanced Linux) is a Linux kernel security module that provides a set of security policies and mechanisms to control access to system resources and enforce mandatory access control (MAC) on Linux systems. SELinux allows administrators to define fine-grained security policies that can limit the actions of users, applications, and processes based on their role, context, and type.

- getenforce to see the policy

- sudo setenforce enforcing to set policy

Enforcing - SELinux is enabled and enforcing its security policies.

Permissive - SELinux is enabled but only generating alerts and not enforcing its policies.

Disabled - SELinux is not enabled.

/etc/selinux/config

# This file controls the state of SELinux on the system.

# SELINUX= can take one of these three values:

# enforcing - SELinux security policy is enforced.

# permissive - SELinux prints warnings instead of enforcing.

# disabled - No SELinux policy is loaded.

SELINUX=enforcing

# SELINUXTYPE= can take one of three two values:

# targeted - Targeted processes are protected,

# minimum - Modification of targeted policy. Only selected processes are protected.

# mls - Multi Level Security protection.

SELINUXTYPE=targeted

# SETLOCALDEFS= Check local definition changes

SETLOCALDEFS=0

**36. How to see the particular process in Linux? [Zenforce mock]**

- ps -ef| grep process name

- ps -ef| grep web

**37.How to kill a particular process in Linux**?

- ps aux | grep myprocess

- kill -9 12345

**38.What is nice value?how to set nice value?**

In Linux, the nice value is a parameter that determines the priority of a process. The nice value ranges from -20 (highest priority) to 19 (lowest priority), with 0 being the default value.

- nice -n 10 myprocess

- renice -n 5 -p 11230 p = process id

**39.What are process state in Linux?[HCL 2021]**

Type top command and press Enter.

The top command displays a real-time view of the processes running on your system, along with other system metrics.

The process state is displayed in the "S" column, which stands for "state".

The process state is represented by a single character: R for Running, S for Sleeping, D for Uninterruptible Sleep, Z for Zombie, T for Stopped, and so on.

Running (R): The process is currently running or executing.

Sleeping (S): The process is sleeping or waiting for an event to occur, such as I/O or a signal.

Interruptible Sleep (S): The process is sleeping, waiting for an event to occur, and can be interrupted by a signal.

Uninterruptible Sleep (D): The process is sleeping and waiting for an event to occur, but cannot be interrupted by a signal.

Zombie (Z): The process has completed execution but still has an entry in the process table. The process is essentially dead but has not been removed from the process table yet.

Stopped (T): The process has been stopped, usually by a signal such as SIGSTOP or SIGTSTP.

**40.Type of Backup in Linux? [HCL 2021]**

In NIC we take snapshot and Tapebackup on DELL EMC XTREAMIO

1. Full backup: A full backup is a complete copy of all files and directories on a system.

This type of backup takes the longest to complete and requires the most storage space,

but it provides the most comprehensive backup and makes it easy to restore the system to its original state.

2. Incremental backup: An incremental backup only copies files that have changed since the last backup,

whether it was a full backup or another incremental backup.

This type of backup takes less time and storage space than a full backup, but it requires multiple backups to fully restore the system.

3.Differential backup: A differential backup only copies files that have changed since the last full backup.

This type of backup takes less time than an incremental backup and requires fewer backups to fully restore the system, but it still requires more storage space than an incremental backup.

4.Snapshot backup: A snapshot backup takes a point-in-time snapshot of the system,

allowing you to back up files as they are at that moment without worrying about changes that occur during the backup process. T

his type of backup can be useful for large databases or other systems that are constantly changing.

5. Remote backup: A remote backup stores backup data on a separate system, typically over a network.

This type of backup provides an additional level of protection in case of system failure or disaster.

6. Cloud backup: Cloud backup stores backup data on a remote server, typically provided by a third-party service.

This type of backup can be convenient and cost-effective, but it may require a reliable and fast internet connection.

**41.How to compress and extract a file in Linux? [UPRMS 2022]**

To compress a file in Linux, you can use the gzip or bzip2 command. The following command will compress the file myfile.txt and create a compressed file called myfile.txt.gz:

gzip myfile.txt

To extract a compressed file in Linux, you can use the gunzip or bunzip2 command. The following command will extract the compressed file myfile.txt.gz and create the original file myfile.txt:

gunzip myfile.txt.gz

You can also use the tar command to compress and extract files. The tar command can create an archive of files, and then compress the archive with gzip or bzip2. The following command will create an archive of the files myfile1.txt, myfile2.txt, and myfile3.txt, and compress it with gzip:

tar -czvf myfiles.tar.gz myfile1.txt myfile2.txt myfile3.txt

To extract the archive, you can use the following command:

tar -xzvf myfiles.tar.gz

This will extract the files myfile1.txt, myfile2.txt, and myfile3.txt to the current directory.

**42.What is I-node number? What I-node no contain and doesn’t contain? [Turing Exams 2023]**

Inode number plays an important role in the file system of a Linux or Unix-based operating system,

providing a unique identifier for each file or directory and enabling efficient file system

operations such as file access, modification, and deletion.

An inode, or index node, in a Linux file system contains various pieces of metadata about a file or directory, including:

File type (regular file, directory, symbolic link, etc.)

Ownership information (user ID and group ID)

Permissions (read, write, execute)

Timestamps (creation, modification, access)

File size in bytes

Number of hard links to the file

Block addresses pointing to the file's data on the storage device

Other file attributes (e.g., extended attributes, ACLs)

However, there are some pieces of information that are not stored in the inode, such as the file name itself. The file name is stored in the directory entry that points to the inode. This allows multiple directory entries to refer to the same inode (i.e., hard links), which can be useful for creating aliases or sharing files among multiple users.

**43.What is Soft and Hard Link what is the difference?**

Hard links and soft links are two types of links that can be created in Linux to point to a file or directory.

