Assignment-2 Linux

Format:Lab Session
Time:90 mins

Instruction:
1.Complete the assignment and also upload the solutions in your Git repo
2.Take screenshot of the solution and paste it in word document
3.Convert the word doc into pdf before uploading.

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

In the Linux FHS (Filesystem Hierarchy Standard), "/" refers to the root directory of the file system. This is the topmost directory in the file hierarchy, below which all other directories and files in the file system reside.

2.What is stored in each of the following paths?
/bin, /sbin, /usr/bin and /usr/sbin
/etc
/home
/var
/tmp

- /bin: This directory contains important executables required for the proper functioning of the system. Basic system utilities such as ls, cp, mv, cat.

- /sbin: This directory contains important system administration binaries that are typically used by system administrators rather than regular users.

- /usr/bin: This directory contains executable programs installed by the system administrator or users. It usually contains most user-level binaries such as text editors, compilers, and graphical applications.

- /usr/sbin: This directory contains non-essential system administration binaries. It contains various system management tools such as B. Daemons and system utilities not required to boot the system.

- /etc: This directory contains system configuration files used to set up and configure the system and its applications.

- /home: This directory contains the user's home his directory. All users in the system have a subdirectory with their user name here where they can store their personal files and settings.
- /var: This directory contains variable data files such as system log files, mail spools, and print spools. These files are expected to grow and change over time.

- /tmp: This directory is used to store temporary files created by the system or its applications. These files are usually automatically deleted on system restart or after a period of time.

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

One reason to use a volatile file system such as tmpfs for the /tmp directory is to improve system performance. This is because accessing files in memory is usually faster than accessing files on disk. Additionally, deleting the /tmp directory regularly helps the system free up disk space and keep it from cluttering up with unnecessary files.

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

System, process, network, kernel information can be accessed

5.What makes /proc different from other filesystems?

Virtual filesystem, Dynamic content, Non-persistent, No disk space usage, Process information

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

False

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

Boot parameters, Kernel modules, Debugging options, Security options

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

In Linux, the system devices, including block storage devices, can be found in the /dev directory

Permissions

9.How to change the permissions of a file?

The chmod command takes a numeric or symbolic mode argument to specify the new permissions. The numeric mode argument uses a three-digit octal value, where each digit represents the permissions for a different user or group: The first digit represents the permissions for the owner of the file. The second digit represents the permissions for users in the file's group. The third digit represents the permissions for all other users. Each digit is a sum of the following values: 4: read permission 2: write permission 1: execute permission

10.What does the following permissions mean?:

777
644
750

- 777: This means that the file can be read, written, and executed by the file owner, group owner, and all other users on the system. This is the most permissive permission setting and should be used with caution.
- 644: This means that the file can be read and written by the file owner, but only read by the file's group and all other users on the system. This is a common permission setting for files that should not be modified by anyone except the file owner.
- 750: This means that the file can be read, written, and executed by the file owner, can be read and executed by the file's group, but can only be accessed by all other users on the system. This is a common permission setting for files that should be accessed by a specific group of users but not by others on the system.

11.What this command does? chmod +x some_file

the chmod +x some_file command makes the file "some_file" executable by all users on the system. This can be useful for running shell scripts or other executable files.

12.Explain what is setgid and setuid

Setgid (set group ID) is a special permission that can be set on directories to ensure that files and directories created within the directory inherit the group ownership of the parent directory. When the setgid bit is set on a directory, any new file or directory created within it will inherit the same group ownership as the parent directory, rather than the group ownership of the user who created the file or directory. This can be useful for sharing files between users who belong to the same group, as it ensures that all files created within the directory are owned by the same group. Setuid (set user ID) is a special permission that can be set on executable files to ensure that they are executed with the privileges of the owner of the file, rather than the privileges of the user who executed the file. When the setuid bit is set on an executable file, any user who executes the file will run it with the permissions of the file owner, rather than their own permissions. This can be useful for running programs that require elevated privileges, such as system maintenance utilities or installers.

13.What is the purpose of sticky bit?

sticky bit is a special permission that can be set on directories to control who can delete or modify files within the directory. When the sticky bit is set on a directory, it ensures that only the owner of a file or directory, or the root user, can delete or rename the file or directory within that directory. The purpose of the sticky bit is to prevent accidental deletion or modification of important files by other users on the system. For example, the sticky bit might be set on a directory used by multiple users to store files that are shared between them. By setting the sticky bit on this directory, the owner of a file can be sure that no other user will accidentally delete or modify the file.

