Assignment-2 Linux

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

   In Linux FHS, the / (root) directory is the top-level directory in the file system hierarchy, which contains all other directories and files.

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

   i. **/bin:** essential command binaries that are required for the system to boot and run properly.
   ii. **/sbin:** essential system binaries that are required for system administration and maintenance.
   iii. **/usr/bin:** user command binaries that are not essential for the system to run but are used by regular users.
   iv. **/usr/sbin:** system command binaries that are not essential for the system to run but are used by system administrators.
   v. **/etc:** configuration files for the system and applications.
   vi. **/home:** home directories for regular users.
   vii. **/var:** variable data files for the system and applications.
   viii. **/tmp:** temporary files that are meant to be deleted automatically.

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

   The /tmp directory is special because it is used for temporary files that are meant to be deleted automatically by the system, usually on a regular basis. This directory is usually writable by all users on the system.

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

   In the /proc directory, the kernel provides a virtual file system that contains information about running processes and system resources, such as memory usage, network interfaces, and hardware devices.

5. What makes /proc different from other filesystems?

/proc is different from other filesystems because it is a virtual filesystem that provides access to kernel data structures and system information, rather than a traditional filesystem that stores data on disk.

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

   False, any user can create files in /proc, but some files can only be read by the root user

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

   In /proc/cmdline, the kernel provides the command line parameters that were passed to it during boot time.

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

   System devices can usually be found in the /dev directory.

Permissions:

9. How to change the permissions of a file?

   To change the permissions of a file, you can use the chmod command followed by a three-digit number that represents the desired permissions for the owner, group, and others. For example, chmod 644 file.txt would set the file.txt to be readable and writable by the owner, and readable by everyone else.

10. What does the following permissions mean?

    777: The file or directory can be read, written, and executed by anyone.

    644: The file can be read and written by the owner and read by everyone else.

    750: The file can be read, written, and executed by the owner, and read and executed by members of the group that owns the file.

11. What this command does? chmod +x some_file

    This command makes the file named "some_file" executable for the user who owns it, using the symbolic notation "+x" to add the execute permission.

12. Explain what is setgid and setuid

setgid and setuid are special permissions that can be set on files or directories. Setgid sets the group ID for new files and directories to the group that owns the parent directory, while setuid sets the user ID for new files and directories to the user who owns the parent directory.

13. What is the purpose of sticky bit?

The sticky bit is a special permission that can be set on directories, which prevents users from deleting or renaming files that they do not own within that directory. This is commonly used on directories like /tmp, to prevent users from accidentally deleting important files.

14. What the following commands do?

chmod: changes the permissions of a file or directory.

chown: changes the owner of a file or directory.

chgrp: changes the group owner of a file or directory.

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

Sudo is a command that allows users to execute commands with the privileges of another user, usually the root user. It can be set up by adding users to the sudoers file, which lists which users are allowed to use sudo and what commands they can run with it.

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

True, installing packages usually requires root privileges, either by logging in as the root user or by using the sudo command to temporarily elevate the privileges of the current user.

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

ACL stands for Access Control List, which is a mechanism for setting finer-grained permissions on files and directories beyond the standard owner/group/other permissions. ACLs allow you to specify permissions for specific users or groups, which can be useful in situations where the standard permissions are not sufficient. For example, if you have a directory that is owned by one user but needs to be accessed by multiple users, you can use ACLs to grant those users the necessary permissions. ACLs can also be used to grant specific permissions to groups or users that are not the owner or part of the group that owns the file or directory.

Use cases for ACLs include file sharing between users, granting permissions to a service account, allowing a user to access specific files in a shared directory, and many others.

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

There are many reasons why creating a file could fail, some possible reasons include:

1. Insufficient permissions: If you don't have the necessary permissions to create a file in a particular directory, the operation will fail. Check the permissions on the directory and make sure you have the required permissions.
2. Disk full: If the disk or filesystem is full, you won't be able to create any more files until you free up space.
3. File already exists: If a file with the same name already exists in the directory and you're trying to create a new file with the same name, the operation will fail. You may need to choose a different name for the file or delete the existing file first.

