|  |
| --- |
| Experiment No.1 |
| Explore the internal and external commands of Linux. |
| Date of Performance: 12/01/2024 |
| Date of Submission: 16/02/2024 |

**Aim:** Explore the internal and external commands of Linux.

**Objective:** The Linux command is a utility of the Linux operating system. All basic and advanced tasks can be done by executing commands.

**Theory:**

**They are grouped into two categories:**

* **Internal Commands :** Commands which are built into the shell. For all the shell built-in commands, execution of the same is fast in the sense that the shell doesn’t have to search the given path for them in the PATH variable, and also no process needs to be spawned for executing it.  
  Examples: source, cd, fg, etc.
* **External Commands :** Commands which aren’t built into the shell. When an external command has to be executed, the shell looks for its path given in the PATH variable, and also a new process has to be spawned and the command gets executed. They are usually located in /bin or /usr/bin. For example, when you execute the “cat” command, which usually is at /usr/bin, the executable /usr/bin/cat gets executed.  
  Examples: ls, cat etc.

**pwd** : It gives absolute path to your current location i.e. current working directory.

napster@napster-Veriton-Series:~$ pwd

/home/napster

**mkdir:** It creates new directory/ folder.

napster@napster-Veriton-Series:~$ mkdir TRIAL

**cd:** It is used to change directory.

napster@napster-Veriton-Series:~$ cd TRIAL

**cd ..** : To come back to previous directory.

**cd \**  : Return to root directory.

**touch:** To create new files.

napster@napster-Veriton-Series:~/TRIAL$ touch hello.txt

            To verify hello.txt is created or not:

napster@napster-Veriton-Series:~/TRIAL$ ls

Hello.txt

6) **ls:** It lists the contents of files and directories.

napster@napster-Veriton-Series:~/TRIAL$ ls

hello.txt  TRIAL2

**cat :** 1) It can also use to create new file with content as shown below.

cat> hello.txt

hello from Shamika

2) It can also use to append the data into existing file as shown below.

cat>>hello.txt

How are you?

3) It is used to concatenate files.

napster@napster-Veriton-Series:~/TRIAL$ cat hello.txt hi.txt

hello from Shamika

How are you?

Welcome to Os Lab.

**mv:**  To rename a file from source to destination and To move file from one location to other location.

napster@napster-Veriton-Series:~/TRIAL$ mv hi.txt how.txt

**grep:** It searches all text files in the current directory for lines containing “hello”

napster@napster-Veriton-Series:~/TRIAL$ grep hello \*.txt

hello from Shamika

**rm:** remove / delete files.

napster@napster-Veriton-Series:~/TRIAL/TRIAL2$ rm how.txt

**date:** Print or set the system date and time ,Display the current time in the given

FORMAT, or set the system date.

**b1@comp:~$ date**

Fri Feb 16 15:43:44 IST 2018

**time:** Displays time of the system.

**free:** Shows amount of RAM  In use.

**echo:** Echoes output on the screen.

**clear:** Clears the screen.

**exit:** Exit from the terminal.

**man :** (man commandname) Gives  description about  the command.

**gedit:** To open text editor.

**ps :**  Report a snapshot of the current processes. ps displays information about a selection of the active processes.

**cal** : Displays a calendar.

**wc** **:** print newline, word, and byte counts for each file, Print newline, word, and byte counts for each FILE, and a total line if more than one FILE is specified.

**chmod** - change file mode bits

chmod changes the file mode bits of each given file according to mode, which can be