14. What the following commands do?
chmod
chown
Chgrp

The following commands are used for managing file permissions and ownership in Linux:

1. chmod: Short for "change mode", chmod is used to change the permissions of a file or directory. It allows the user to add or remove read, write, and execute permissions for the owner, group, and others. The command is typically used in combination with octal or symbolic permissions to specify the new permissions for the file or directory.
2. chown: Short for "change owner", chown is used to change the owner of a file or directory. It allows the user to specify a new user or group as the owner of the file or directory. The command is typically used in combination with the -R option to recursively change the ownership of all files and directories within a directory.
3. chgrp: Short for "change group", chgrp is used to change the group ownership of a file or directory. It allows the user to specify a new group as the owner of the file or directory. The command is typically used in combination with the -R option to recursively change the group ownership of all files and directories within a directory.

Overall, these commands are essential for managing file permissions and ownership in Linux, and are often used by system administrators and users to maintain system security and control access to files and directories

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

Sudo is a command-line utility in Linux that allows users to run commands with administrative or root privileges without logging in as the root user. This is an important tool for managing system security and control access to sensitive files and commands. To set up sudo, you need to install it using a package manager like apt-get and add the user to the sudoers file, which is located at "/etc/sudoers". This file controls which users can use sudo and what commands they can execute with it. Once sudo is set up, users can run commands with elevated privileges by typing "sudo" before the command. For example, "sudo apt-get install <package_name>" would install software with elevated privileges. Overall, sudo is an essential tool in Linux for managing system security and ensuring that users have access to the tools they need to perform their tasks without compromising system stability.

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

True. In order to install packages on the system, one must either be logged in as the root user or use the sudocommand to execute the installation command with superuser privileges

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

ACLs, or Access Control Lists, are a way of defining more granular permissions for files and directories in Linux. They allow for permissions to be set

for specific users and groups on a file or directory, while still preserving the traditional read, write, and execute permissions for the owner, group, and others. This can be helpful in scenarios where you need to grant or restrict access to specific files or directories for certain users or groups, without affecting the permissions of others. ACLs can add complexity to the permission system, so it's important to use them judiciously and only in cases where they're really needed.

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

There could be several reasons why creating a file could fail. Here are three possible reasons:

1. Permission issues: The user attempting to create the file may not have the necessary permissions to create files in the directory or location they are attempting to create it in. This could be due to file permissions, ownership issues, or access control lists (ACLs) that restrict access.
2. Disk space issues: If there is not enough free space on the disk, attempts to create a file may fail. This can happen if the disk is full or if quotas are in place that limit the amount of space available to a particular user or group.
3. File already exists: If a file with the same name already exists in the directory, attempts to create a new file with the same name may fail. In this case, the user may need to choose a different name or delete the existing file first.

There may be other reasons for file creation failures as well, such as disk errors or file system corruption, but the above reasons are some of the most common.

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

When a user accidentally executes "chmod -x $(which chmod)" command, it removes the execute permission from the chmod binary file, making it impossible to execute any chmod command in the future. This is a serious issue because chmod is an essential tool for managing file permissions on Linux systems. To fix this issue, one can reboot the system into a single-user mode or use a live CD to access the root file system in read-write mode. Then, navigate to the directory where the chmod binary is located and grant execute permissions to the chmod binary using the command "chmod +x chmod". This will restore the execute permission to the chmod binary and allow users to execute chmod commands again.

Alternatively, if there is another user with sudo privileges, they can use the "sudo chmod +x /bin/chmod" command to grant execute permissions to the chmod binary file. It is important to be careful when modifying file permissions as it can have

unintended consequences on the system. In general, it is recommended to create a backup of important system files before making any changes to file permissions or ownership.

Scenarios

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

There are several ways to copy a file to a remote Linux host. One common way is to use the scp(secure copy) command, which uses SSH to securely copyfiles between hosts. Here's an example command to copy a file called file.txtfrom the local host to a remote host at IP address 192.0.2.1:scp file.txt user@192.0.2.1:/path/to/destinationIn this command, useris the username you use to log into the remote host, and /path/to/destinationis the path on the remote host where you want to copy the file to. You will be prompted to enter the password for the remote user account to complete the copy operation.21.

21.How to generate a random string?

