



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: Samyak Ukey

USN: CD24042

Semester / Year: 4th Sem / 2nd Year

Academic Session: 2025-26

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

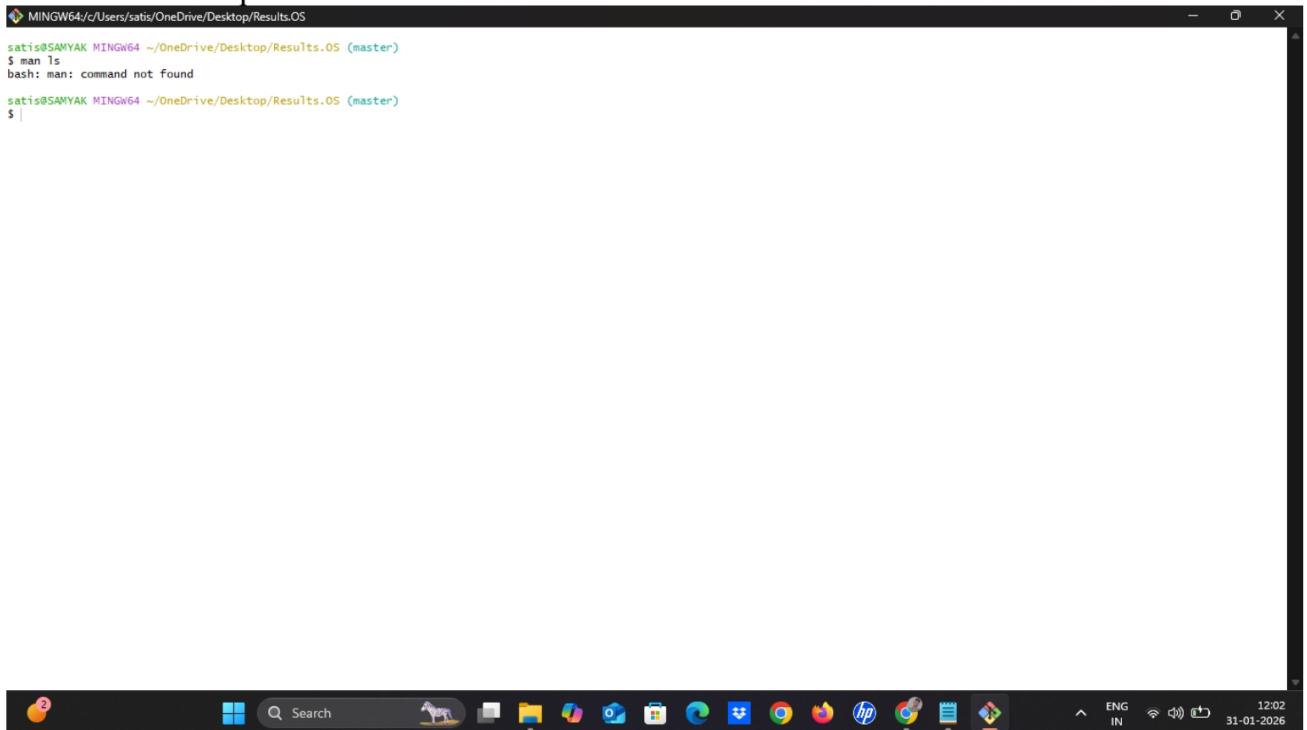
❖ **Theory:**

In system administration, command-line interfaces (CLI) are essential tools for managing and interacting with operating systems like Windows, Linux, and UNIX. Commands allow users to perform various tasks such as navigating directories, managing files, controlling permissions, and monitoring system performance. Each operating system provides a set of built-in commands, such as ‘man’, ‘ls’, ‘cd’, ‘mkdir’, and ‘chmod’, to facilitate efficient system management. Understanding these commands and their syntax is crucial for automating tasks, enhancing security, and ensuring optimal system functionality. This practical aims to develop foundational skills in executing and applying basic commands across different platforms.