Hard links are special files that point to the same data as another file. This means that any changes made to the data in the original file will be reflected in the hard link, and vice versa. Hard links are often used to create backups of files, or to make it easier to access files from different directories.

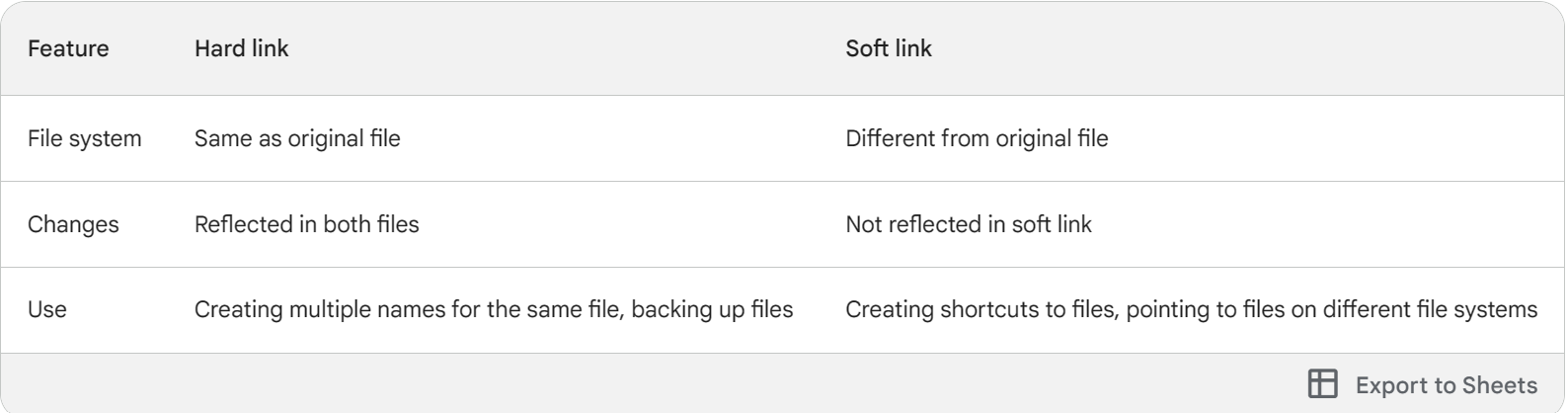
Soft links are also special files that point to another file, but they do not share the same data as the original file. Instead, soft links point to the path of the original file. This means that any changes made to the original file will not be reflected in the soft link, and vice versa. Soft links are often used to create shortcuts to files, or to make it easier to access files from different filesystems.

To create a hard link in Linux, you can use the ln command. The following command will create a hard link to the file myfile.txt called mylink.txt:

ln myfile.txt mylink.txt

To create a soft link in Linux, you can use the ln -s command. The following command will create a soft link to the file myfile.txt called mylink.txt:

ln -s myfile.txt mylink.txt



**44.Display file name in acending format?**

Here is the command to see the file size in ascending format in Linux, incorporating the safety guidelines:

du -sh \* | sort -Srn

The du command will calculate the size of each file in the current directory. The -sh option will display the size in human-readable format (e.g., 100M = 100 megabytes). The sort command will sort the output of the du command by size in ascending order. The -Srn option will sort by size, with the largest files first.

This command will print the size of each file in the current directory, sorted by size in ascending order. The first file will be the smallest file, and the last file will be the largest file.

**45. Where kernel files and grub files are stored in Linux? What is the use?**

GRUB files are typically located in the /boot/grub directory. This directory contains the GRUB configuration file, which is named grub.cfg. The /boot/grub directory may also contain other GRUB files, such as the GRUB bootloader code, which is located in the MBR of the boot device.

The kernel files are the core of the operating system and are responsible for managing the system's resources, such as memory, processors, and devices. The kernel is loaded by the boot loader (such as GRUB) and is the first program to run when the system boots up.

GRUB is a boot loader that is responsible for loading the kernel into memory and initializing the system. GRUB allows the user to choose which kernel to boot and provides a menu of boot options. It also allows the user to configure various boot parameters, such as the kernel command line parameters.

In summary, the kernel files are stored in the /boot directory and are responsible for managing the system's resources, while the GRUB files are stored in the /boot/grub directory and are responsible for loading the kernel and initializing the system.

**46.What is Virtualization? What is Hypervisor name some Hypervisor? What is BareMetal Hypervisor?**

Virtualitation is defined as combining and spliting of a hardware resource and then convert into logical form it is also defined as it converts hardware into software

Hypervisor creates virtual machines

Hypervisor is the main who create virtulization there are two type of Hypervisor

A hypervisor, also known as a virtual machine monitor (VMM), is a software layer that enables virtualization.

Type 1 (or bare metal) hypervisors: These hypervisors run directly on the host machine's hardware, without the need for an underlying operating system. They provide better performance and scalability than type 2 hypervisors, as they have direct access to the hardware resources. Examples of type 1 hypervisors include VMware ESXi, Microsoft Hyper-V, and Citrix XenServer.

Type 2 (or hosted) hypervisors: These hypervisors run on top of an existing operating system, and virtual machines run on top of the hypervisor. They are easier to set up and use than type 1 hypervisors, but they provide lower performance and scalability. Examples of type 2 hypervisors include Oracle VirtualBox, VMware Workstation, and Parallels Desktop.

Some popular hypervisors include:

* VMware ESXi
* Microsoft Hyper-V
* Citrix XenServer
* Oracle VirtualBox
* KVM (Kernel-based Virtual Machine)
* Xen Project

In summary, virtualization is the technology that enables multiple operating systems to run on a single physical machine, and a hypervisor is the software layer that enables virtualization. There are two types of hypervisors: type 1 (bare metal) and type 2 (hosted), with examples including VMware ESXi, Microsoft Hyper-V, and Oracle VirtualBox.