In Linux, you can use the openssl command to generate a random string. The openssl rand command can be used to generate random data. By default, openssl rand generates binary data, but you can use the -base64 option to encode the data in base64 format, which produces a string that is easier to read and use. To generate a random string of a specific length, you can use the -base64 and -rand options together with the head command to extract the desired number of characters. For example, the following command generates a random string of length 10:openssl rand -base64 15 | head -c 10 ; echo
The head -c 10 command extracts the first 10 characters of the output of openssl rand -base64 15. The echo command adds a newline character to the end of the output, to make the output look cleaner. You can adjust the length of the output string by changing the value passed to head -c.

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

In Linux, you can use the openssl rand command to generate a random string of a specified length. To generate a random string of 7 characters, you can use the following command:
openssl rand -base64 4 | cut -c1-7
This command generates 4 random bytes in Base64 format, then selects the first 7 characters using the cut command. The resulting output is a random string of 7 characters.
You can also use other methods to generate random strings in Linux, such as using the head command to select random lines from a dictionary file or using the shuf command to shuffle and select random characters from a set of characters.

Systemd

23. What is systemd?

Systemd is a system and service manager for Linux operating systems. It is responsible for controlling the startup and shutdown processes of the system, managing system services and daemons, and providing advanced logging and monitoring capabilities.

24. How to start or stop a service?

To start a service in Systemd,
sudo systemctl start <service-name>
To stop a service in Systemd,
sudo systemctl stop <service-name>

25. How to check the status of a service?

sudo service <service-name> status

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

sudo journalctl -u <service-name>

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

To make a process or application a service in Linux, you need to create a systemd unit file that describes the service and its configuration options. The unit file should be created in the **/etc/systemd/system/** directory with a name that ends with the **.service** extension. The unit file should define the service properties in the **[Unit]** section, the executable that starts the service in the **[Service]** section, and the dependencies and installation options in the **[Install]** section. After creating the unit file, you can start, stop, enable, and disable the service using the **systemctl** command. When writing a unit file, it's important to be clear and concise while providing all the necessary information to ensure that the service is properly managed by systemd.

28. Troubleshooting and Debugging

Troubleshooting and debugging are essential skills for Linux system administrators. To troubleshoot and debug issues on a Linux system, you should read the system logs, check the system resources, test network connectivity, use tools such as **strace** and **gdb**, check file permissions, and check for software updates. By being systematic and thorough in your approach and documenting your steps and findings, you can identify and resolve issues quickly and effectively, minimizing downtime and maximizing system performance.

29. Where system logs are located?

System logs in Linux are typically located in the **/var/log/** directory. This directory contains log files generated by various system services and applications, such as the syslog, kernel, Apache web server, and others. The log files are usually named according to the service or application that generates them, and they contain information about events, errors, and activities that occur on the system.

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

To follow a file's content as it is being appended without opening the file every time, you can use the **tail** command with the **-f** (follow) option. The **tail** command displays the last few lines of a file, and with the **-f** option, it will continue to display any new lines that are appended to the file in real-time, you can use the following command:

tail -f /var/log/syslog

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

Some commonly used tools for troubleshooting and debugging network issues in Linux are ping, traceroute, netstat, tcpdump, wireshark, and dig.

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

Some commonly used tools for troubleshooting and debugging disk and file system issues in Linux are fsck, smartctl, lsof, mount, and df.

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

Some commonly used tools for troubleshooting and debugging process issues in Linux are ps, top, kill, strace, and lsof.

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

Some commonly used tools for debugging CPU related issues in Linux are top, htop, mpstat, iostat, perf, and sar.

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

- Verify that the system is indeed slow by checking system resource utilization, using tools like top, htop, or sar.

- Check the system logs for any error messages or warnings that could indicate a problem.
- Check the network connectivity using tools like ping, traceroute, or netstat, in case the issue is network-related.
- Check disk and file system usage and performance using tools like df, du, or iostat.
- Check running processes and services using tools like ps or systemctl, to identify any processes that are using excessive resources.
- If the issue is related to a specific application, try restarting the application or checking its logs for errors.
- If all else fails, consider hardware troubleshooting and diagnostics.

36.Explain iostat output

The output of iostat includes the following columns:

- Device: The name of the device or partition being monitored.
- tps: The number of I/O transactions per second that have been issued to the device.
- kB_read/s: The amount of data in kilobytes that has been read from the device per second.
- kB_wrtn/s: The amount of data in kilobytes that has been written to the device per second.
- kB_read: The total amount of data in kilobytes that has been read from the device since the system was started.
- kB_wrtn: The total amount of data in kilobytes that has been written to the device since the system was started.
- %util: The percentage of time that the device is busy handling I/O requests.

