

## Linux Programming Assignment:6

NAME: SANDEEP

USN: ENG24CY0047

ROLL NO: 40

SECTION: B

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

- ls command

- **Examples :**

```
paul@debian8:~$ ls allfiles.txt dmesg.txt services stuff summer.txt
```

2. Write the **command to create a new directory named 123test\_dir**.

- mkdir 123test\_dir

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

- **sed** command: Stream editor for filtering and transforming text

- **Example:**

#Replace text

```
paul@debian5:~/pipes$ echo level5 | sed 's/5/42/' level42
```

```
paul@debian5:~/pipes$ echo level5 | sed 's/level/jump/' jump5
```

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

- whatis command

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

- touch notes.txt

6. Differentiate between **grep** and **awk** commands with an example.

**grep:**

- Global Regular Expression Print.
- Primarily for looking up text which matches a pattern.
- Simply prints the lines with the pattern.

Example:

```
paul@RHEL4b pipes]$ cat tennis.txtAmelie Mauresmo, FraKim Clijsters, BELJustine Henin, BelSerena Williams, usaVenus Williams, USA# Basic search paul@RHEL4b pipes]$ grep Williams tennis.txt Serena Williams, usa
```

Williams, usa

Venus Williams, USA



awk:

- A language for scanning and processing patterns.
- Can search like grep, but also manipulate text, extract fields, and take actions/calculations.
- Far more powerful for structured data such as CSV or logs.

Example: CSV file employees.csv:

John,Developer,5000

Jane,Manager,7000

Bob,Developer,5500

Alice,Designer,4500

**To calculate average salary by department:**

**awk -F',' '{dept[\$2]+=\$3; count[\$2]++} END {for (d in dept) print d, dept[d]/count[d]}' employees.csv**

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

**-chmod u+rx script.sh**

Or

**chmod 700 script.sh**

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

1. Purpose chown: Is used to change the owner of a file or directory.

Syntax: **chown new\_owner filename**

Example: **sudo chown manasa report.txt**

2. chgrp: Will only change the group ownership of a file or directory.

Syntax: **chgrp new\_group filename**

Example: **sudo chgrp students report.txt**

9. 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**?

**- Step 1: Check current user**

**whoami**

**Step 2: Check file permissions**

**ls -l filename**



Output displays permissions in format: -rwxr-xr--

- Position 1: File type
- Positions 2-4: User owner permissions (rwx)
- Positions 5-7: Group owner permissions (r-x)
- Positions 8-10: Others permissions (r--)

### Step 3: Add execute permission (Page 313)

chmod u+x filename# ORchmod +x filename **Full Troubleshooting Process:**

# 1. Self-identificationpaul@debian8:~\$ whoami paul# 2. Permissions check on filepaul@debian8:~\$ ls -l script.sh-rw-r--r-- 1 paul paul 0 Oct 15 10:30 script.sh# Observe: no 'x' (execute) permission for anyone# 3. Execute permission added for user ownerpaul@debian8:~\$ chmod u+x script.sh# 4. Change confirmedpaul@debian8:~\$ ls -l script.sh-rwxr--r-- 1 paul paul 0 Oct 15 10:30 script.sh# Now includes 'x' for user owner# 5. File executionpaul@debian8:~\$ ./script.sh"You have to explicitly do a chmod +x to make a file executable."

10. 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**.

- **Commands involved:**

- find
- tee

#### find command

# Find files newer than a referencefind. -newer file42.txt# Find files of specific typefind. -type f -name "\*.conf"

#### tee command

"The tee filter puts stdin on stdout and also into a file."

[paul@RHEL4b pipes]\$ tac count.txt | tee temp.txt | tacContent of count.txt appears

hereonetwothreefourfive\*\*Solution for the given problem:\*\*``bashfind /var/log -name "\*.log" -type f -mtime -2 | tee recent\_logs.txt

#### Explanation:

- find /var/log: Search in /var/log directory
- -name "\*.log": Find files ending with .log
- -type f: Only regular files
- -mtime -2: Modified in last 2 days
- | tee recent\_logs.txt: Display on screen AND save to file