**47.How to see hidden files in Linux?How to create Hidden file**

- ls -lart

- touch .filename

**48.How to see the current shell and its PID in Linux?**

-echo $$ -to see pid or ps -p $$

-echo $0

**49.What is LVM in Linux?**

Brief summary of the steps and commands to create, reduce, and extend an LVM:

Creating an LVM:

Partition the disk using a partitioning tool, such as fdisk or parted.

**Create a Physical Volume (PV) using the pvcreate command, for example:**

pvcreate /dev/sdb1

**Create a Volume Group (VG) using the vgcreate command, for example:**

vgcreate myvg /dev/sdb1

**Create a Logical Volume (LV) using the lvcreate command, for example:**

Lvcreate -L 10G -n mylv myvg

**Format the LV with a filesystem using the appropriate command for the filesystem of your choice, for example:**

mkfs.ext4 /dev/myvg/mylv

**Mount the LV to a directory using the mount command, for example:**

mount /dev/myvg/mylv /mnt/mylv

**Reducing an LVM:**

**Unmount the LV using the umount command, for example:**

umount /mnt/mylv

**Shrink the filesystem using the appropriate command for the filesystem of your choice, for example:**

resize2fs /dev/myvg/mylv 5G

**Reduce the size of the LV using the lvreduce command, for example**:

lvreduce -L 5G /dev/myvg/mylv

**using the Mount the LV back to the directory mount command, for example:**

mount /dev/myvg/mylv /mnt/mylv

**Extending an LVM:**

Unmount the LV using the umount command, for example:

umount /mnt/mylv

**Extend the size of the LV using the lvextend command, for example:**

lvextend -L 15G /dev/myvg/mylv

**Expand the filesystem to fill the new space using the appropriate command for the filesystem of your choice, for example:**

resize2fs /dev/myvg/mylv

**Mount the LV back to the directory using the mount command, for example:**

mount /dev/myvg/mylv /mnt/mylv

**These are the general commands and steps to create, reduce, and extend an LVM. The specific commands may vary depending on the version of Linux and LVM tools you are using.**

**53 What are the prerequisite of lvreduce ? [ITC 2022]**

Backup Your Data: Always back up your data before making any changes.

Unmount the Filesystem: If the logical volume contains a mounted filesystem, unmount it:

umount /dev/vg\_name/lv\_name

**Resize the Filesystem: Resize the filesystem to a smaller size to match the desired reduced size of the logical volume. Run the e2fsck command to check the filesystem first, and then resize it:**

e2fsck -f /dev/vg\_name/lv\_name

resize2fs /dev/vg\_name/lv\_name <new\_size>

Replace <new\_size> with the desired size, e.g., "10G" for 10 gigabytes.

**Reduce the Logical Volume: Use the lvreduce command to reduce the size of the logical volume:**

lvreduce -L <new\_size> /dev/vg\_name/lv\_name

Replace <new\_size> with the desired size.

**Resize the Filesystem Again: Resize the filesystem to fully utilize the reduced logical volume size:**

resize2fs /dev/vg\_name/lv\_name

**Remount the Filesystem: If you unmounted the filesystem earlier, remount it:**

mount /dev/vg\_name/lv\_name /mount/point

Replace /mount/point with the actual mount point of your filesystem.

Verify and Test: Thoroughly test the system to ensure everything is working as expected. Check the filesystem's integrity and verify that the data is intact.

Thank you for bringing up the importance of the e2fsck command in this process. It's a crucial step to ensure the filesystem's health after resizing.

**50.what are the entries of /etc/fstab? [NAGOSA Projects]**

The /etc/fstab file is a system configuration file on Linux systems that contains information about filesystems that are automatically mounted during system startup. Each line in the file represents a separate filesystem mount point and consists of several fields separated by whitespace. Here are the fields:

**Filesystem**: This field specifies the device name or UUID of the filesystem to be mounted.

**Mount point**: This field specifies the directory where the filesystem will be mounted.

**Filesystem type**: This field specifies the type of filesystem, such as ext4, xfs, ntfs, etc.

**Mount options:** This field specifies the mount options to be used when mounting the filesystem, such as read-only, noauto, user, etc. The options are separated by commas and enclosed in parentheses.

**Dump frequency:** This field specifies whether or not the filesystem should be backed up using the dump utility. A value of 0 means the filesystem will not be backed up.

**Filesystem check order**: This field specifies the order in which the filesystem should be checked by the fsck utility during system startup. A value of 0 means the filesystem will not be checked.

**/dev/sda1 / ext4 defaults 1 1**

file system------mount point---------file system type------------mount point option------------dump frequency-----------filesystem check order

**58.What is SAMBA? What protocol SAMBA follows?**

Samba is a free, open-source implementation of the SMB/CIFS networking protocol used for sharing files, printers, and other resources between computers running different operating systems,

such as Linux, Windows, and macOS. The Samba server software runs on Linux/Unix servers and enables them to act as file and print servers for Windows clients.

Samba provides a way to share files and printers across a network using SMB/CIFS protocol

* SMB (Server Message Block): SMB is the original protocol developed by Microsoft for file and printer sharing in local area networks (LANs). It enables computers to communicate over a network and share resources such as files, directories, and devices like printers.
* CIFS (Common Internet File System): CIFS is an extension of the SMB protocol that adds enhancements to support sharing files and resources over the Internet. It includes improvements for security, data integrity, and remote access. CIFS was introduced as a more standardized version of SMB to enable cross-platform compatibility.