❖ **Commands:**

1. Display User Manual of a Command

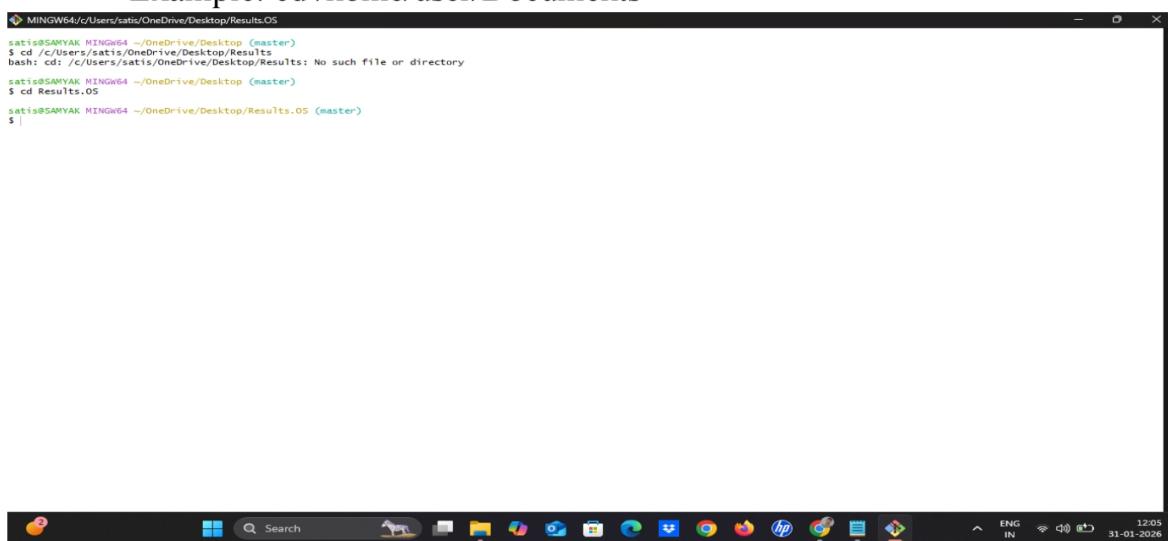
- Functionality: Shows the manual page with details about a command's usage, options, and arguments.
- Syntax: man <command>
- Example: man ls



```
MINGW64/c/Users/satis/OneDrive/Desktop/Results.OS
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$ man ls
bash: man: command not found
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$ |
```

2. Change Current Working Directory.

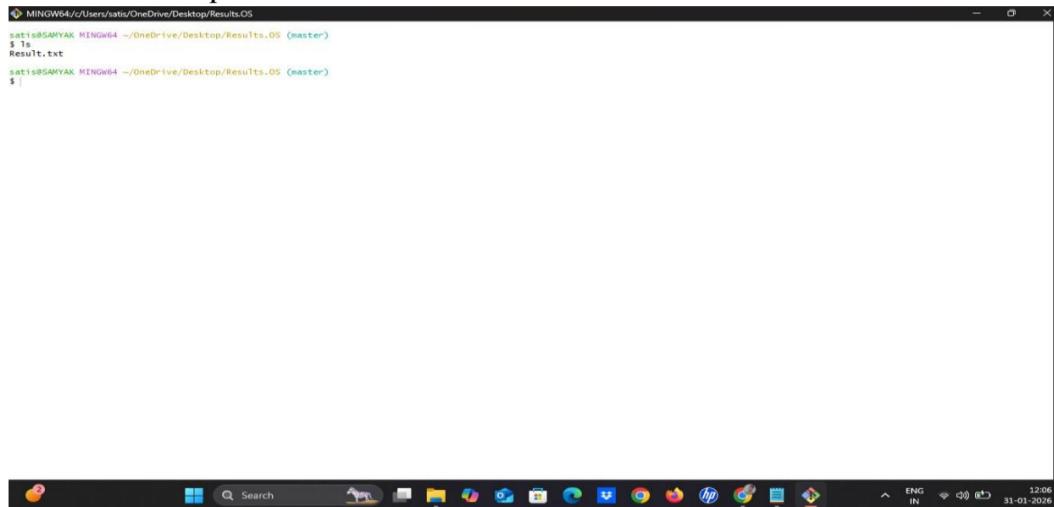
- Functionality: Changes the terminal's current working directory.
- Syntax: cd <directory-path>
- Example: cd /home/user/Documents



```
MINGW64/c/Users/satis/OneDrive/Desktop/Results.OS
satis@SAMYAK MINGW64 ~/OneDrive/Desktop (master)
$ cd /c/Users/satis/OneDrive/Desktop/Results
bash: cd: /c/Users/satis/OneDrive/Desktop/Results: No such file or directory
satis@SAMYAK MINGW64 ~/OneDrive/Desktop (master)
$ cd Results.OS
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$ |
```