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

The command chmod -x $(which chmod) removes the execute permission from the chmod command, which means that you can no longer use it to change permissions on files. To fix this, you can use the full path to the chmod command to restore its execute permission. For example:

sudo chmod +x /bin/chmod

This will restore the execute permission on the chmod command and allow you to use it again.

Scenarios

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

To copy a file to a remote Linux host, you can use the scp command. For example, to copy a file named file.txt to a remote host with IP address 192.168.0.100 and place it in the remote home directory, you can use the following command:

scp file.txt user@192.168.0.100:~

21. How to generate a random string?

To generate a random string in Linux, you can use the openssl command. For example, to generate a random string of 10 characters, you can use the following command:

openssl rand -hex 5

This will generate a random string of 10 characters in hexadecimal format.

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

To generate a random string of 7 characters, you can use the same command as in the previous answer, but specify a different number of bytes:

openssl rand -hex 4

This will generate a random string of 8 characters in hexadecimal format, which is equivalent to 4 bytes. You can then remove the last character to get a string of 7 characters.

Systemd

23. What is systemd?

Systemd is a system and service manager for Linux-based operating systems that is designed to manage the boot process, system initialization, and system services. It is responsible for starting and stopping services, tracking system state and services, logging system events, and managing dependencies between services.

24. How to start or stop a service?

To start a service, you can use the systemctl start command followed by the name of the service, for example:

sudo systemctl start apache2

To stop a service, you can use the systemctl stop command followed by the name of the service, for example:

sudo systemctl stop apache2

25. How to check the status of a service?

To check the status of a service, you can use the systemctl status command followed by the name of the service, for example:

systemctl status apache2

This will show you information about the current state of the service, including whether it is running or stopped, any errors that may have occurred, and more.

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

To display the logs for a service on a system that uses systemd, you can use the journalctl command followed by the name of the service, for example:

journalctl -u apache2

This will show you all of the logs for the Apache service, including any errors or warnings that may have occurred.

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

To make a certain process or application a service, you need to create a systemd unit file for it. A unit file is a configuration file that specifies how systemd should manage a service.

Here are the basic steps to create a systemd unit file for a process/app:

1. Create a new unit file in the /etc/systemd/system directory with a .service file extension. For example, myapp.service.
2. Add the necessary configuration options to the unit file, such as the service name, the executable path, the arguments to pass to the executable, and any dependencies on other services.
3. Save the unit file and reload the systemd configuration to apply the changes with the command systemctl daemon-reload.
4. Start the service using systemctl start myapp.service.

Note that the specific steps for creating a systemd unit file will depend on the application or process you want to run as a service.

28. Troubleshooting and Debugging

Troubleshooting and debugging are important skills for any system administrator or developer. Here are some common tools and techniques used for troubleshooting and debugging:

- System logs: Reviewing system logs can help you identify errors or issues that are occurring on the system. Logs can be found in various locations depending on the system and service, but common locations include /var/log/messages, /var/log/syslog, and /var/log/auth.log.
- Monitoring tools: Tools such as top, htop, and iotop can help you monitor system resources such as CPU usage, memory usage, and disk I/O.
- Network troubleshooting: Tools such as ping, traceroute, and netstat can help you diagnose network issues.
- Debuggers: Debuggers such as gdb can help you trace and diagnose issues in software applications.
- Performance profiling: Tools such as perf can help you analyze system performance and identify performance bottlenecks.
- System utilities: Utilities such as ps, lsof, and strace can help you diagnose issues with processes, file systems, and system calls.

29. Where system logs are located?

System logs are typically located in the /var/log directory on Linux systems.

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

To follow a file's content as it's being appended without opening the file every time, you can use the "tail" command with the "-f" option. For example, "tail -f /var/log/syslog" will continuously display the latest lines added to the syslog file.

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

For troubleshooting and debugging network issues, I would use various tools such as ping, traceroute, netstat, tcpdump, and Wireshark.

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

For troubleshooting and debugging disk and file system issues, I would use tools such as fsck, df, du, lsof, and strace.

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

For troubleshooting and debugging process issues, I would use tools such as ps, top, pmap, and strace.

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

For debugging CPU-related issues, I would use tools such as top, htop, mpstat, and perf.