**58.What is RAID? Types of RAIDS? [Turing 2023]**

RAID stands for Redundant Array of Independent Disks. It is a technology used in computer storage systems to combine multiple physical hard disk drives (HDDs) or solid-state drives (SSDs) into a single logical unit to provide data redundancy, performance improvement, or both. RAID arrays are typically used in servers, network-attached storage (NAS) devices, and other high-performance computing environments.

The basic idea behind RAID is to improve data reliability and/or performance by spreading data across multiple drives, so that if one drive fails, the data can still be accessed from the remaining drives. There are several different RAID levels or configurations, each with its own benefits and trade-offs. Some of the commonly used RAID levels include:

**RAID 0:** Also known as "striping," RAID 0 does not provide any redundancy but offers improved performance by striping data across two or more drives. However, if one drive fails, all data in the RAID 0 array may be lost.

**RAID 1:** Also known as "mirroring," RAID 1 duplicates data across two or more drives to provide data redundancy. If one drive fails, the data can still be accessed from the remaining drives.

**RAID 5:** RAID 5 distributes data and parity (error correction) information across three or more drives, providing both data redundancy and improved performance. If one drive fails, the data can be reconstructed from the parity information and the remaining drives.

**RAID 6:** Similar to RAID 5, but with two parity drives instead of one, RAID 6 provides improved data redundancy, allowing for the failure of up to two drives without data loss.

**RAID 10:** Also known as "RAID 1+0," RAID 10 combines mirroring (RAID 1) and striping (RAID 0) to provide both data redundancy and improved performance. Data is mirrored across multiple pairs of drives, and then striped across the mirrored pairs.

These are just a few examples of the different RAID levels available. RAID technology has evolved over the years, and there are other RAID levels and configurations with varying levels of data redundancy, performance, and capacity. It's important to carefully consider the specific requirements and use case before choosing a RAID level for a particular storage system.

**59.How to Break root password in Linux?[TCS 2023]**

1. First, you need console access: Either at a keyboard and monitor locally, or via Virtual Machine remote console, you will need to see and interact with the bootloader.
2. Reboot the machine: As soon as the bootloader comes up with the selection screen, quickly tap the up and down arrows up and down to pause the countdown.
3. Select the kernel you want to boot into, and hit 'e': This will take you into a screen where you can edit the grub bootloader script.
4. Find the line that refers to the kernel: There will be a series of 'boot parameters' here: these are instructions passed during the loading of the kernel.
   1. For RHEL/CentOS 7, the line starts with 'linux16'.
   2. For RHEL/Centos 8x, and Fedora the line starts with 'linux'.
5. Add 'rd.break' at the end of that line (There are other things you can do here, but for now, this is all you need) **[ Note: This change is temporary ]**.
6. Now hit Ctrl-x to run the edited bootloader script.
7. You’ll boot to a 'rescue' prompt that looks like this: switch\_root:/#.
8. Remount the root partition in read-write mode so that you can run commands. Enter the following: mount -o remount rw /sysroot and then hit ENTER.
9. Now type chroot /sysroot and hit enter. This will change you into the sysroot (/) directory, and make that your path for executing commands.
10. Now you can simply change the password for root using the passwd command.
11. Next, before you reboot, you will need to make sure that SELinux allows the file changes. At the prompt ,enter: touch /.autorelabel. This will signal SELinux on the next reboot that the filesystem has changed (the changed password) and allow the change to be loaded. This will cause the whole filesystem to be 'relabeled' which might take a while, depending on the size of the filesystem and the speed of the machine, so be aware of this

**60.What is grep command? Example?**

Global Regular Expression Print

line grep -c - count

grep -i - specific string

grep -n - Line no

grep -l - search the files contain the following pattern

grep -c "apple" fruits.txt

This command will count and display the number of lines containing the word "apple" in the file fruits.txt.

grep -i: Case-Insensitive Search

The -i option is used to perform a case-insensitive search, which means the search will match regardless of the case of the letters.

Example:

grep -i "orange" fruits.txt

This command will search for the word "orange" in a case-insensitive manner in the file fruits.txt.

grep -n: Display Line Numbers

The -n option is used to display line numbers along with the matching lines.

Example:

grep -n "banana" fruits.txt

This command will display lines containing the word "banana" along with their line numbers in the file fruits.txt.

grep -l: List Files with Matching Pattern

The -l option is used to display only the names of the files that contain the specified pattern.

Example:

grep -l "grape" \*.txt

This command will search for the word "grape" in all .txt files in the current directory and display the names of the files that contain the pattern.

Remember to replace the file names and patterns in the examples with actual file names and patterns relevant to your system. These options provide different ways to search for and manipulate text in files using the grep command.

**61.What is AWK command with example? [Subham Training/Max interview]**

awk is a powerful text processing tool that provides a wide range of options and attributes to manipulate and transform text files. Here are some of the most commonly used options and attributes of awk:

Options:

* -F or --field-separator: specifies the field separator character(s) used in the input file.
* -v or --assign: assigns a value to a variable that can be used in the awk script.
* -f or --file: reads the awk script from a file instead of the command line.
* -i or --inplace: modifies the input file in place instead of writing to standard output.
* Attributes:
* NR: the current record number (line number).
* NF: the number of fields in the current record.
* $0: the entire current record.
* $1, $2, ...: the individual fields of the current record.
* FS: the field separator character(s) used in the input file.
* BEGIN: a special pattern that is executed before the first record is read.
* END: a special pattern that is executed after the last record is read.
* if statement: conditional statement that executes a block of code if a certain condition is true.
* for loop: iterates over a block of code for a certain number of times.
* while loop: iterates over a block of code while a certain condition is true.
* print: outputs a string or a variable to standard output.
* printf: outputs a formatted string or a variable to standard output.
* getline: reads the next record from a file or from standard input into a variable.
* split: splits a string into an array of substrings based on a delimiter.
* gsub: replaces all occurrences of a string in a variable or a field with another string.
* substr: extracts a substring from a string or a field.