37.How to debug binaries?

- Using gdb: gdb is a command-line tool that can be used to debug binaries. It allows you to step through the code, set breakpoints, examine variables and memory, and more.
- Using strace: strace is a tool that can be used to trace system calls and signals made by a process. It can be used to identify issues such as file access errors, network connectivity issues, and more.

- Using ltrace: ltrace is a tool that can be used to trace library calls made by a process. It can be used to identify issues related to shared libraries or missing dependencies.
- Using valgrind: valgrind is a tool that can be used to detect memory leaks and other memory-related issues in a binary.

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

- CPU load refers to the number of processes that are waiting to be executed or waiting for I/O resources. It is typically expressed as a decimal value, with a load of 1.0 indicating that the system is fully utilized.
- CPU utilization, on the other hand, refers to the percentage of time that the CPU is busy handling tasks. It is typically expressed as a percentage value.

39.How you measure time execution of a program?

- Using the time command: The time command can be used to measure the execution time of a program. Simply precede the command with the time command, and it will display the total execution time, as well as CPU usage and other statistics.
- Using the date command: The date command can be used to measure the execution time of a program by capturing the system time before and after the program execution and calculating the difference.
- Using profiling tools: Profiling tools such as perf, gprof, or Valgrind can be used to measure the execution time of a program and identify performance bottlenecks.

Scenarios

40.You have a process writing to a file. You don't know which process exactly, you just know the path of the file. You would like to kill the process as it's no longer needed. How would you achieve it?

To kill a process that is writing to a file, you can follow these steps in Linux: Identify the process ID (PID) of the process writing to the file using the lsof command. For example: lsof /path/to/file Use the PID to send a signal to the process using the kill command. For example: kill PID If the process does not respond to the kill command, you can use the kill -9 command to force the process to terminate. For example: kill -9 PID

Kernal

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

The main function of the kernel is to act as an intermediary between the hardware and the software, managing hardware resources and providing services to software applications. The kernel also handles tasks such as process management, memory management, device drivers, file system management, and security.

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

uname -r command

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

A kernel module can be used to add support for new hardware, file systems, or network protocols, or to provide new system services and functions.

To load a new kernel module, follow these steps:

1. Check if the module is already installed: Run the command lsmod to list all the currently loaded modules on your system. You can check if the module you want to load is already present.
2. Install the module if needed: If the module is not installed, you need to install it using the package manager for your Linux distribution.
3. Load the module: Use the modprobe command to load the module into the kernel. For example, to load the nvidia module.
4 . Verify the module is loaded: Run the lsmod command again to verify that the module has been loaded.

44.Explain user space vs. kernel space

Kernel Space User Space

Kernels and OS core execute here. Normal program and applications softwares run here.

Its the core space of OS. It's a form of sand-boxing that restricts user processes to access OS kernel.

It has full access to all memory and machine hardware. It has limited access to memory and access kernel through system calls only.

It contains the page table for process, kernel data structure, threads, and kernel code etc. It contains the program code, data, stacks, and heap of the process.

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

During the compilation phase, you can customize the kernel configuration by selecting which features and modules to include or exclude. This is typically done using a configuration tool such as make menuconfig, make xconfig, or make defconfig. These tools allow you to enable or disable various kernel features, drivers, and subsystems, and to fine-tune the kernel's behavior and performance.

46.Where can you find kernel's configuration?

the kernel configuration is stored in a file called .config in the root directory of the kernel source tree. This file contains a list of configuration options and settings that were selected during the kernel compilation process.

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

On Linux, boot loader configuration files are typically located in the /boot directory and named grub.cfg or menu.lst. This file contains boot loader configuration information, such as command line options and parameters passed to the kernel at boot time.

48.How to list kernel's runtime parameters?

sysctl -a

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

Running sysctl -a as a normal user will only show kernel parameters that are readable by non-privileged users. Many kernel parameters require root privileges to read or modify. So running sysctl -a as a regular user may not show all parameters or their current values.

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

To enable IPv4 forwarding in the kernel, you can use the sysctl command to modify the net.ipv4.ip_forward parameter.

Here are the steps to enable IPv4 forwarding in the kernel:

Open a terminal or command prompt as root or with sudo privileges.

