S.B. JAIN INSTITUTE OF TECHNOLOGY MANAGEMENT & RESEARCH, NAGPUR

Practical 02

Aim: To understand and demonstrate the use of basic commands in different operating systems (Windows, Linux, and UNIX) for managing files, directories, permissions, and user interactions through a terminal or command-line interface.

Name: Prachita Ashok Mahure

Roll No:CM24031

Semester / Year: III / 2nd

Academic Session: 2026-2027

Date of Performance:

Date of submission:

❖ **Aim:** To understand and demonstrate the use of basic commands in different operating systems (Windows, Linux, and UNIX) for managing files, directories, permissions, and user interactions through a terminal or command-line interface.

❖ Objectives:

1. To learn and practice fundamental command-line operations for file and directory management.
2. To explore and utilize user and permission management commands effectively.
3. To enhance system administration skills by working with commands across different operating systems.

❖ Requirements:

Hardware Requirements:

- **Processor**: Multi-core CPU, Intel Core i3 (3.0 GHz) or higher
- **RAM**: Minimum 4 GB (8 GB recommended for optimal performance)
- **Storage**: 100 GB HDD or SSD (Solid State Drive) for faster access
- **Network Interface**: Ethernet or Wi-Fi adapter for connectivity

Software Requirements:

- **Operating System**: Windows 10/11, Linux (Ubuntu 20.04/CentOS 8), UNIX-based OS
- **Command-line Interface**: PowerShell or Command Prompt (Windows), Terminal (Linux/UNIX)
- **Text Editor**: Nano, Vim, or Visual Studio Code for file editing
- **Administrative Privileges**: Superuser (Linux/UNIX) or Administrator (Windows) access

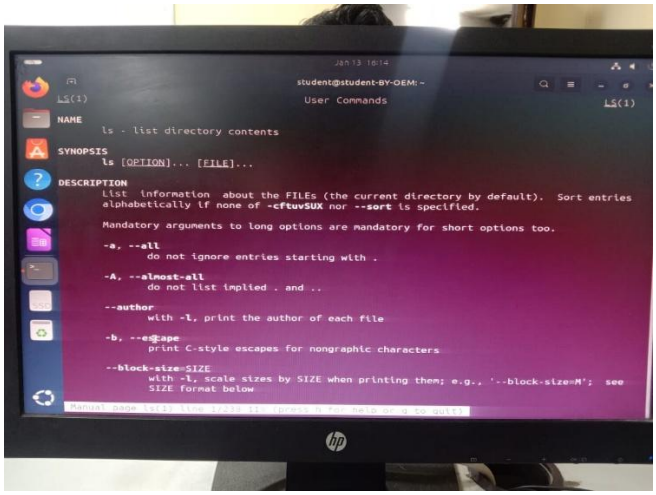
❖ Theory:

Basic commands in operating systems such as Windows, Linux, and UNIX are used to interact with the system through a command-line interface (CLI) or terminal. These commands allow users to manage files, directories, system processes, permissions, and user interactions efficiently.
In Windows, the Command Prompt and PowerShell provide commands like dir, cd, copy, del, and cls for file and directory management.
In Linux and UNIX, the terminal uses commands such as ls, pwd, cd, cp, mv, rm, chmod, and chown to perform similar tasks.
Using command-line tools is faster and more powerful than graphical interfaces, especially for system administration, troubleshooting, and automation. These commands help users understand the internal working of an operating system and improve control over system resources.

❖ Commands:

1. **Display User Manual of a Command**

- Functionality: Shows the manual page with details about a command's usage, options, and arguments.
- Syntax: man <command>
- Example: man ls