These are just a few examples of the many options and attributes available in awk. You can refer to the awk manual or documentation for more details and examples.

**1.Print specific columns from a file:**

To print the first and third columns of a file, you can use the following command

$ awk '{print $1, $3}' file.txt

**2.To print all Line except 3rd and 6th Line from text.txt and then delete**

$ awk 'NR != 3 && NR != 6 {print $0}' file.txt | xargs rm -rf -exec

**3.To print all Line except 3rd and 6th Line from text.csv**

$ awk -F ',' 'NR!=3 && NR!=3 && NR!=6' text.csv

**4.I have a file which contain employee details like---- Name-----address--- phoneno ----salary -------parents name**

1.I want to get the employee name who got maximum salary

awk '{print $1, $4}' employee\_details.txt | sort -k2 -rn | head -n 1 | awk '{print $1}'

This command performs the following steps:

awk '{print $1, $4}' employee\_details.txt extracts the first and fourth fields (name and salary) from the employee\_details.txt file and prints them as space-separated values.

sort -k2 -rn sorts the output based on the second field (salary), in reverse numeric order (i.e., highest salary first).

head -n 1 selects only the first line of the sorted output (i.e., the employee with the highest salary).

awk '{print $1}' extracts the first field (name) from the selected line and prints it.

This will output the name of the employee with the highest salary in the employee\_details.txt file.

**62. Describe SED command in Linux?**

- The sed command in Linux (which stands for "stream editor") is a powerful utility for performing text transformations on input streams

**How to show only a given line or range of lines?**

-sed -n '1p' file\_name (first line)

-sed -n '1,5p' file\_name (first line to fifth line)

-sed -n '$p' file\_name (last line)

**How to see all the users from India Country?**

- sed -n '/India/p' file\_name p is print

**How to use multiple expression in sed command?**

Example: If you wanna only see 2 and 5th line

- sed -n -e '2p' -e '5p' file\_name e = expression

**How to see all the users from India and Germany?**

- sed -n -e '/India/p' -e '/Germany/p' file\_nam

**How to see next 4 lines from 2nd line?**

- sed -n ‘2,+4p’ file\_name

**How to see every 2nd line from first line?**

- sed -n ‘1~2p’ file\_nam

**How to replace a word in a file and show?**

- sed 's/<string\_to\_change>/<new\_string>/g' /file path file\_name s = substitute g = global

**How to change salary or country of a user (Paul)?**

- sed '/Paul/ s/25000/35000/g' file\_name 12:12:00

- sed '/Paul/ s/India/US/g' file\_name

**How to delete empty line?**

-sed '/^$/d' file\_name

**How to replace a word in a file and show except a given line or only in given line?**

- sed '5 s/<string\_to\_change>/<new\_string>/g' file\_name

- sed '5! s/<string\_to\_change>/<new\_string>/g' file\_name

**How to add new line after a given line no.?**

- sed '5 a new\_text' file\_name

**How to see the line number in file?**

- sed '=' file\_name

**What are wild cards? SED command wild cards?**

^ start of line

$ end of line

. single character

[] match character set

[^] exclusive Set

\* zero or more occurance

? only have to match single string

**How to find a 5 letter name which start with S and end with a?**

- sed -n '/^S...a$/p' names

**How to see names start with only A and C?**

- sed -n '/[AC]/p' names

- sed -n '/[AC]/p' names

**63. How to troubleshoot SSH service not working?**

When SSH (Secure Shell) is not working, there could be various reasons behind the issue. To troubleshoot SSH connectivity problems, you can follow these steps:

Check SSH Service Status:

On the server: Make sure the SSH server is running. You can use the command sudo service ssh status or systemctl status ssh (depending on your Linux distribution) to check the status.

On the client: Ensure that you have the ssh client software installed and accessible.

Check SSH Port:

By default, SSH uses port 22. Ensure that the server is listening on port 22 and your client is connecting to the correct port.

You can specify a different port using the -p flag: ssh user@hostname -p port.

Firewall and Security Groups:

On the server: Check your firewall settings to make sure that port 22 (or the custom SSH port) is open for incoming connections.

On the client: If you're connecting from a restricted network, ensure that outbound connections to port 22 (or the SSH port) are allowed.

Network Connectivity:

Check if there's a network connectivity issue between your client and the server. You can use tools like ping or traceroute to diagnose potential network problems.

Authentication and Credentials:

Double-check the username and hostname you're using for the SSH connection.

Ensure you're using the correct private key (if using key-based authentication) and that the corresponding public key is in the ~/.ssh/authorized\_keys file on the server.

SSH Logs:

Check the SSH logs on both the server and client. On the server, check **/var/log/auth.log** or /var/log/secure (location might vary by distribution). On the client, check /var/log/auth.log or similar.

IP Blocking or DenyHosts:

Some systems might block repeated failed SSH login attempts. Check if your IP is not blocked by tools like fail2ban or DenyHosts.

SSH Server Configuration:

Check the server's SSH configuration file (/etc/ssh/sshd\_config) for any misconfigurations. Make sure the necessary options are correctly set.

Try a Different Client:

If the problem persists, try using a different SSH client. This can help rule out client-specific issues.

Temporary Disabling Firewall:

As a troubleshooting step, you can temporarily disable firewalls on both the client and server sides to see if the problem is firewall-related.

Update SSH Software:

Ensure that both your SSH client and server software are up-to-date.

Consult Documentation:

Refer to the documentation of your Linux distribution and SSH software for specific troubleshooting steps.

Remember to proceed cautiously and avoid making major changes without understanding the potential impact. If you're not comfortable troubleshooting or the issue persists, consider seeking assistance from someone experienced or your system administrator.

