School Of Engineering

Linux Programming
Assignment-6

Name: Neehara Lakshmi
USN No: ENG24CY0138
Roll No: 55
Section: B (CyberSecurity)
Semester: 3rd

Q1. Which command is used to list the contents of a directory? Justify with proper example.

Ans: The command used to list the contents of a directory in Linux is **ls**. This command displays the names of files and subdirectories within the specified directory. If no directory is provided, it lists the contents of the current working directory. The ls command is highly versatile and supports various options to display additional information, such as file permissions, sizes, and hidden files.

The command to list directory contents is `ls`.

Example Code:

```
$ ls
file1.txt  folder1  folder2
```

This shows the directory contains one file and two folders.

Q2. Write the command to create a new directory named 123test_dir.

Ans: The command to create a new directory named `123test_dir` is:

Example Code:

```
mkdir 123test_dir
ls
123test_dir  file1.txt  folder1  folder2
```

Q3. What is the purpose of the sed command? Justify with proper example.

Ans: The sed command (stream editor) is used for parsing and transforming text in Linux. It performs basic text editing operations on input streams without user interaction.

Purpose:

- Find and replace text
- Delete lines
- Insert/append text
- Search for patterns
- Text transformation

To Replace "apple" with "orange" in a file:

Example Code:

```
sed 's/apple/orange/g' fruits.txt
```

Q4. Which distinct command is used to display one-line descriptions of any commands?

Ans: The command used to display one-line descriptions of any commands is `whatis`. It searches the manual page names and displays a short description from the manual page synopsis.

Example Code:

```
whatis ls
```

Output:

```
ls (1)              - list directory contents
```

Q5. Write the command to create an empty file named “notes.txt”.

Ans: The command to create an empty file named "notes.txt" is:

Example Code:

```
touch notes.txt
```

Q6. Differentiate between grep and awk commands with an example.

Ans:

grep

- Searches for patterns in text files.
- Outputs entire lines containing the pattern.
- Simple syntax for basic text searching.
- Uses regular expressions.
- Fast for line-based pattern matching.
- **Example Code:** grep "error" server.log
- **Output:**

```
[2023] error: connection failed
[2024] critical error: timeout
```

awk

- Programming language for text processing.
- Processes files field by field.
- Can perform calculations and transformations.
- Built-in variables.
- Handles structured data efficiently.
- **Example Code:** awk -F: '{print $1, $7}' /etc/passwd
- **Output:**

```
root /bin/bash
daemon /usr/sbin/nologin
john /bin/zsh
```

When to Use:

Use ‘grep’ when you just need to find lines matching a pattern

Use ‘awk’ when you need to process, transform, or extract specific fields from structured data

Q7. Write the command to give read, write, and execute permission to the owner of a file script.sh.

Ans: The command to give read, write, and execute permission to the owner of a file `script.sh` is:

Example Code:

```
chmod u+rwx script.sh
```

Explanation:

- 'chmod' - Change file mode bits.
- 'u' - Refers to the file owner (user).
- '+' - Adds the specified permissions.
- 'rwx' - Read, Write, and Execute permissions.

Q8. How is chown different from chgrp? Give one example for each.

Ans:

chown (Change Owner)

- Changes both the **owner** and optionally the **group** of a file/directory
- Can change owner and group simultaneously
- Requires root/sudo privileges for most changes
- **Example Code:** sudo chown john file.txt

chgrp (Change Group)

- Changes only the **group** ownership of a file/directory
- Cannot change the owner, only the group
- Users can change group to any group they belong to
- **Example Code**: chgrp developers file.txt

Q9. A user complains that they cannot execute a file even though it exists in their Directory. How would you troubleshoot this using ls -l, chmod, and whoami?

Ans : To troubleshoot why a user cannot execute a file in their directory, follow these steps:

1. Identify the current user by running `whoami`. This confirms the username you are logged in as, ensuring you are the expected user.

2. Check the file's permissions and ownership using `ls -l filename`. This displays the permissions, owner, and group of the file. Look for the execute permission in the permissions string.

3. Verify if the current user is the owner of the file. If the owner from 'ls -l' matches the output of 'whoami', but the execute permission is missing for the owner, you need to add it.

4. Add execute permission for the owner using 'chmod u+x filename'. This command grants execute permission to the user (owner) of the file.

5. Confirm the changes by running `ls -l filename` again. The permissions should now include 'x' for the owner.

6. Test execution by running the file. It should now execute successfully.

Q10. Design a command pipeline to: find all .log files modified in the last 2 days in /var/log, display them on screen, and save the results into a file recent_logs.txt using tee command.

Ans: The command pipeline to find recent log files and save the results using `tee`:

Example Code:

```
find /var/log -name "*.log" -mtime -2 | tee recent_logs.txt
```

Command Breakdown:

- **‘ find /var/log ’** - Search starting from the /var/log directory
- **‘ -name "*.log" ’** - Find files ending with .log extension
- **‘ -mtime -2 ’** - Filter files modified in the last 2 days
- **‘ | tee recent_logs.txt ’** - Pipe results to `tee` which displays output on screen AND saves to recent_logs.txt

THANK YOU