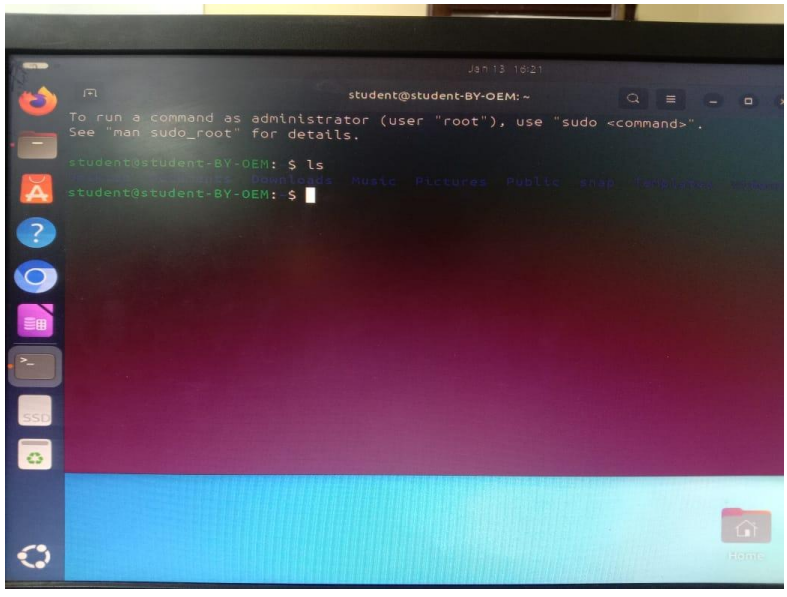


2. **Change Current Working Directory**.

- Functionality: Changes the terminal's current working directory.
- Syntax: cd <directory-path>
- Example: cd /home/user/Documents.

3. **List Contents of the Current Directory.**

- Functionality: Lists all files and directories in the current location.
- Syntax: ls
- Example: ls



4. **Read/Modify/Concatenate Text Files.**

- Functionality: Displays or manipulates file content.

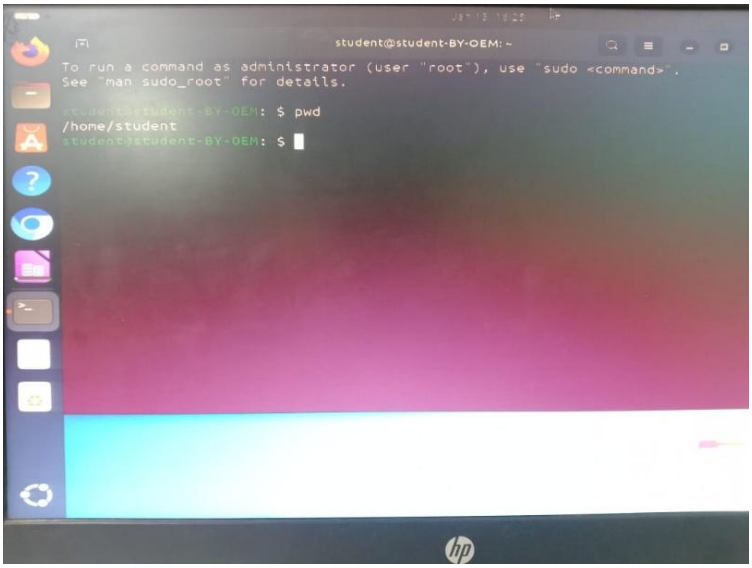
- Syntax:
  - Read: cat <filename>
  - Modify: ‘nano <filename>
  - Concatenate: cat <file1> <file2> > <outputfile>

5. **Create a New Directory**.

- Functionality: Creates a new directory at the specified path.
- Syntax: mkdir <directory-name
- Example: mkdir newdir

6. **Display Current Working Directory.**

- Functionality: Prints the current directory path.
- Syntax: pwd
- Example: pwd



7. **Write Arguments to Standard Output.**

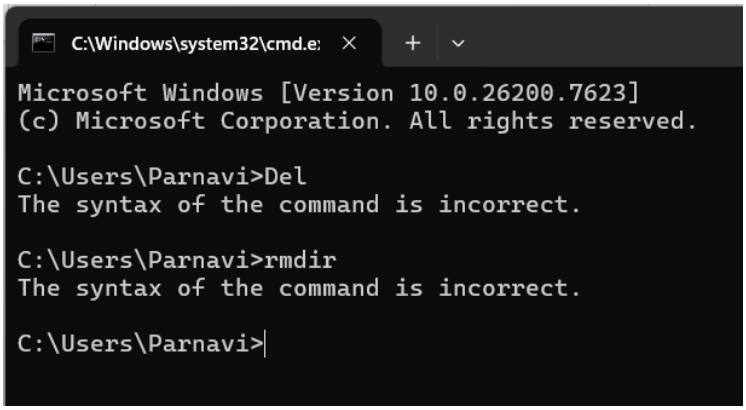
- Functionality: Prints the provided string or variables.
- Syntax: echo <arguments>
- Example: echo Hello World

8. **Remove a File**.