**64. What is chronyd ? How to setup chronyd? [TCS 2023]**

Chronyd is a network time synchronization daemon for Linux systems that is used to maintain accurate and synchronized time across computers by synchronizing with reliable time sources over the network. It's designed to provide a more flexible and efficient approach to time synchronization compared to the older NTP (Network Time Protocol) daemon, ntpd.

Here's how to use chronyd to sync time along with an example:

Installing Chronyd:

Install Chronyd: First, make sure chronyd is installed on your system. You can typically install it using your distribution's package manager. For example, on Ubuntu/Debian, you can use:

sudo apt-get install chrony

Configuring Chronyd:

Edit Configuration File: Open the configuration file /etc/chrony/chrony.conf with a text editor. You might need administrative privileges:

sudo nano /etc/chrony/chrony.conf

Specify Time Sources: In the configuration file, you'll see a list of time sources. These can be NTP servers that chronyd will synchronize with. Edit or add server lines like this:

server time.example.com iburst

Replace time.example.com with the hostname or IP address of an NTP server. The iburst option indicates that chronyd should attempt to quickly synchronize the clock.

Save and Close: After editing the configuration file, save your changes and exit the text editor.

Syncing Time:

Start Chronyd: Start the chronyd service:

sudo systemctl start chronyd

Monitoring Time Sync:

Tracking Information: To check the synchronization status and system time offset, use the chronyc tracking command:

chronyc tracking

Sources Information: To see a list of time sources being used by chronyd, use the chronyc sources command:

chronyc sources

Force Synchronization: If you want to force an immediate synchronization, you can use the chronyc makestep command:

sudo chronyc makestep

Enable Chronyd at Startup: To make sure chronyd starts automatically on system boot:

sudo systemctl enable chronyd

Chronyd will now work in the background to keep your system's clock accurately synchronized with the chosen time sources. It dynamically adjusts the time sources it uses based on their reliability and performance, ensuring that your system's time remainsaccurate.

**65.What is the difference between Red-hat 7 and Red-hat 8 systems? [TCS 2023]**

Red Hat Enterprise Linux (RHEL) 7 and RHEL 8 are two major releases of the Red Hat Enterprise Linux operating system. Each release introduces various changes, improvements, and new features. Here are some key differences between RHEL 7 and RHEL 8:

**Systemd as Default Init System:**

RHEL 7 uses the traditional init system (init or SysVinit) for system initialization and management.

RHEL 8 switches to systemd as the default init system. Systemd provides improved performance, parallelization, and enhanced service management.

**Application Streams:**

RHEL 7 uses a monolithic package repository structure, where software packages are tightly coupled with the operating system release.

RHEL 8 introduces the concept of "AppStreams," which allows decoupling of user space applications from the base operating system. This enables more flexibility in updating applications independently of the OS.

**Package Management:**

RHEL 7 uses the yum package manager for software installation and updates.

RHEL 8 transitions to dnf as the default package manager. dnf offers improved performance and better dependency resolution.

**Cockpit Management Console:**

RHEL 7 provides limited web-based system management capabilities.

RHEL 8 introduces Cockpit, a user-friendly web-based interface for managing system tasks, monitoring, and administrative tasks.

**Containerization and Docker:**

RHEL 7 includes support for Docker containers.

RHEL 8 extends container support with Red Hat Universal Base Image (UBI) and introduces Podman, a tool for managing containers and pods.

**Security Features:**

Both RHEL 7 and RHEL 8 focus on security enhancements, but RHEL 8 introduces features like Trusted Platform Module (TPM) 2.0 support, OpenSSL 1.1.1, and a more secure default configuration.

**SELinux Improvements:**

RHEL 8 continues to enhance SELinux security features for more granular access control and improved system security.

Filesystem and Storage:

RHEL 8 introduces Stratis, a new storage management solution that simplifies filesystem and volume management.

Networking and NetworkManager:

RHEL 8 enhances network management through NetworkManager improvements, including better integration with IPv6.

Performance and Scalability:

RHEL 8 brings performance and scalability improvements, particularly for cloud and virtualization environments.

Predictable Release Cycle:

RHEL 7 follows the traditional release model.

RHEL 8 introduces a more predictable release cycle, with major releases expected every three years.

Both RHEL 7 and RHEL 8 are enterprise-grade operating systems that offer stability, security, and a range of features to cater to different use cases. The choice between the two depends on factors such as specific requirements, compatibility with existing applications, and the desire for new features and improvements.

**66.What is stdin stdout stderr? How redirect both stdout and stderr to a file?**

stdin (Standard Input): This is where a command reads its input data. By default, it reads from the keyboard.

stdout (Standard Output): This is where a command sends its normal output. By default, it displays on the terminal.

stderr (Standard Error): This is where a command sends its error messages and diagnostic information. It's separated from stdout so that errors can be distinguished from regular output.

To redirect stderr and stdout in Linux, you can use the redirection operators and the file descriptors associated with these streams:

* >: Redirects stdout to a file. If the file already exists, it's overwritten.
* >>: Appends stdout to a file.
* 2>: Redirects stderr to a file. If the file already exists, it's overwritten.
* 2>>: Appends stderr to a file.
* &> or >&: Redirects both stdout and stderr to a file.
* &>>: Appends both stdout and stderr to a file.

Here are some examples of how to use these redirection operators:

Redirect stdout to a file:

ls -l /etc > listing.txt

This runs the ls command and writes the listing of /etc directory to the file listing.txt.

Append stdout to a file:

echo "Hello, World!" >> greetings.txt

This appends the "Hello, World!" message to the file greetings.txt.

Redirect stderr to a file:

command\_does\_not\_exist 2> error.log

This redirects the error message produced by the non-existent command to the file error.log.

