Linux Practical Assignment 4

Create a word file having all the command output with screenshot

Upload File in .pdf format , add all command output
Each page header must contains- Full Enrollment no- Full Name
Each page footer must contains- SVGU MCA SEM 1 SUB- Linux Foundation

Your Linux Prompt Must be - MCAFullEnrollmentNo:~$

FileName= FullEnrollmentNo_Surname_Name.pdf

Set 1

1.Create a directory named /digitribe.

2.Allow user1 and user2 to share documents in the /digitribe directory using a group called rhce.

3.Both of them can read, write and remove documents from /digitribe directory, but any user not member of the group rhce unable to read, write and remove from /digitribe directory.

4.Schedule a Backup Script Create a cron job that runs a script named backup.sh every day at 2:00 AM.The script is located in /home/user1/scripts/.

Set 2

Write a Shell Script to perform the following operations:

1. **Count the number of hidden files in the current directory.**
2. **Count the number of regular files larger than 1 MB.**

3. Display the number of symbolic links in the current directory.

4. Show the three smallest files in the current directory.

5. Display the oldest file (based on modification time).

6. List all directories in /var.

7. List all files with .log extension in the current directory.

8. Display all files owned by the current user.

9. Display the total disk usage of the current directory.

10. Exit.

Set 3

1. Create a file named system.log containing at least 20 lines. Display only the first 8 lines of the file.
2. Using the same file, display only the last 12 lines of system.log. 3. Combine head and tail to display lines 5 to 15 of a file named report.txt. 4. Search for all lines containing the word "error" inside system.log. 5. Search for the word "root" inside /etc/passwd, ignoring case. 6. Display only the lines that do NOT contain the word "success" inside results.txt.
7. Count the total lines, total words, and total characters in a file named notes.txt.
8. Count only the number of lines in a file named data.txt. 9. Search for all files under /home that have the extension .txt. 10. Search inside all .log files under /var/log for lines containing "failed login".

Set 4

1. Create three users: user1 with uid/gid 2000, password digitribe and user2 with uid/gid 3000, password digitribe and user3 with nologin shell.
2. user1 and user2 should be member of group rhce.
3. Make user1 account validity stopping one month.
4. user2 minimum password age 2 days, maximum password age 60 days and set warning to 5 days. Allow user1 to get full access to user2 home

directory.

5. Clean Temp Files: Schedule a cron job that deletes all files from /tmp/project/ every 6 hours.

Set 5

Write a Shell Script to display the following Operations.

1. Count the number of uppercase and lowercase letters in a file.
2. Display all words ending with "ing".
3. Count and display the number of words starting with A–M.
4. Count occurrences of a user-provided pattern.
5. Display all words sorted according to length (shortest → longest).
6. Display a file sorted by line length in descending order.
7. Display each line reversed word-by-word (not character-by-character).
8. Display the owner, group, and permissions of a file.
9. Display whether a given file is a directory, regular file, link, or socket.
10. Display only the lines that contain exactly 3 words.

Set 6

Write a Shell Script to perform the following Operations.

1. Count the no. of chars, words, lines.
2. Count the no. of words that exactly 5 & display them.
3. Count the no. of words that start with '0' & end by '*'.
4. Count the no. of occurrences of a particular word.
5. Display all the words of file in ascending order.
6. Display a file in descending order.
7. Display a file in reverse.
8. Display type of a file.
9. Display the attributes of a file.
10. Display all the lines that have length exactly ten.

Set 7

1. Create a user named trainer and set an initial password.

2. Ensure the trainer user must change their password upon first login.

3. Create a group called developers with a GID of 45000.

4. Grant the developers group passwordless sudo access to run only the /usr/bin/apt command without editing /etc/sudoers directly.

5. Create dev1, dev2, and dev3 users and add them to the developers group.

6. Set the users dev1, dev2, and dev3 to expire after 120 days.

7. Configure dev1 to require a password change every 10 days and dev2 every 20 days.

8. Lock the dev3 account temporarily and later unlock it.

Set 8

1. **Create a group named projectgrp.**
2. **Create two users named Mahesh and Ramesh.**
3. **Ensure that newly created users must change their passwords every 30 days.**
4. **Add both users to the projectgrp group.**
5. **Create a shared directory /projectX.**
6. **Assign group ownership of /projectX to projectgrp.**
7. **Apply SGID so new files inherit the group.**
8. **Weekly Disk Usage Report: Schedule a cron job to save the output of df -h into /home/student/disk_report.txt every Monday at 8:30 AM.**

Set 9

1. Create a file named examfile.txt.
2. Set permissions: user (read/write), group (read), others (no permission).
3. Change file ownership to user examuser and group teamA. 4. Add execute permission for the user and group using symbolic mode. 5. Create a directory named /public_exam.
6. Give full permissions to all users.
7. Apply the Sticky Bit to prevent users from deleting each other's files.