- Functionality: Deletes a specified file.
- Syntax: rm <filename>
- Example: rm file.txt

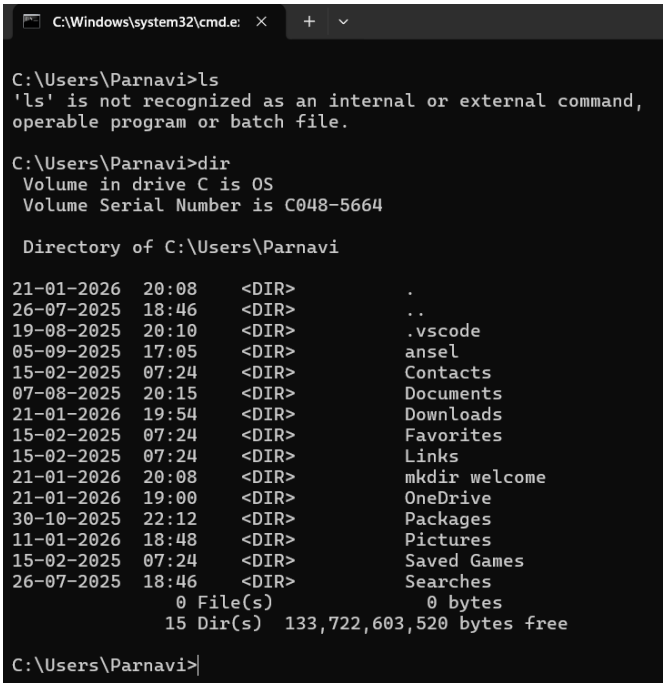
9. **Delete a Directory.**

- Functionality: Removes an empty directory.
- Syntax: rmdir <directory-name>
- Example: rmdir olddir



10. **Copy a File or Directory.**

- Functionality: Copies a file or directory to a destination.
- Syntax: cp <source> <destination>
- Example: cp file.txt backup/



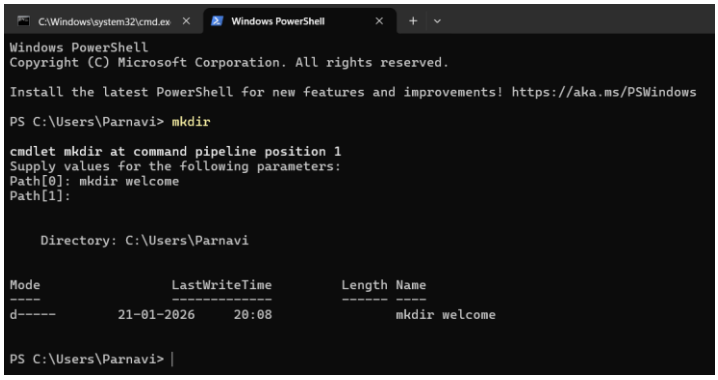
11. **Move Files or Directories.**

- Functionality: Moves or renames files and directories.
-Syntax: mv <source> <destination>

- Example: mv file.txt newdir/

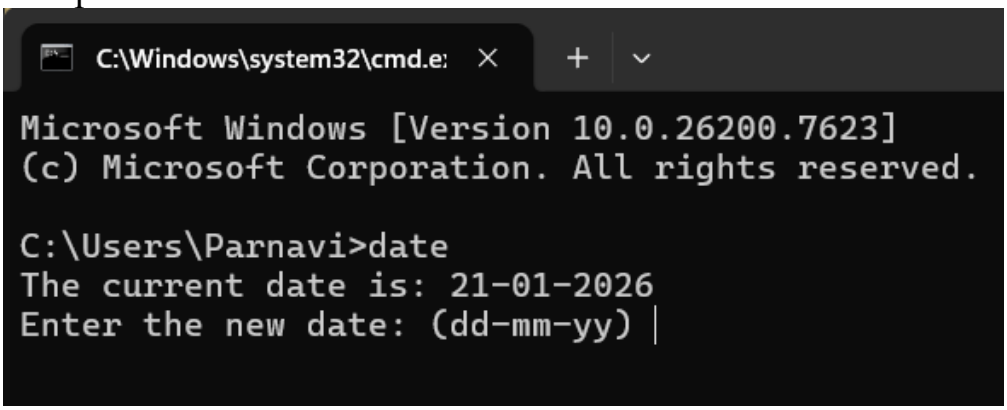
12. **Search for a String in a File.**
    - Functionality: Searches for a specific word or pattern in a file.
    - Syntax: grep "<string>" <file>
    - Example: grep "error" log.txt