Redirect both stdout and stderr to a file:

ls -l /etc &> output.txt

This redirects both the output and any error messages from the ls command to the file output.txt.

Remember that when you redirect stderr or stdout, you're sending them to a file instead of the terminal. This is useful for capturing output, analyzing errors, and storing logs. Additionally, you can use these redirection techniques in combination with pipes (|) to further process data or filter errors.

**67.How to secure a ssh connection?**

Here are some tips on how to secure a SSH connection:

* Use strong passwords. Passwords should be at least 12 characters long and contain a mix of uppercase and lowercase letters, numbers, and symbols.
* Disable root logins. Root is the most powerful user account on a system, so it is important to disable root logins over SSH. This will prevent attackers from gaining full control of your system if they are able to compromise your password.
* Change the default SSH port. The default SSH port is 22, which is well-known to attackers. By changing the port, you can make it more difficult for attackers to find your SSH server.
* Use SSH keys instead of passwords. SSH keys are a more secure way to authenticate to a remote server. They are a pair of files, a public key and a private key. The public key is placed on the remote server, and the private key is kept on your local computer. When you connect to the remote server, the SSH client will use the private key to authenticate you.
* Limit login attempts. You can limit the number of login attempts that a user can make before their account is locked. This will help to prevent brute-force attacks.
* Keep your SSH software up to date. Software updates often include security patches that can help to protect your SSH server from known vulnerabilities.
* Use a firewall to restrict access to your SSH server. A firewall can be used to block connections from unauthorized IP addresses.

You can also use a third-party SSH security tool to help you secure your SSH connection. These tools can help you to monitor your SSH server for suspicious activity, and they can also help you to enforce security policies.

**68.How to allow a particular user to connect with ssh?**

To allow a particular user in SSH and group, you can use the AllowUsers and AllowGroups directives in the SSH server configuration file.

The AllowUsers directive specifies the list of users who are allowed to SSH into the server. The AllowGroups directive specifies the list of groups of users who are allowed to SSH into the server.

To allow a particular user and group, you can add the username of the user to the AllowUsers directive and the name of the group to the AllowGroups directive.

For example, to allow the user johndoe and all members of the group admin to SSH into the server, you would add the following lines to the configuration file:

AllowUsers johndoe

AllowGroups admin

Save the configuration file and restart the SSH server for the changes to take effect.

Here are the steps in detail:

Open the SSH server configuration file with a text editor.

Find the AllowUsers and AllowGroups directives.

Add the username of the user and the name of the group to the appropriate directive.

Save the configuration file.

Restart the SSH server.

To restart the SSH server, you can use the following command:

sudo service ssh restart

Once you have restarted the SSH server, the user and group you specified will be able to SSH into the server.

Here are some additional things to keep in mind:

You can also use the DenyUsers and DenyGroups directives to specify the list of users and groups who are not allowed to SSH into the server.

The AllowUsers and DenyUsers directives are processed in the following order:

* DenyUsers
* AllowUsers
* DenyGroups
* AllowGroups

If a user or group is listed in both the AllowUsers and DenyUsers or AllowGroups and DenyGroups directives, the DenyUsers or DenyGroups directive takes precedence.

**69.Types of Files in Linux?**

In Linux, files can be categorized based on their purpose and content. Here are some common types of files you'll encounter:

Regular Files (Text and Binary):

Text Files: Contain human-readable text and can be edited using text editors.

Binary Files: Contain non-textual data (executables, images, videos, etc.) and require specialized tools to interpret.

Directories (Folders):

Containers for grouping and organizing files and subdirectories.

Symbolic Links (Symlinks):

Special files that point to another file or directory, acting as shortcuts.

Device Files:

Represent physical or virtual devices on the system.

Character Devices: Transfer data as a stream of characters (e.g., serial ports).

Block Devices: Transfer data in blocks (e.g., hard drives, SSDs).

Special Files:

Used for specific purposes, such as /dev/null (discards data), /dev/random (provides random data), etc.

FIFOs (Named Pipes):

Enable inter-process communication by allowing data to be passed between processes.

Sockets:

Used for inter-process communication, especially over a network.

Configuration Files:

Used to configure system or application settings.

Often found in /etc directory.

Executable Files:

Programs that can be run directly from the command line or by other programs.

Compressed and Archive Files:

Files that contain one or more other files, often used for backup or distribution purposes.

Examples: .zip, .tar, .gz, .tar.gz, .tar.bz2, etc.

Text Files with Special Formats:

Files with structured content like JSON, XML, CSV, etc.

System Files:

Files that are essential for the operating system's functioning.

Located in various system directories.

Log Files:

Records system and application activities, helpful for troubleshooting and monitoring.

Temporary Files:

Created and used temporarily by applications, often stored in /tmp or /var/tmp.

These categories cover the majority of file types you'll encounter in a Linux system. Each type serves a specific purpose and plays a role in the overall functionality of the system.

**70. When an application is installed on a Linux system, several parameters and artifacts are often created or modified to ensure the proper functioning of the application What are they?**

When an application is installed on a Linux system, several parameters and artifacts are often created or modified to ensure the proper functioning of the application within the system environment. These parameters and artifacts can include:

**Configuration Files:** Applications typically come with configuration files that determine how the application behaves. These files can be located in various directories, such as /etc, /usr/local/etc, or within the application's installation directory. Configuration files may control settings related to behavior, paths, logging, and more.

**Binary Executables:** The main executable files of the application, also known as binaries, are installed in directories like /usr/bin, /usr/local/bin, or /opt/application/bin. These binaries are what you execute to run the application.

**Library Files: If** the application relies on shared libraries, these libraries are often installed in directories like /usr/lib, /usr/local/lib, or /opt/application/lib. These files contain code that the application uses but is not bundled within its executable.