3. List Contents of the Current Directory.

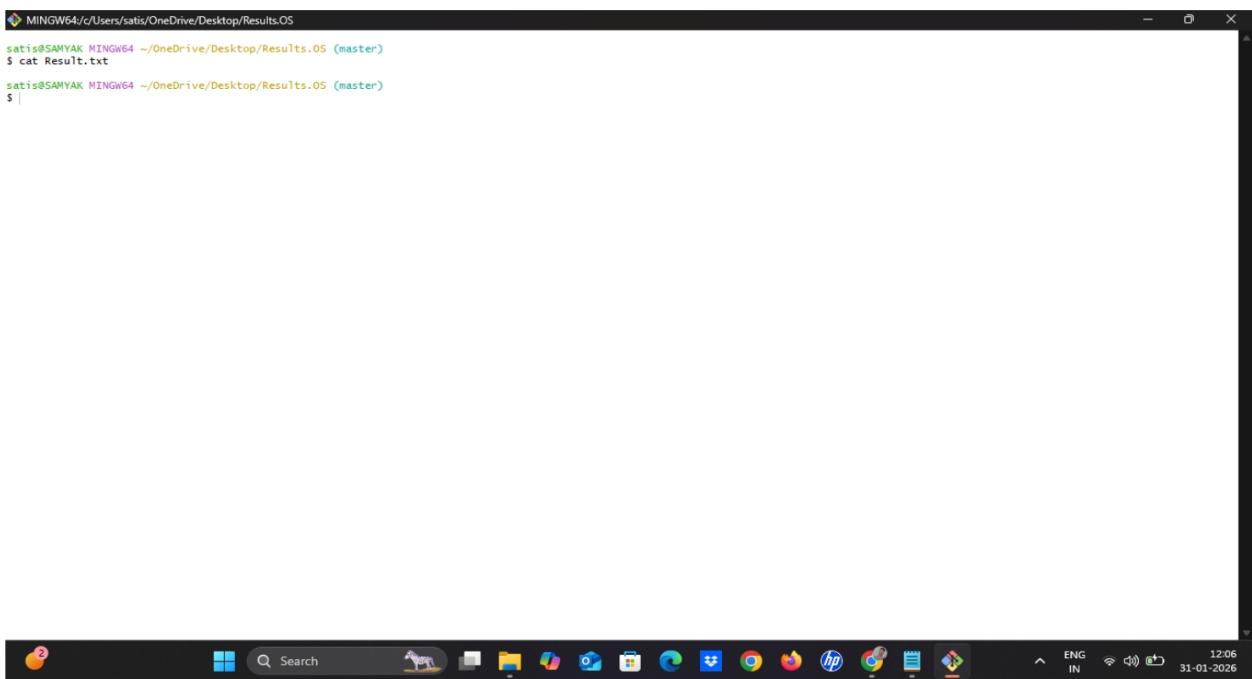
- Functionality: Lists all files and directories in the current location.
- Syntax: ls
- Example: ls