13. **Create a Dictionary.**
    - Functionality: Creates a new directory (folder).
    - Syntax**:** mkdir directory_name
    - Example**:** mkdir myFolder

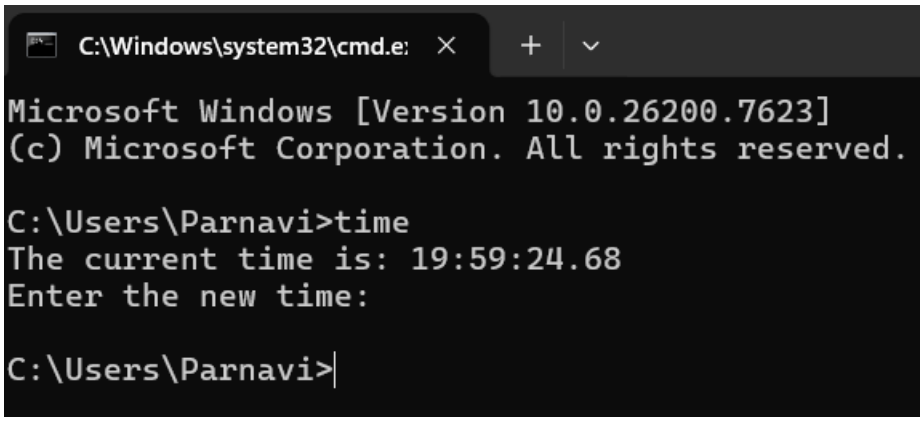


14. **Print Last N Lines of a File.**
    - Functionality: Displays the last N lines of a file.
    - Syntax: tail -n <N> <file>
    - Example: ‘tail -n 10 file.txt

15.**Display Current Date.**
    - Functionality: Displays the current system date.
    - Syntax: date +%d-%m-%Y
    - Example: date +%d-%m-%Y



16.**Display Current Time.**
    - Functionality: Displays the current system time.
    - Syntax: date +%H:%M:%S
    - Example: date +%H:%M:%S



15. **Remove All Permissions from All Users.**
    - Functionality: Clears all permissions on a file.
    - Syntax: 'chmod a-rwx <filename>
    - Example: 'chmod a-rwx file.txt

16. **Remove Read Permission Using Absolute Mode.**
    - Functionality: Uses numeric mode to restrict read access.
    - Syntax: chmod 700 <filename>
    - Example: chmod 700 file.txt

17. **Set R/W for Owner, None for Group/Other.**
    - Functionality: Assigns permissions in numeric mode.
    - Syntax: chmod 600 <filename>
    - Example: chmod 600 file.txt'

18. **Add Execute for Owner, Read for Group/Others.**
    - Functionality: Adds execution and read access.
    - Syntax: chmod u+x,g+r,o+r <filename>
    - Example: chmod u+x,g+r,o+r file.txt

19. **Add Execute Permission to All Users.**
    - Functionality: Enables execution by everyone.
    - Syntax: chmod a+x <filename>
    - Example: chmod a+x script.sh

❖ Conclusion:

The use of basic commands in Windows, Linux, and UNIX was successfully understood and demonstrated. This experiment enhanced practical knowledge of file handling, directory navigation, permission management, and user interaction through the command-line interface. Learning these commands is essential for efficient system usage, administration, and problem-solving in different operating environments.

❖ **Discussion Questions:**

1. **What is the significance of the pwd command in a Linux environment?**
2. **Explain the function of the cp command and its common options.**
3. **How does chmod 700 affect file permissions, and what does each digit represent?**
4. **Describe the difference between head and tail commands in Linux.**
5. **What is the purpose of the grep command, and how is it used with regular expressions?**

❖ **References**:

https://ubuntu.com/tutorials/command-line-for-beginners#1-overview
https://www.geeksforgeeks.org/25-basic-ubuntu-commands/

Date: / /2026

Signature
Course Coordinator
B.Tech CSE(AIML)
Sem: 4 / 2025-26