Linux Programming: Assignment-5

Samriddhi Guha | ENG24CY0157 | 3-C (25)

1. What is a shell in Linux OS? How many categories of shell is currently exists in Linux? Why bash shell is very popular in Linux distribution? (CO2)

A shell is a command-line interpreter in Linux that serves as a bridge between users and the operating system's kernel. It takes the user's command (or script), transforms it into a language that the kernel recognizes, runs the program(s), and sends the result back to the user.

Shells are generally categorized by their key characteristics; for the most part, the syntax they use and their family of origin are determining characteristics. The most commonly found categories are:
Bourne Shell (sh) family: Original sh, Bash (Bourne-Again Shell), ksh (Korn Shell), zsh (Z Shell)
C Shell (csh) family: csh and tcsh (TENEX C Shell)

Popularity of bash:

- *Compatibility:* Since Bash is a superset of the original Bourne Shell (sh), it maintains a high degree of compatibility with the older scripts.
- *Features:* Bash has robust features such as command line editing, command history which allows users to see and recall previously used commands, job control, and extensive shell scripting capabilities.
- *Default:* Bash is the default shell on nearly all of the major Linux distributions (eg. Ubuntu, Red Hat/Fedora, and others). This guarantees considerable use and familiarity by users and administrators.

2. What does the ls -Z command display? (CO1)

The ls -Z command is used to show the files and folders in the present working directory with their SELinux (Security Enhanced Linux) security content.

Example:

```
samriddhi@samriddhi:~$ ls -Z
```

Here,

- -Z - Option used along with ls to display the SELinux content.
- SELinux Content- It contains the user, role, type and sensitivity of the files that are used to enforce access control policies.

3. Write a command to list all hidden files in the current directory. (CO1)

The command *ls -a* can be used to display hidden files in the current directory. Essentially, all hidden files and directories start with a (.) including hidden directories and parent directories start with (..).

Example:

```
samriddhi@samriddhi:~$ ls -a
```

Here,

- ls - Used to list files in the current directory
- -a - Used as an option to check all the hidden files in the present working directory.

4. Explain the difference between hard links and soft links (symbolic links) in Linux. (CO1)

Feature	Hard Link	Soft Link (Symbolic Link)
Creation Command	ln [target_file][link_name]	ln -s [target_file][link_name]
Inodes	Shares the same inode number as the original file.	Has a different inode number as it is a new file that stores the path to the original.
File Type	Acts as an alias for the original file; the OS treats it as the same file data.	Acts as a pointer or a shortcut to the original file.
File System	Cannot span across different file systems.	Can span across different file systems.
Deletion	Deleting the original file does not break the link; the data remains accessible until the last hard link is removed.	Deleting the original file breaks the link, making the soft link "dangling" or unusable.
Directory Linking	Cannot be created for directories (only files).	Can be created for both files and directories.

5. A file has permissions -rwxr-x--x. Explain who can read, write, and execute it. (CO1)

A file having permissions -rwxr-x–x can be explained as:

- (-rwx): Only accessible by owner, who can read, write and execute. Owner can perform all actions on the file.
- (r-x): Accessible by a group, who are only granted permission to read and execute the file. They cannot modify the file but can run it.
- (--x): Accessible by all other users on the system. They can only execute the file, neither read nor write to it.

6. Write the command to change the group ownership of a file data.txt to group staff. (CO1)

The command chgrp can be used to change ownership of a file data.txt to group staff.

Example:

```
samriddhi@samriddhi:~$ chgrp staff data.txt
```

Here,

- chgrp: Used to change group ownership.
- staff: The name of new group to assign to the file.
- data.txt: The target file whose ownership is being changed.

7. Why is it dangerous to give 777 permissions to a file? Explain with an example. (CO1)

When a file is given *777 permissions*, more elaborately, the *-rwxrwxrwx* permissions, it puts the file into a major security risk as every user on the system which includes the owner, group and others, will get access to read write and execute data on the file.

An advisory can use this as an opportunity as he can:

- Read sensitive data on the file.
- Perform modifications or corrupt file contents by writing into it.
- Execute the file and run malicious code.

Hence it can be dangerous to give a file 777 permissions as it makes the file more sensitive and vulnerable to attacks or corruption.

8. What is the difference between apropos (i.e., man -k) and whatis (i.e., man -f)? (CO1)

Command	Equivalent	Function	Search Method	Detail Level
whatis	man -f	Displays a one-line description of the command's purpose.	Requires an exact match for the command name.	Very brief; quick reference.
apropos	man -k	Searches the manual page descriptions for keywords and displays relevant results.	Performs a keyword search (approximate/fuzzy) across all manual pages.	Lists all relevant commands and their one-line descriptions.

9. Write a command to redirect the error output of a command to a file named error.log. (CO1)

Our objective is to redirect the error output of a command to a file named error.log. The error output in Linux is sent to Standard Error (file descriptor 2) and in order to redirect it, we use the descriptor number followed by the *redirection operator (>).*

```
samriddhi@samriddhi:~$find /root -name "*.txt" 2> error.log
```

Here,

- 2: The file descriptor for Standard Error (stderr).
- (>): The redirection operator to send output to a file.
- error.log: The file where all error messages from the find command will be saved.

10. How can you use the tee command to append output to a file instead of overwriting it? (CO4)

The tee command can be used along with -a option (append) to append data to a file instead of overwriting its contents.

Example:

```
samriddhi@samriddhi:~$ls -l | tee -a output.txt
```

Here,

- | (pipe): Takes the output of ls -l and sends it as input to the tee command.
- tee: Writes the input to a file and simultaneously displays it on the terminal.
- -a (append): Modifies tee's behavior to add the new output to the end of output.txt instead of replacing the file's existing content.