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

If someone claims that their system is slow, I would start by asking them to provide more details about the symptoms they are experiencing, such as what specifically is slow, when did the issue start, and if they made any changes to the system recently. I would then use various tools to diagnose the issue, such as top, vmstat, iostat, and strace.

36. Explain iostat output

iostat is a command-line tool that is used to monitor input/output (I/O) statistics for devices and partitions. Its output includes information on the number of reads and writes per second, the amount of data read and written, the average time to read and write, and the percentage of CPU time used for I/O operations.

37. How to debug binaries?

To debug binaries, you can use tools such as gdb, strace, and ltrace.

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

CPU load refers to the number of processes waiting to be executed, while CPU utilization refers to the amount of time the CPU is being used to execute processes.

39. How you measure time execution of a program?

You can measure the time execution of a program by using the "time" command followed by the name of the program. For example, "time ls" would measure the time it takes to execute the "ls" command.

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 identify the process writing to a file, you can use the "lsof" command followed by the path of the file. This will show you a list of all processes that have the file open. From there, you can use the "kill" command with the process ID to terminate the process.

Kernel

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

A kernel is the core component of an operating system that manages system resources and provides services to applications.

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

You can find out which Kernel version your system is using by using the "uname -r" command.

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

A Linux kernel module is a piece of code that can be dynamically loaded into the kernel at runtime to add new functionality. You can load a new module using the "insmod" command followed by the name of the module.

44. Explain user space vs. kernel space

User space refers to the portion of memory that is used by user applications, while kernel space refers to the portion of memory that is used by the operating system kernel.

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

You can change the kernel configuration during the build process or by modifying the configuration file located in the /boot directory.

46. Where can you find kernel's configuration?

The kernel's configuration file can typically be found in the /boot directory, and is named "config-<kernel version>".

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

The file that contains the command passed to the boot loader to run the kernel can typically be found in the /boot directory, and is named "grub.cfg".

48. How to list kernel's runtime parameters?

You can list the kernel's runtime parameters by using the "sysctl -a" command.

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

Running "sysctl -a" as a regular user will not display any results as it requires root privileges to view and modify kernel parameters.

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 following command:

sudo sysctl net.ipv4.ip_forward=1

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

When you run the "sysctl" command to change kernel parameters, it immediately updates the values of the corresponding kernel variables in the running system.

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

Changes made to kernel runtime parameters persist even after a system reboot. The changes are stored in the "/etc/sysctl.conf" file. The file contains a list of kernel parameters and their corresponding values, and the system reads this file during boot to apply the changes.

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?

The changes made to kernel parameters in a container affect only the kernel parameters of that container, and not the host on which the container runs. The host's kernel parameters remain unaffected.

SSH

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

"Host key verification failed" error occurs when the SSH client fails to verify the identity of the remote server because the public key of the server has changed since the last time the client connected to it. This may indicate a security issue such as a man-in-the-middle attack.

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

SSH (Secure Shell) is a network protocol that provides secure encrypted communication between two untrusted hosts over an insecure network. To check if a Linux server is running SSH, you can run the following command:

sudo service ssh status
or
systemctl status ssh

56. Why SSH is considered better than telnet?

SSH is considered better than telnet because it provides encryption and authentication, which means that the data sent between the two hosts is secure and cannot be intercepted by attackers.

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

The ~/.ssh/known_hosts file stores the public keys of remote servers that you have connected to using SSH. The file is used by SSH to verify the identity of the remote server when you connect to it.

58. What is the difference between SSH and SSL?

SSH and SSL (Secure Sockets Layer) are both protocols used for secure communication over a network, but they serve different purposes. SSH is used for remote login and executing commands on remote systems, while SSL is used for secure communication between web browsers and web servers.

59. What ssh-keygen is used for?

ssh-keygen is a command-line utility that is used to generate, manage, and convert SSH authentication keys. It is typically used to generate public and private key pairs for use with SSH.

60. What is SSH port forwarding?

SSH port forwarding, also known as SSH tunneling, is a technique that allows you to forward traffic from a local port on your machine to a remote port on a remote machine, over an encrypted SSH connection. This can be used to securely access remote services or to bypass firewalls and other network restrictions.