Date 15/7/25

Expt. No. Page No. 1

1. Write the use of some basic linux commands.

File commands:

1. ls : Directory listing
2. cd dir : change directory to dir
3. cd : change to home directory
4. pwd : show current working directory
5. mkdir dir : Creating a directory dir
6. cat >file : Places the standard input into the file
7. more file : Output the contents of the file
8. touch file : Create or update file
9. rm file : Deleting the file
10. cp file1 file2 : Copy the contents of file1 to file2
11. mv file1 file2 : Rename or move file1 to file2, if file2 is an existing directory
12. ln -s file link : Create symbolic link link to file

Process management:

1. ps : To display the currently working processes
2. top : Display all running process
3. kill pid : Kill the process with given pid
4. killall proc : Kill all the process named proc
5. pkill pattern : Will kill all processes matching the pattern
6. bg : List stopped or background jobs, resume a stopped job in the background.

Teacher's Signature

7. fg: Brings the most recent job to foreground.
8. fg n: Brings job n to the foreground.

File permission:

1. chmod octal file: Change the permission of file to octal, which can be found separately for user, group, world by adding,
   - 4-read(r)
   - 2-write(w)
   - 1-execute(x)

Searching:

1. grep pattern file: Search for pattern in file
2. grep -r pattern dir: Search recursively for pattern in dir
3. command | grep pattern: Search pattern in the ouput of a command
4. locate file: Find all instances of file
5. find . -name filename: Searches in the current directory (represented by a period) and below it, for files and directories with names starting with filename.
6. pgrep pattern: Search for all the named processes that matches with the pattern and, by default, return their ID

System Info:

1. cal : Show this month's calender
2. uptime : Show current uptime
3. w : Display who is on line
4. finger user : Display information about user
5. uname -a : Show kernel information
6. man command : Show the manual for command.
7. free : Show memory and swap usage

Compression:

1. tar cf file.tar file : Create tar named file.tar containing file
2. tar xf file.tar : Extract the files from file.tar
3. gzip file : Compresses file and renames it to file.gz

Network:

1. ping host : Ping host and output results
2. dig domain : Get DNS information for domain
3. wget file : Download file

Shortcuts:

1. ctrl+c : Halts the current command
2. !! : Repeats the last command
3. exit : Logout the current session

Teacher's Signature

2. Write a report on the features of Unix and Linux commands.

Features of Unix and Linux commands:

Introduction:

Unix and Linux are powerful operating systems widely used in system administration, development, and server environments. A key aspect of their power lies in the command-line interface (CLI), which allows users to interact with the system using commands. Though Unix and Linux differ in their origin and licensing, they share many similar command-line features.

Main Features:

1. Command structure: Commands follow a simple structure:
command [options] [arguments]
Example: ls -l /home

2. Case Sensitivity: Commands, file names, and arguments are case-sensitive. For example, LS and ls are different.

3. Pipelining and Redirection:
Both systems allow output of one command to be used as input for another using pipes (|). Redirection symbols

like >, >>, and < are also supported.
Example : cat file.txt | grep "hello"

4. Shell environment: Unix and Linux commands run in a shell environment such as Bash, Ksh, or Zsh. Shells allow command execution, scripting, and environment customisation.

5. Powerful File Management: Commands like ls, cp, mv, rm, find, and touch provide complete control over files and directories.

6. User and Permission Management: Commands such as chmod, chown, passwd, who and id help in managing users, groups, and permissions.

7. Process Management: You can manage system processes using commands like ps, top, kill, nice and jobs.

8. Networking Tools: Commands like ping, netstat, ifconfig, scp, and ssh allow monitoring and managing network connections.

9. Package Management: Linux supports package management through tools like apt, yum, dnf, and zypper. Unix systems may use different tools based on the vendor.

10. Scripting Support: Shell scripting allows automation of tasks using .sh files. Scripts can include loops, conditions, and functions.

11. Manual and Help: The man and --help commands provide built-in documentation. Example: man ls

12. Custom commands and Aliases: Users can define their own commands or set aliases for frequently used commands.
Example: alias ll='ls -l'

Conclusion:

Unix and Linux commands offer a rich set of features for system management, programming, and automation. Their efficiency, flexibility, and scripting capabilities make them essential tools for developers, sysadmins, and power users. While unix is mostly commercial and Linux is open-source, both share a strong foundation in command-line utilities.

3. Write a report with example of shell script.

Introduction to Shell Scripting in Unix and Linux:

Shell scripting is a powerful feature of Unix and Linux that allows users to automate tasks by writing a series of commands in a script file. These scripts are executed by the shell (commonly Bash) and can be used for file manipulation, program execution, user management, system monitoring and more.

Features of shell scripting:

1. Automation of Tasks: Scripts can automate repititive tasks like backups, updates, and log analysis.
2. Conditional Execution: Shell scripts support if, else, elif statements to perform conditional logic.
3. Loops: Scripts can use loops (for, while, until) to repeat actions.
4. Variables: You can define and use variables to store values dynamically.
5. User Interaction: Shell scripts can take input from users using read.
6. Error Handling: Scripts can handle errors using conditional statements or special variables like $?

Example Shell Script :

This script takes a file name as input and checks if the file exists.

```
#!/bin/bash
echo "Enter the filename:"
read filename
if [ -f "$filename" ]; then
    echo "File '$filename' exists."
    echo "Displaying contents:"
    cat "$filename"
else
    echo "File '$filename' does not exist."
fi
```

Explanation:

- #!/bin/bash indicates that the script should be run using the Bash shell.
- read takes user input.
- if [-f "$filename"] checks if the file exists.
- cat "$filename" displays the content of the file.

To run this script:

1. Save it as check_file.sh
2. Give it executable permission using chmod +x check_file.sh
3. Run it using ./check_file.sh

Conclusion:

Shell scripting is an essential skill in Unix and Linux environments. It enables users to automate complex tasks efficiently and manage system operations effectively. With simple syntax and flexibility, shell scripts enhance productivity and system control.

Submitted by:
Alpana Mohanty
2302081053
IT-2

Teacher's Signature