**Linux Commands**

Experiment No : 1

Aim :- To study basic Linux commands

Theory :

Linux Commands

1. date: Display the current date and time.
2. tput clear: Clear the terminal screen.
3. cal: Display the calendar for the current month.
4. cal year: Display the calendar for the entire year.
5. cal <month> <year>: Display the calendar for a specific month and year.
6. who: Show who is logged into the system.
7. whoami: Display the current user's username.
8. ps: List currently running processes.
9. ls: List files and directories in the current directory.

* ls -l: List files and directories in long format.
* ls > filename : redirects the output of the "ls" command, which lists files and directories in the current directory, to a specified filename, creating or overwriting the file if it exists.

1. cat file\_name : Display the contents of a file
2. cat >filename: Create or overwrite a file's content.
3. cat >>filename: Append to a file's content.
4. wc file\_name: Count lines, words, and characters in a file.
5. uname: Display system information.
6. tty: Display the terminal file name.
7. pwd: Print the current working directory.
8. echo “Text”: Print text to the terminal.
9. printf: Format and print text.

* printf "Hello, %s!\n" “name”

1. bc: Command-line calculator.
2. passwd: Change user password.
3. stty: used to control and configure terminal settings
4. rm file: Remove files or directories.
5. mv : Move or rename files or directories.

* mv <source> <destination>
* mv <old\_name> <new\_name>

1. cmp: Compare two files byte by byte.

* cmp <file1> <file2>

1. comm: Compare two sorted files line by line.

* comm <file1> <file2>

1. diff: Display differences between two files.

* diff <file1> <file2>

1. mkdir: Create a new directory.

* mkdir <directory\_name>

1. rmdir: Remove an empty directory.

* rmdir <directory\_name>

1. cd <directory>: Change the current directory.
2. cd : without any attributes is used to change the working directory to the user's home directory.
3. cd .. : Move to the parent directory.
4. tree: Display directory structure as a tree.
5. chmod: Change file permissions.

* chmod <permissions> <file\_name>

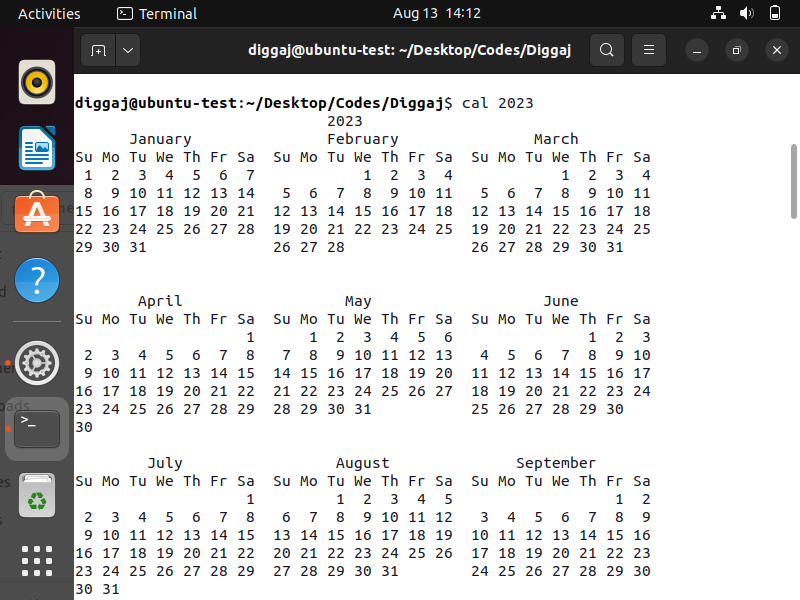
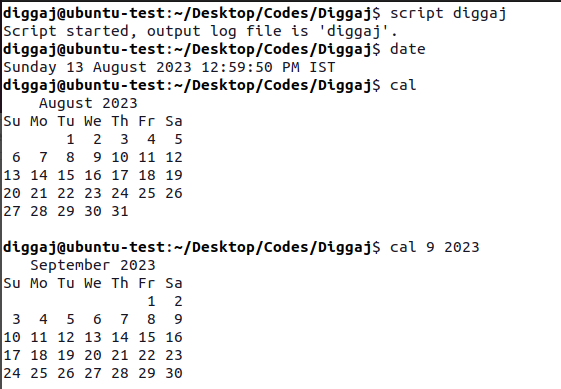
1. Relative Permissions (Symbolic): The chmod command with symbolic notation modifies file permissions based on user, group, and others using symbols like + (add), - (remove), and = (set).