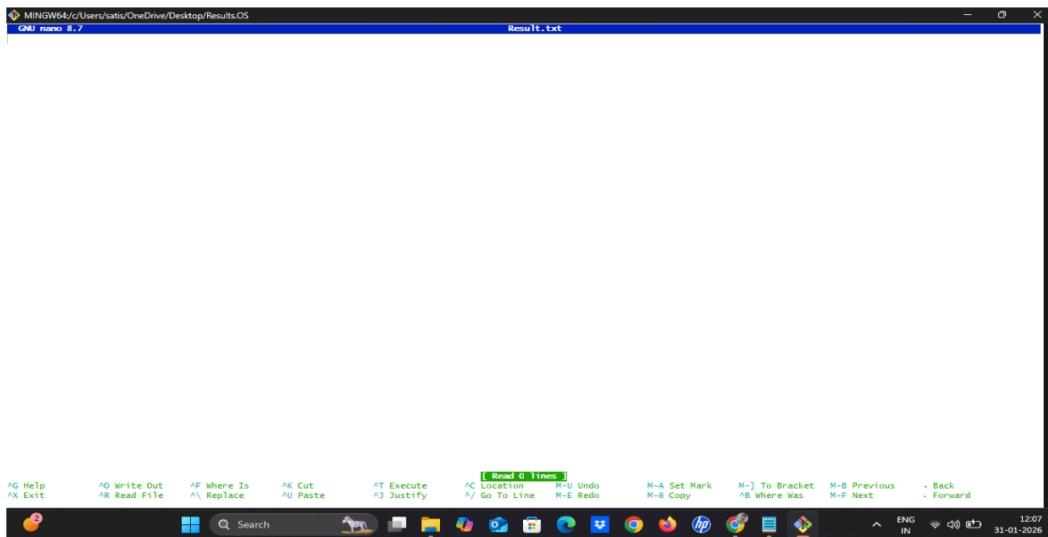
```
MINGW64/c/Users/satis/OneDrive/Desktop/Results.OS
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$ ls
Result.txt
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$ |
```

4. Read/Modify/Concatenate Text Files.

- Functionality: Displays or manipulates file content.
- Syntax:
 - Read: cat <filename>
 - Modify: 'nano <filename>
 - Concatenate: cat <file1> <file2> > <outputfile>

