Linux

Assignment-2

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

In the Linux FHS (Filesystem Hierarchy Standard), the / directory is the root directory, which is the starting point of the directory tree.

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

/bin, /sbin, /usr/bin and /usr/sbin

/etc

/home

/var

/tmp

/bin: This directory contains executable programs that are essential for booting the system and running basic system commands. Examples include ls, cp, and mv.

/sbin: This directory contains executable programs that are essential for system administration, such as disk partitioning and system recovery. Examples include fdisk and iptables.

/usr/bin: This directory contains non-essential executable programs that are not required for system administration or booting the system. Examples include games and productivity software.

/usr/sbin: This directory contains non-essential executable programs that are not required for system administration, but may be necessary for certain system services. Examples include httpd and sendmail.

/etc: This directory contains system configuration files. Examples include passwd and fstab.

/home: This directory contains the home directories of regular users.

/var: This directory contains variable data files, such as logs and spool files.

/tmp: This directory is used to store temporary files. It is typically emptied on reboot, and may be stored in memory for faster access.

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

The important difference is that the /tmp directory is often stored in memory (i.e., RAM) rather than on disk, which allows for faster access to the temporary files. This can be especially important for applications that need to access temporary files frequently, such as compilers or editors.

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

The /proc directory is a special filesystem that contains information about running processes and system configuration. One can find various system and process related information such as memory usage, CPU usage, system uptime, kernel version, loaded modules, etc.

1. What makes /proc different from other filesystems?

The /proc filesystem is different from other filesystems because it is not a physical filesystem, but rather a virtual filesystem that is generated on the fly by the kernel. It provides an interface to kernel data structures that can be used to extract information about the system and its processes.

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

It is true that only root can create files in the /proc directory.

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

The /proc/cmdline file contains the command line parameters that were passed to the kernel when it was booted. This includes information such as the kernel version, the root device, and any kernel parameters that were specified at boot time.

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

System devices, such as block storage devices, can typically be found in the /dev directory.

**Permissions**

1. How to change the permissions of a file?

To change the permissions of a file, you can use the chmod command in a terminal. The syntax is:

chmod [permissions] [filename]

1. What does the following permissions mean?:

777

644

750

chmod 777 myfile.txt: This gives read, write, and execute permissions to the owner, group, and everyone else who has access to the file.

chmod 644 myfile.txt: This gives read and write permissions to the owner, and read-only permissions to the group and everyone else.

chmod 750 myfile.txt: This gives read, write, and execute permissions to the owner, and read and execute permissions to the group. Everyone else has no access to the file.

1. What this command does? chmod +x some\_file

The command chmod +x some\_file gives execute permission to the owner of the file some\_file. Specifically, the +x option adds the execute permission to the file for the owner.

1. Explain what is setgid and setuid

Setgid and setuid are special permissions in Linux/Unix systems. Setgid allows a user who is a member of a group that owns a file or directory to inherit the group ownership of that file or directory when creating new files or directories within it. Setuid allows a user to execute a program with the permissions of the file's owner, rather than the user's own permissions.

1. What is the purpose of sticky bit?

The purpose of the sticky bit is to restrict the deletion of files in a directory. When the sticky bit is set on a directory, only the owner of a file or the root user can delete the file, even if other users have write permission to the directory. The sticky bit is represented by the "t" bit in the permissions of a directory.

1. What the following commands do?

chmod

chown

Chgrp

chmod is a command in Linux/Unix systems used to change the permissions of files and directories. It can be used to add or remove read, write, and execute permissions for the owner, group, and other users.

chown is a command used to change the ownership of files or directories. It can be used to change the user owner and/or group owner of a file or directory.

chgrp is a command used to change the group ownership of a file or directory. It can be used to change the group owner of a file or directory to a specific group.

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

sudo is a command in Linux/Unix systems that allows a user to execute commands as another user, typically the root user, which has administrative privileges. By default, only the root user has the permissions to perform certain administrative tasks, but sudo allows other users to perform those tasks as well.

1. 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 have administrative privileges, which can be achieved by either logging in as the root user or using the sudo command to execute the package installation command with administrative privileges.

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

ACLs (Access Control Lists) are a mechanism in Linux/Unix systems that allow more granular control over file and directory permissions than traditional Unix file permissions. ACLs allow you to set permissions for specific users and groups, rather than just the owner, group, and others as in traditional Unix file permissions. ACLs are useful in scenarios where you need to grant different levels of access to different users or groups for the same file or directory. For example, if you have a file that needs to be accessible by multiple users but you want to restrict some users from modifying it, you can use ACLs to achieve that.

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

* Insufficient permissions: If you don't have the necessary permissions to create files in the target directory, the file creation will fail.
* Insufficient disk space: If the disk is full or there isn't enough space for the new file, the file creation will fail.
* File name or path errors: If the file name or path contains invalid characters or exceeds the maximum length allowed by the filesystem, the file creation will fail.

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

If a user accidentally executes the command chmod -x $(which chmod), it will remove the execute permission from the chmod command, which is required to change file permissions. To fix it, you can use the full path to the chmod command, which is /bin/chmod, to restore the execute permission. The command to fix it would be sudo /bin/chmod +x $(which chmod).

**Scenarios**

1. 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 (secure copy) command. The syntax is:

scp /path/to/local/file username@remotehost:/path/to/destination

Replace /path/to/local/file with the path to the file you want to copy, username with the username of the remote host, remotehost with the hostname or IP address of the remote host, and /path/to/destination with the destination path on the remote host.

1. How to generate a random string?

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

openssl rand -hex 5

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

To generate a random string of 7 characters, you can modify the command above to generate half the number of characters and then use the head command to truncate the output:

openssl rand -hex 4 | head -c 7

**Systemd**

1. What is systemd?
2. How to start or stop a service?
3. How to check the status of a service?
4. On a system which uses systemd, how would you display the logs?
5. Describe how to make a certain process/app a service
6. Troubleshooting and Debugging
7. Where system logs are located?
8. How to follow file's content as it being appended without opening the file every time?
9. What are you using for troubleshooting and debugging network issues?
10. What are you using for troubleshooting and debugging disk & file system issues?
11. What are you using for troubleshooting and debugging process issues?
12. What are you using for debugging CPU related issues?
13. You get a call from someone claiming "my system is SLOW". What do you do?
14. Explain iostat output
15. How to debug binaries?
16. What is the difference between CPU load and utilization?
17. How you measure time execution of a program?

**Scenarios**

1. 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?

**Kernel**

1. What is a kernel, and what does it do?
2. How do you find out which Kernel version your system is using?
3. What is a Linux kernel module and how do you load a new module?
4. Explain user space vs. kernel space
5. In what phases of kernel lifecycle, can you change its configuration?
6. Where can you find kernel's configuration?
7. Where can you find the file that contains the command passed to the boot loader to run the kernel?
8. How to list kernel's runtime parameters?
9. Will running sysctl -a as a regular user vs. root, produce different result?
10. You would like to enable IPv4 forwarding in the kernel, how would you do it?
11. How sysctl applies the changes to kernel's runtime parameters the moment you run sysctl command?
12. How changes to kernel runtime parameters persist? (applied even after reboot to the system for example)
13. Are the changes you make to kernel parameters in a container, affects also the kernel parameters of the host on which the container runs?

**SSH**

1. What is SSH? How to check if a Linux server is running SSH?
2. Why SSH is considered better than telnet?
3. What is stored in ~/.ssh/known\_hosts?
4. You try to ssh to a server and you get "Host key verification failed". What does it mean?
5. What is the difference between SSH and SSL?
6. What ssh-keygen is used for?
7. What is SSH port forwarding?