SUID and SGID Permissions:

8. Create a script named checkID.sh.
9. Add content to display the current user ID.
10. Apply SUID permission on the script.
11. Create a directory /grpStorage and apply SGID permission on it.

-

Set 10

1.Schedule a recurring job as the student user that appends the current date and time to the /home/student/my_first_cron_job.txt file every two minutes. Use the date command to display the current date and time. The job must run only from one day before to one day after the current time. The job must not run on any other day.

2. list the scheduled recurring jobs. Inspect the command that you scheduled to run as a recurring job in the preceding step. Verify that the job runs the /usr/bin/date command and appends its output to the /home/student/my_first_cron_job.txt file.

3.Have your shell prompt sleep until the /home/student/my_first_cron_job.txt file is created as a result of the successful execution of the recurring job that you scheduled. Wait for your shell prompt to return.

4.The while command uses! test -f to continue to run a loop and sleeps for one second until the my_first_cron_job.txt file is created in the /home/student directory.

student@server ~] $ while! test -f my_first_cron_job.txt; do sleep 1s; done

5.Verify that the contents of the /home/student/my_first_cron_job.txt file match the output of the date command.

6.Remove all the scheduled recurring jobs for the student user and list the jobs.

Set 11

1. Create a user named analyst with a password and force password reset on first login.

2. Create a group called sysops with GID 47000.

3. Grant passwordless sudo to sysops group to run only /usr/bin/dnf, using /etc/sudoers.d/sysops.

4. Create users ops1, ops2, and ops3, and add them to sysops.

5. Set account expiration for ops1, ops2, ops3 after 100 days.

6. Apply password policies:

ops1: password change every 7 days

ops2: password change every 14 days

7. Lock the ops3 account, then later unlock it.

8. Create a directory /exam_area, give full permissions to everyone, apply the Sticky Bit, create a script showUser.sh that prints current UID, and assign SUID; also create directory /sysData and apply SGID.

Set 12

Write a Shell Script to perform the following operations:

1. Count the number of blank lines, non-blank lines, and comment lines (starting with #).

2. Display all words that contain only digits.

3. Count and list all words that start and end with the same letter.

4. Count occurrences of a specific line in the file (exact match). 5.

Display all words of the file sorted in reverse dictionary order. 6.

Display file contents in reverse line order.

7. Display the last modification date, creation time of the file.

8. Display whether the file is empty or not, and its type.

9. Display all lines where line length is between 5 and 12 characters.

-

Set 13

1. create a student user and switch to the root user.

2. ensure that newly created users must change their passwords every 30

days. 3. Create the consultants group with a GID of 35000.

4. Configure administrative rights to enable all consultants group members to execute any command as any user. Avoid using visudo to edit the /etc/sudoers file, instead, follow the best practice of placing the configuration file in the /etc/sudoers.d directory.

5. Create the consultant1, consultant2, and consultant3 users with the

consultants group as their supplementary group.

6. Set the consultant1, consultant2, and consultant3 passwords to redhat. 7. Set

the consultant1, consultant2, and consultant3 accounts to expire in 90 days from

the current day.

8. Change the password policy for the consultant2 account to require a new password every 15 days. Additionally, force the consultant1, consultant2, and consultant3 users to change their passwords on the first login.

Set 14

1. Create a user named analyst and assign /bin/bash as the login

shell. 2. Create a group named research with a specific GID of 36000.

3. Add the analyst user to the research group.

4. Configure all members of research to run any command using sudo only after providing a password.

5. Set the analyst account to expire after 60 days.

6. Configure analyst to be locked automatically after 3 failed login attempts.

7. Enforce a password expiration policy of 45 days for the entire research group.

8. Create a new user intern with a restricted shell /usr/sbin/nologin.

-

Set 15

Write a Shell Script to perform the following operations:

1. Count the number of uppercase and lowercase letters in a file.

2. Display all words ending with "ing".

3. Count and display the number of words starting with A–M.

4. Count occurrences of a user-provided pattern.

5. Display all words sorted according to length (shortest → longest).

6. Display a file sorted by line length in descending order.

7. Display each line reversed word-by-word (not character-by-character).

8. Display the owner, group, and permissions of a file.

9. Display whether a given file is a directory, regular file, link, or socket.

10.Display only the lines that contain exactly 3 words.

Set 16

1. Log in to server as the student user. Run the sudo -i command at the shell prompt to

become the root user. Use student as the student user password.

2. Create a /home/techdocs directory.

3. Change the group ownership of the /home/techdocs directory to the techdocs group.

4. Verify that users in the techdocs group cannot create files in the /home/techdocs

directory.