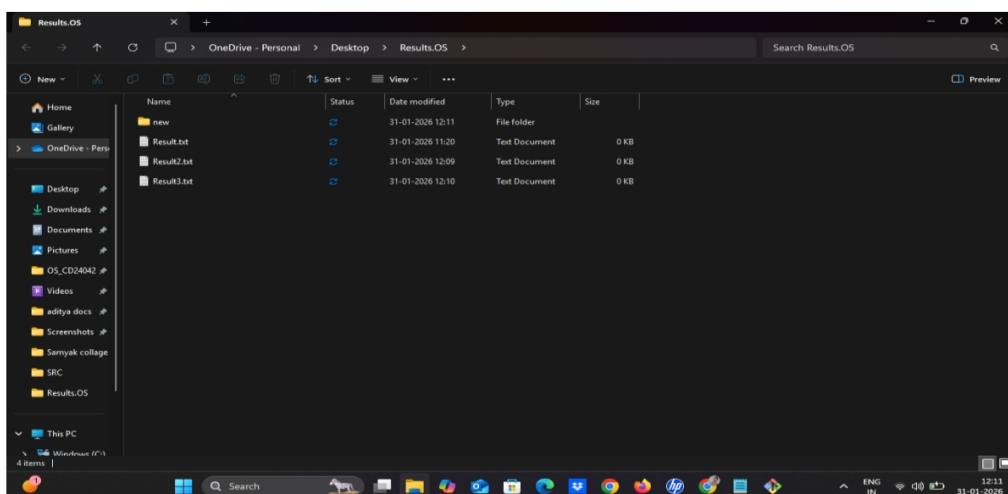
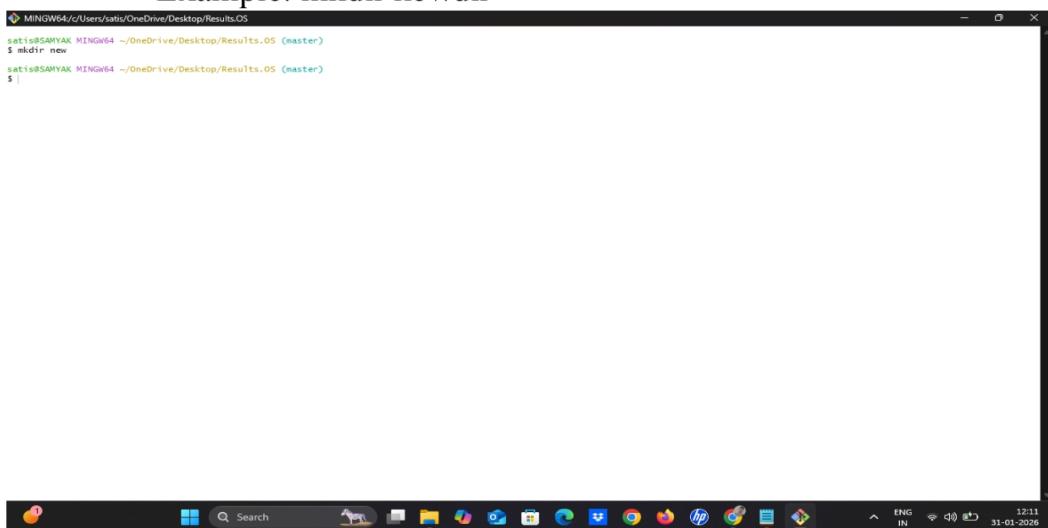


```
MINGW64/c/Users/satis/OneDrive/Desktop/Results.OS
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$ cat Result.txt
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$ |
```



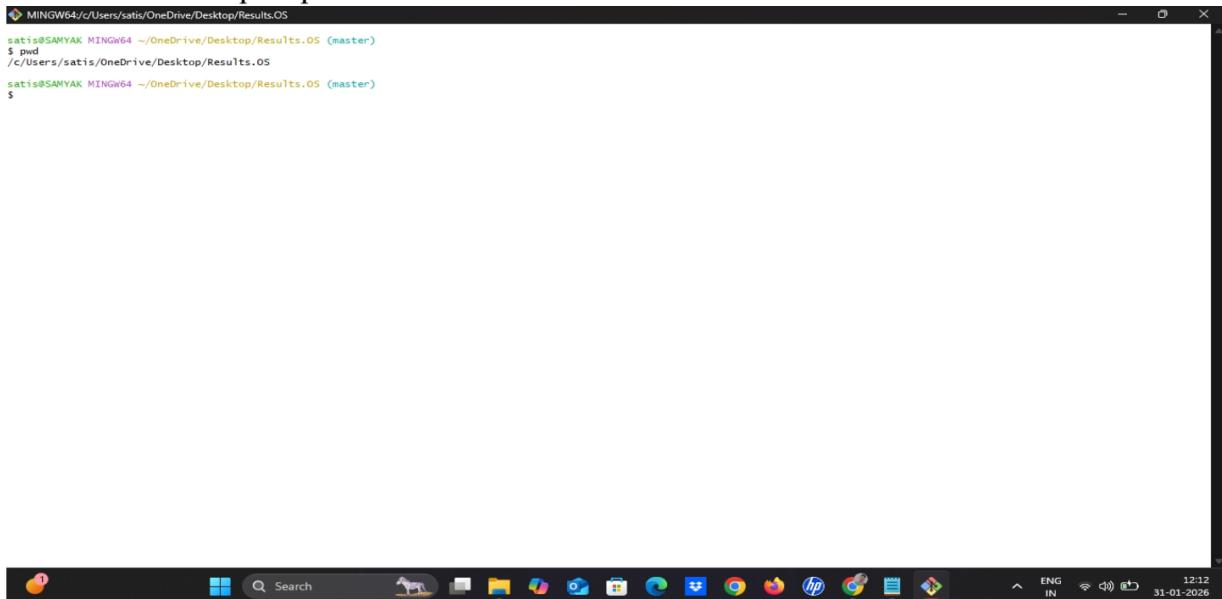
5. Create a New Directory.