Check the current value of the net.ipv4.ip_forward parameter by running sysctl net.ipv4.ip_forward

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

Check if the value of the net.ipv4.ip_forward parameter was changed to 1 by running the following command
sysctl net.ipv4.ip_forward

51. How sysctl applies the changes to kernel's runtime parameters the moment you run sysctl command?
When you run sysctl commands to change kernel parameters, the changes take effect immediately and affect the real-time behavior of your system.

The sysctl command uses the sysctl() system call to communicate with the kernel and change the value of specified kernel parameters. The kernel then updates the appropriate data structures and applies the new settings immediately.

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

- Identify the kernel parameter that you want to modify and its current value. You can use the sysctl command to view the current values of kernel parameters.
- Decide on the new value that you want to set for the parameter.
- Modify the appropriate configuration file in the /etc/sysctl.d directory or add the parameter to the /etc/sysctl.conf file. To modify an existing file, you can use a text editor such as vi or nano. To create a new file, you can use the touch command. For example, to set the net.ipv4.tcp_syncookies parameter to 0, you can create a new file called /etc/sysctl.d/99-sysctl.conf
- Save the changes to the file(s).
- To apply the changes immediately, run the sysctl command with the -p
- To verify that the changes have been applied, you can run the sysctl
- Finally, reboot the system to ensure that the changes persist across reboots. After the reboot, you can use the sysctl command to verify that the changes are still in effect.

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

No, changes made to kernel settings inside the container do not affect the kernel settings of the server on which the container is running.

Containers run on the host OS and share the same kernel with the host. However, the kernel container view is isolated from the host view, and changes to the kernel settings inside the container only affect the kernel container view.

SSH

SSH

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

SSH is a network protocol used to securely connect to a server or remote computer. To check if Linux server is running SSH you can check if SSH service is running using systemctl command or check if port 22 is open and listening by use the netstat command. You can also try connecting to the server using an SSH client to see if the connection is successful.

55.Why SSH is considered better than telnet?

SSH is considered better than Telnet because it provides a secure and encrypted connection between the client and the server, supports stronger authentication methods, is available on a wide range of platforms, and provides additional functionality beyond remote command-line access. Telnet, on the other hand, sends all data in plain text, has limited support and compatibility with modern systems, and is limited to remote command-line access.

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

~/.ssh/known_hosts file contains a list of host keys for SSH servers that the user has connected to previously. When a user connects to an SSH server for the first time, the server's public key is added to the known_hosts file. On subsequent connections, the client verifies the server's identity by checking the public key against the known_hosts file.

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

In Linux, "Host key verification failed" error message means that the public key of the remote SSH server that you are trying to connect to does not match the key stored in your ~/.ssh/known_hosts file.
This error can occur for a few reasons:

- The SSH server's key has been regenerated since you last connected to it
- You are connecting to a different server with the same IP address or hostname
- Your known_hosts file has been modified or corrupted

To resolve this issue, you can either remove the old key from your known_hosts file or update the file with the new key. When connecting to a new SSH server for the first time, you should always verify the server's key fingerprint to ensure that you are connecting to the correct server.

58.What is the difference between SSH and SSL?

SSH (Secure Shell) is primarily used for secure remote login and command execution on a server. It provides strong authentication, encryption, and integrity protection, ensuring that communications between the client and server are secure and private.

SSL (Secure Sockets Layer), now commonly referred to as TLS (Transport Layer Security), is mainly used for securing web traffic between a web server and a web client (such as a browser). SSL/TLS provides secure communication over HTTP by encrypting the data in transit and verifying the server's identity using digital certificates.

59.What ssh-keygen is used for?

ssh-keygen is a command-line utility used to generate, manage, and manipulate public and private authentication keys for SSH (Secure Shell) protocol. It allows users to create key pairs, which consist of a private key and a public key. The private key is kept on the user's local machine, while the public key is added to the remote server's authorized_keys file. This enables the user to authenticate to the remote server securely without entering a password.

ssh-keygen can also be used to convert keys between different formats, change the passphrase used to encrypt private keys, and perform other key-related tasks. It is a versatile tool that is widely used in Linux environments for secure remote access to servers and systems.

60.What is SSH port forwarding?

SSH port forwarding (also known as SSH tunneling) is a technique used to securely tunnel traffic from one networked device to another using the SSH protocol.

SSH port forwarding works by forwarding a TCP/IP port on the local machine to a port on a remote machine over an encrypted SSH connection. This allows users to securely access remote services that may not be directly accessible from their local network. For example, users can use SSH port forwarding to access a web server running on a remote machine, even if the web server is only listening on its local loopback interface.