either a symbolic representation of changes to make, or an octal number

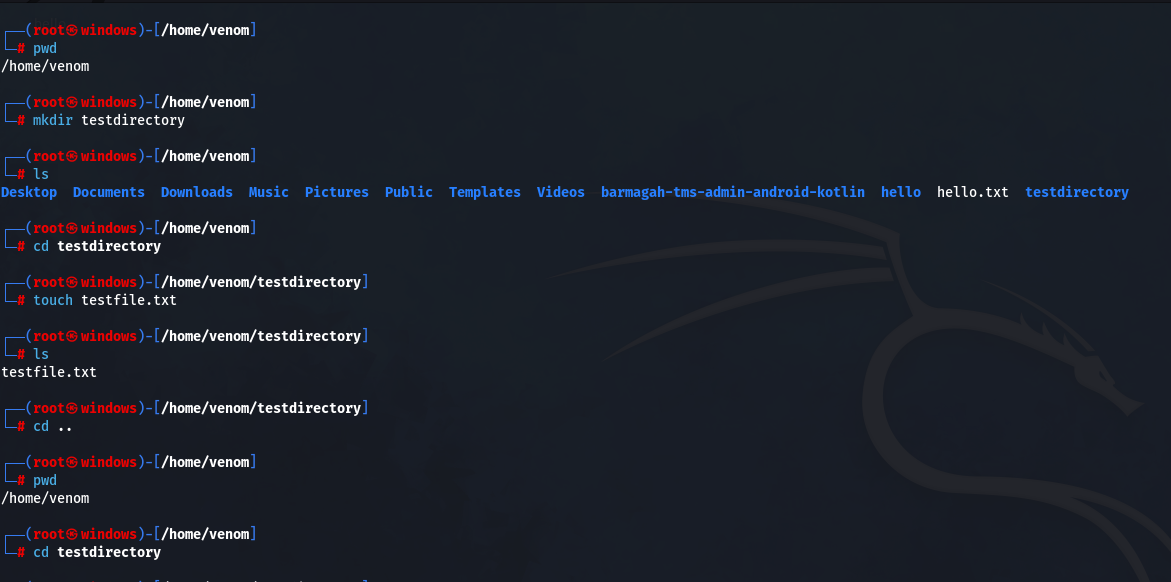
representing the bit pattern for the new mode bits.

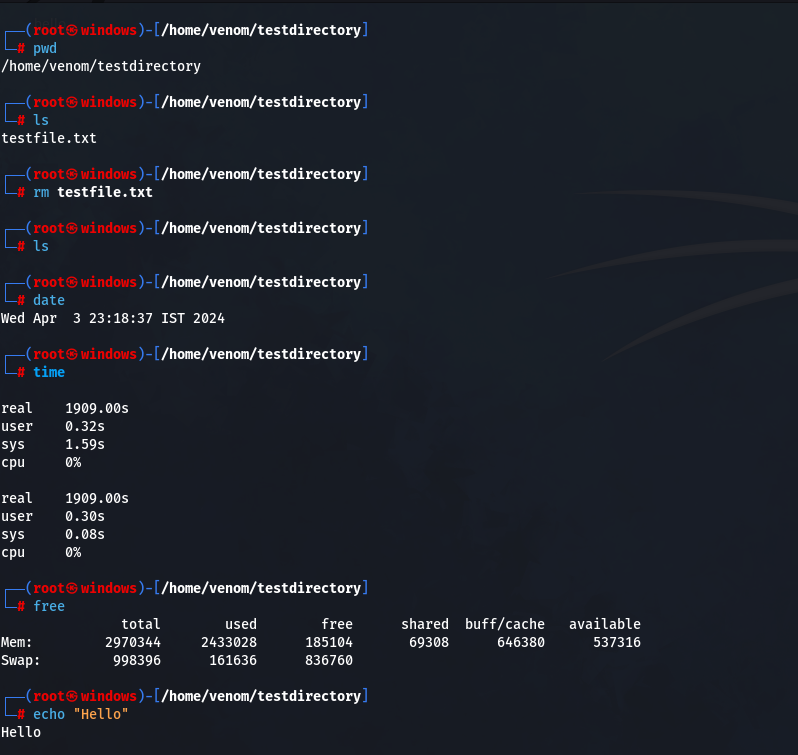
**chown** - change file owner and group

chown changes the user and/or group ownership of each given file. If only an owner (a

user name or numeric user ID) is given, that user is made the owner of each given file, and the files' group is not changed. If the owner is followed by a colon and a group name (or numeric group ID), with no spaces between them, the group ownership of the files is changed as well.

**Output:**





**Conclusion**:

The basic difference between internal and external commands in Linux lies in their execution location and handling by the shell. Internal commands are built-in functionalities of the shell itself, executed directly within the shell's process without invoking external programs. Examples include cd and echo. Conversely, external commands are standalone programs stored in separate executable files, located in directories listed in the PATH variable. These commands are executed by spawning new processes, such as ls and grep. Understanding this distinction is crucial for efficient command execution and shell operation in Linux environments.