**Documentation:** Documentation files, including manuals, help files, or READMEs, may be installed to provide users with guidance on how to use the application effectively. These files might be placed in locations such as /usr/share/doc or within the application's installation directory.

**Data Files:** Applications may require data files for various purposes, such as templates, images, language packs, or sample content. These data files can be placed in directories like /usr/share/application or /opt/application/share.

**User and Group Accounts:** Some applications create their own user and group accounts during installation to ensure that the application runs with the appropriate privileges and isolation.

**Log Files:** Log files are often created in directories like /var/log to record the application's activities and events. These logs can be useful for debugging and monitoring.

**Temporary Files and Directories:** Applications might create temporary files and directories for their runtime needs. These are often located in the system's default temporary directory, such as /tmp.

**Startup Scripts:** System services or daemons associated with the application may have startup scripts placed in directories like /etc/init.d, /etc/systemd/system, or /etc/rc.d/init.d. These scripts control the application's automatic startup and shutdown behavior.

Symbolic Links: Symbolic links might be created in common directories like /usr/bin to point to the actual binary executable of the application. This allows users to run the application by typing its name without specifying the full path.

**Desktop Integration:** If the application has a graphical interface, it might create desktop shortcuts or entries in application menus for easy access.

**System Environment Variables:** The installation process might modify or create system-wide environment variables to allow the application to function properly. These variables could be added to configuration files like /etc/profile, /etc/environment, or scripts in /etc/profile.d.

**71.What to do if ping command not available?**

We can use traceroute command -> traceroute [www.example.com](http://www.example.com)

**72.What is Pacemaker in a Cluster define? Also define Jboss and WebLogic’s Clustering features [TCS 2022]**

In the context of computer clusters and high availability systems, "Pacemaker" refers to a resource management and high availability framework. Pacemaker is an open-source software that helps manage and control resources, services, and applications in a cluster environment to ensure continuous availability and fault tolerance.

A computer cluster consists of multiple interconnected computers or servers that work together as a single system to improve performance, reliability, and scalability. Clusters are often used for mission-critical applications where downtime is not acceptable.

Pacemaker provides the following key functionalities in a clustered environment:

**Resource Management:** Pacemaker manages resources, which can include services, applications, IP addresses, storage devices, virtual machines, and more. It monitors the health and status of these resources, ensuring that they are running and available as intended.

**Failover and High Availability:** Pacemaker monitors the health of resources and can automatically move resources from a failed node (server) to a healthy node in the cluster. This process is known as failover. The goal is to minimize downtime and ensure that services remain available even if a node fails.

**Load Balancing:** Pacemaker can distribute resources across multiple nodes in the cluster to balance the workload and prevent overloading of individual nodes. This helps optimize resource utilization and performance.

**Cluster Management:** Pacemaker provides tools for configuring, managing, and monitoring the cluster and its resources. Administrators can define rules, policies, and constraints that govern resource behavior and failover behavior.

**Complex Resource Dependencies:** Pacemaker supports complex resource dependencies and ordering rules. This allows administrators to define relationships between resources and ensure that they start, stop, or move in a specific sequence.

**Integration with Other Cluster Software:** Pacemaker can work alongside other cluster management software, such as Corosync or Heartbeat, to provide a comprehensive solution for high availability and resource management.

Pacemaker uses a concept called "resource agents" to manage various types of resources. Resource agents are scripts or executables that define how to start, stop, monitor, and manage specific resources. These agents provide the necessary logic to integrate resources into the Pacemaker framework.

Overall, Pacemaker plays a critical role in building and maintaining highly available systems by orchestrating resource management, failover, and load balancing in a clustered environment. It's widely used in enterprise settings to ensure uninterrupted availability of services and applications.

Yes, both Pacemaker and the clustering features of middleware applications like JBoss (JBoss EAP) offer similar high availability and fault tolerance capabilities, albeit with different implementations and integration approaches. They are used to achieve similar goals in ensuring that services and applications remain available and responsive even in the face of hardware or software failures.

Here's a comparison of the two:

**Pacemaker:**

* General Purpose Framework: Pacemaker is a general-purpose cluster resource manager and high availability framework that can be used with a variety of applications and services.
* Resource Management: Pacemaker can manage a wide range of resources, not limited to specific middleware. This includes IP addresses, services, applications, and more.
* External to Middleware: While Pacemaker can be used with middleware applications, it is a separate framework that operates at the system level, managing resources and failover across multiple applications and services.
* Integration: Middleware applications need to be integrated with Pacemaker using custom resource agents that define how Pacemaker interacts with the application.
* Complexity: Setting up and configuring Pacemaker can be complex, and it requires knowledge of the framework itself as well as the specific middleware applications being used.

**Middleware Clustering Features (e.g., JBoss EAP):**

* Specific to Middleware: Middleware applications that offer clustering features, such as JBoss EAP, provide built-in support for clustering and high availability specific to those applications.
* Integrated Solution: Middleware clustering features are integrated into the application itself, making them easier to set up and manage within the context of the application.
* Resource Scope: Clustering features in middleware typically focus on the resources managed by that middleware (e.g., application instances, services, databases).
* Simplified Integration: Middleware clustering features are typically integrated and configured using application-specific configuration files and settings.
* Simpler Setup: While still requiring proper configuration and planning, using built-in clustering features of middleware can be simpler and more straightforward compared to setting up an external framework like Pacemaker.
* In summary, both Pacemaker and middleware clustering features aim to achieve high availability and fault tolerance, but they do so using different approaches. Pacemaker is a general-purpose framework that can be used with various applications, while middleware clustering features are tailored to specific middleware applications, offering simpler integration and configuration for those applications. Your choice between the two might depend on factors such as your familiarity with the tools, your specific application requirements, and the level of control and customization needed.

**73 .How to make password less authentication between two servers?**