Lab Cycle-2 : Open Source Platforms - Lab- 20INMCA332

1. **Describe how to navigate between directories in Linux using command-line tools.**
    a. What command would you use to move to a parent directory, and how does it work?
    b. How can you display the current working directory? Give an example.
    c. What command is used to list the contents of a directory, and what options can you use with it to display hidden files?
2. **How can you manipulate files and directories in Linux using file manipulation commands**
    a. What command would you use to create a new empty file in Linux? Provide an example of its usage.
    b. How would you delete a file in Linux, and what safety precautions should you take before performing this operation?
    c. What command would you use to move a file from one directory to another, and what is the role of the -i option in this command?
3. **Explain the process of changing file permissions in Linux.**
    a. How can you view the permissions of a file in Linux? Provide an example output.
    b. What does the command chmod 755 filename do? Break down the permission change for this example.
    c. How do you give read and write permissions to a specific user for a particular file? Provide the exact command.
4. **In Linux, directory management is a crucial task for system administrators and users to effectively organize and navigate the file system. Explain the key directory commands in Linux, focusing on their usage, syntax, and practical applications. In your answer, include the following:**
    a. Identifying the Current Directory Path:
    b. Changing the Current Directory:
    c. Creating a Directory:
    d. Removing a Directory:
    e. Listing the Contents of a Directory:
    f. Listing the Contents of a Directory with file access permissions:
5. **You are working on a Linux server that contains a directory named *projects* with several text files of varying sizes and contents. Perform the following tasks using appropriate Linux file commands:**
    a. Display the content of a file named report.txt. Ensure that you view the file's content without scrolling, and also handle cases where the content exceeds the terminal's window size.
    b. Using the head command, display the first 10 lines of the file logfile.txt.
    c. Use the tail command to show the last 20 lines of the same file.
    d. Copy the file project1.txt from the projects directory to a backup directory named backup_projects. Ensure that the file is copied correctly and verify the copy in the new directory.
    e. Identify a file named draft.txt that is no longer needed. Remove it from the projects directory. Verify that the file is deleted and cannot be found in the directory anymore. f) Move a file named summary.txt from the projects directory to a subdirectory named archive. Also, rename the file to summary_old.txt while moving it. Verify both the file's location and the new name.

6. **You are a system administrator tasked with ensuring that files on a Linux server are securely managed. The server contains a variety of files, and you must apply the appropriate file access permissions to protect sensitive data.**
7. **Perform the following tasks related to securing files using File Access Permissions (FAPs) in Linux:**
    a. You need to examine the access permissions of a file named project_report.txt located in the /home/user/projects/ directory. Use the appropriate command to view the file's permissions, and explain what each part of the output represents.
    b. The file financial_data.csv located in /home/user/private/ needs to be modified so that only the owner can read and write to the file, and the group and others have no permissions. Change the file's permissions accordingly and explain the command you used.
    c. The file public_image.jpg located in /home/user/images/ needs to be accessible by the owner for reading, writing, and executing, while the group should only have read and execute permissions, and others should have no access. Change the permissions using numeric representation (octal values).
    d. A script file named backup.sh located in /home/user/scripts/ is not executable by the owner. You are required to modify the permissions to allow the owner to execute the script. Describe how you would grant the owner execute permissions, and explain the impact of the changes.
    e. You are required to ensure that the directory /home/user/secure_folder/ is only accessible to the owner for reading, writing, and executing files, while the group and others have no access. After making the necessary changes, verify the directory's permissions.
    f. Explain how directory permissions differ from regular file permissions and the importance of setting them correctly.
8. **creating a Bash script to perform two specific functions:**
    a. Check if a Number is Even or Odd:
        i. The script should prompt the user to input a number and check if the number is even or odd.
        ii. If the number is even, it should print "The number is even."
        iii. If the number is odd, it should print "The number is odd."
        iv. The script should handle invalid inputs (e.g., non-numeric inputs) by printing "Invalid input, please enter a number."
    b. Print a Message Based on the Day of the Week:
        i. The script should prompt the user to input a day of the week (e.g., "Monday").
        ii. Using a case construct, it should print a message depending on the day:
        iii. ▪ If it's "Monday", print "Start of the work week."
        iv. ▪ If it's "Friday", print "Almost weekend!"
        v. ▪ For any other day, print "Keep going!"
9. **Write a shell script using if-else that takes a user input number and checks whether it is divisible by both 2 and 3. Print a message stating whether it is divisible by both, just by one, or by neither.**

10. **Create a shell script using the case statement that accepts a month number (1-12) from the user and prints the corresponding month name. For example, if the user enters 1, the script should print "January".**

11. **Write a shell script using if-elif that prompts the user for a number and assigns a rating based on the input. If the number is between 90 and 100, the rating should be "Excellent". Between 70 and 89, it should be "Good", and below 70 should be "Needs Improvement".**

12. **Write a shell script that uses a for loop to iterate through the files in a directory and checks if each file is readable, writable, and executable. Print the permissions for each file in a human-readable format (e.g., "Readable, Writable, Executable" or the combination that applies).**

13. **Write a shell script using a while loop that prints all the prime numbers between 1 and 100. Use a nested while loop to check if a number is divisible by any number other than 1 and itself.**