5. Set permissions on the /home/techdocs directory. On the /home/techdocs directory,

configure setgid (2); read, write, and execute permissions (7) for the owner/user and group; and no permissions (0) for other users.

6. Verify that the permissions are set properly. The techdocs group now has write permission.

7. Confirm that users in the techdocs group can now create and edit files in the /home/
techdocs directory. Users that are not in the techdocs group cannot edit or create files in the /home/techdocs directory. The tech1 and tech2 users are in the techdocs group.

The database1 user is not in that group.

8. Modify the /etc/login.defs file to adjust the default umask for login shells. Normal users should have a umask setting that allows the user and group to create, write and execute files and directories, while preventing other users from viewing, modifying, or executing new files and directories.

Set 17

1. Create a file named access_log.txt.

2. Set permissions:

- **user: rwx**
- **group: r--**
- **others: ---**

3. Change the owner to user logadmin and group opsTeam.

4. Add write permission to the group using symbolic mode.

5. **Create a directory named /exam_share.**

6. **Assign full permissions to user, group, and others (777).**

7. **Apply the Sticky Bit to protect files inside it.**

8. **Create a shell script named userCheck.sh.**

9. **Add content to display UID and GID of the logged-in user.**

10. **Apply SUID on the script.**

11. **Create the directory /teamID and apply SGID so files inherit the group.**

Set 18

1. **Create a user named *examuser* with a home directory and assign /bin/bash as the default shell.**

2. **Create a group named *teamA* and add *examuser* to that group.**

3. **Display all groups to which the user *examuser* belongs.**

4. **Create a directory /secure and give the owner full permissions, group read-only permissions, and no permissions to others.**

5. **Change the password of user examuser to “exampass”**

6. **Schedule a cron job that sends a message "System will reboot soon" into a file named /var/log/reboot_notice.log every Saturday at 11:45 PM.**

Set 19

1. **Create a user named operator with an initial password and force password change on first login.**

2. **Create a group named qaTeam with a GID of 46000.**

3. **Configure passwordless sudo for the qaTeam group, allowing only the command /usr/bin/yum through a file in /etc/sudoers.d/.**

4. **Create users qa1, qa2, qa3 and add them to qaTeam.**

5. Set account expiration for qa1, qa2, qa3 to 150 days from today.

6. Configure password aging:

qa1: must change password every 12 days

qa2: must change password every 18 days

7. Temporarily lock user qa3, then unlock the account.

8. Create a file labtask.txt, set permissions to user (rw), group (r), others (none), change ownership to user labuser and group qaTeam, and add execute permission for user and group.

Set 20

Write a Shell Script to perform the following operations:

1. Count the number of files with the .txt extension in the current directory. 2. Display the number of files modified in the last 24 hours.
3. Count how many files have read + write permissions for the owner.
4. Show the top five files consuming most disk space.
5. Display the newest directory created in the current location.
6. List all subdirectories under /home (only one level deep).
7. Display all files in current directory whose name contains "test".
8. List all files in the current directory which are not executable.
9. Display the total number of lines in all .sh files.
10. **Exit.**

Set 21

1. Create a file named project_data.txt.

2. Set permissions:

· user: read/write

· group: no permission

· others: read only

3. Change ownership of the file to user projuser and group devTeam.

4. Add execute permission to the user only using symbolic mode. 5.

Create a directory named /team_public.

6. Give read/write/execute permissions to user and group only (others: no permission).
7. Apply the Sticky Bit on /team_public and Create a script named showInfo.sh.

8. Add content to print the current username and home directory.

9. Apply SUID permission to the script.

10. Create a directory /sharedDev and apply SGID permission on it.

Set 22

1. Create a file named access_log.txt.

2. Set permissions:

user: rwx

group: r--

others: ---

3. Change the owner to user logadmin and group opsTeam.

4. Add write permission to the group using symbolic mode.

5. Create a directory named /exam_share.

6. Assign full permissions to user, group, and others (777).

7. Apply the Sticky Bit to protect files inside it.

8. Create a shell script named userCheck.sh.

9. Add content to display UID and GID of the logged-in user.

10. Apply SUID on the script.

11. Create the directory /teamID and apply SGID so files inherit the group.

Set 23

Write a Shell Script to perform the following Operations.

1. **No. of ordinary files in current directory.**
2. **No. of subdirectories in current directory.**
3. **No. of executable files in current directory.**
4. **Display five largest files in current directory.** 5. **Display last modified file in current directory.** 6. **Display all subdirectories in parent directory.** 7. **Display all subdirectories in current directory.** 8. **Display all executable files in current directory.** 9. **Display all files in current directory including hidden files.** 10. **Exit**