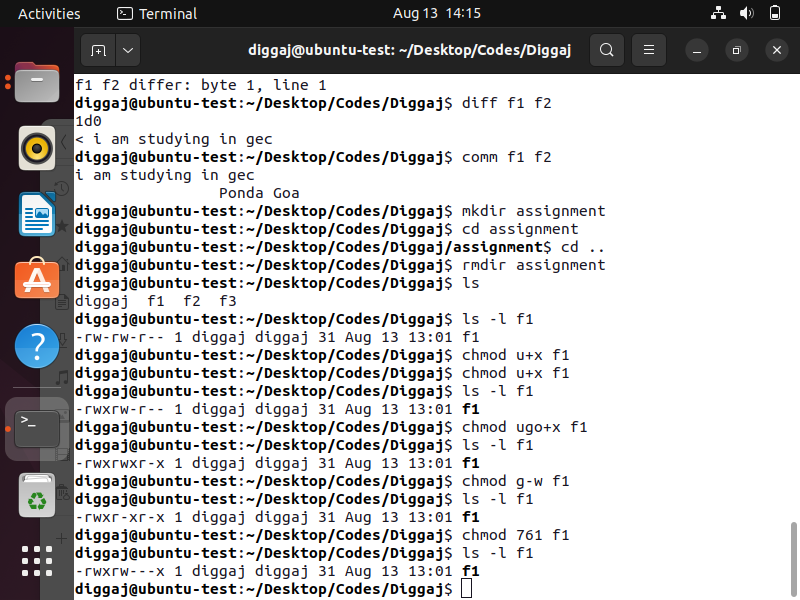
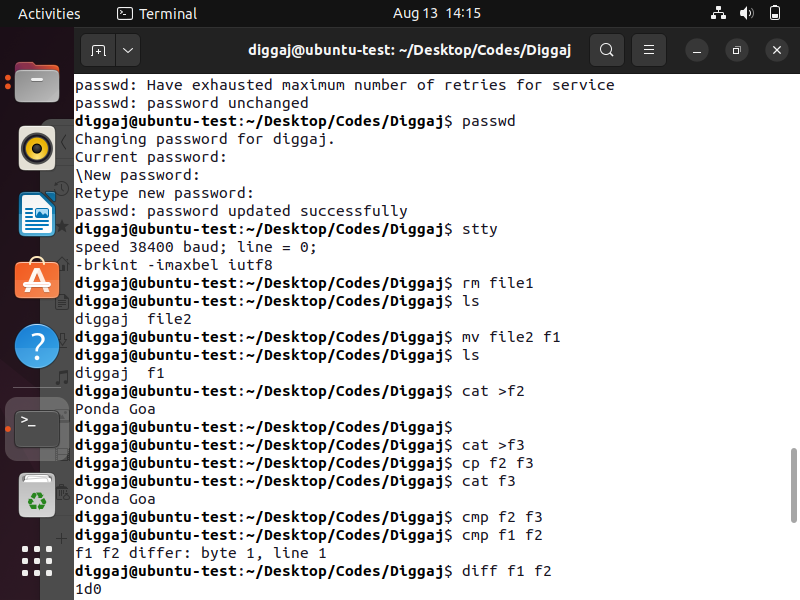
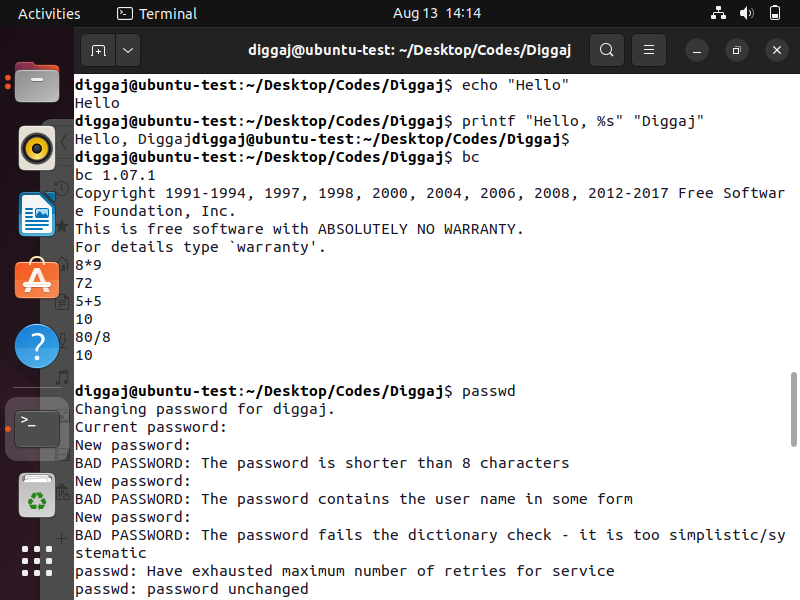
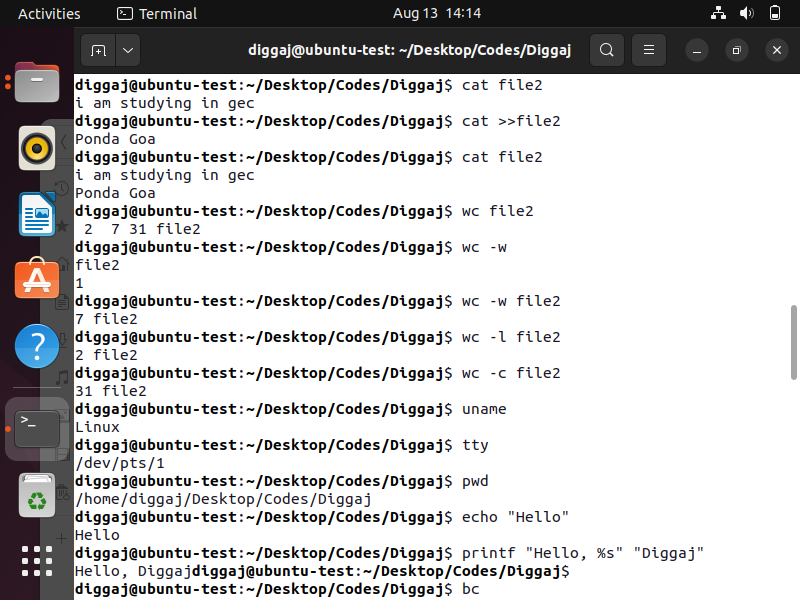
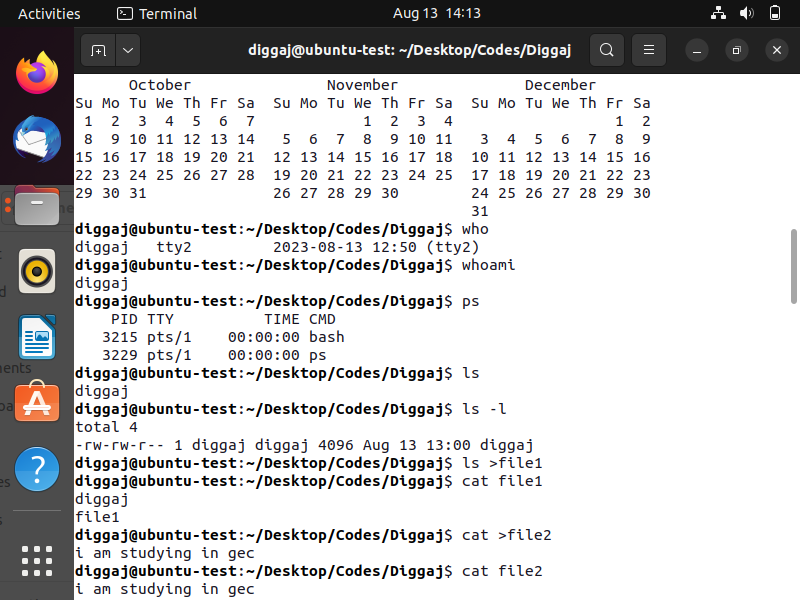
* chmod u+x file.txt (Add execute permission for the user)
* chmod go-rw file.txt (Remove read and write permissions for group and others)
* chmod a=rwx file.txt (Set read, write, and execute permissions for all)

1. Absolute Permissions (Numeric): The chmod command with numeric notation assigns permissions using a three-digit octal code (e.g., 755) where each digit corresponds to user, group, and others, representing read (4), write (2), and execute (1).

* chmod 644 file.txt (Read and write for user, read for group and others)
* chmod 755 script.sh (Read, write, and execute for user, read and execute for group and others)
* chmod 600 private.txt (Read and write for user, no permissions for others)

Output :





**Conclusion** : Basic Linux commands were studied and implemented successfully.