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



6. Display Current Working Directory.

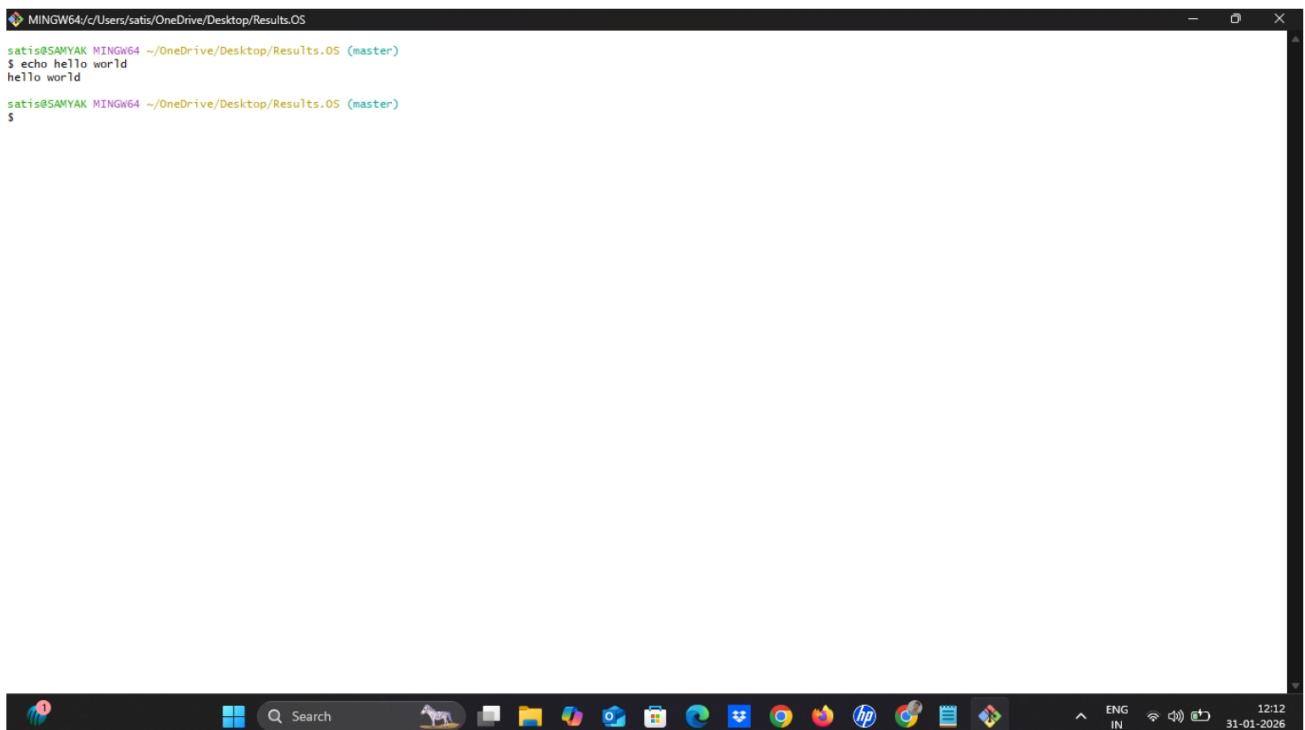
- Functionality: Prints the current directory path.
- Syntax: pwd
- Example: pwd



```
MINGW64:/c/Users/satis/OneDrive/Desktop/Results.OS
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$ pwd
/c/Users/satis/OneDrive/Desktop/Results.OS
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$
```

7. Write Arguments to Standard Output.

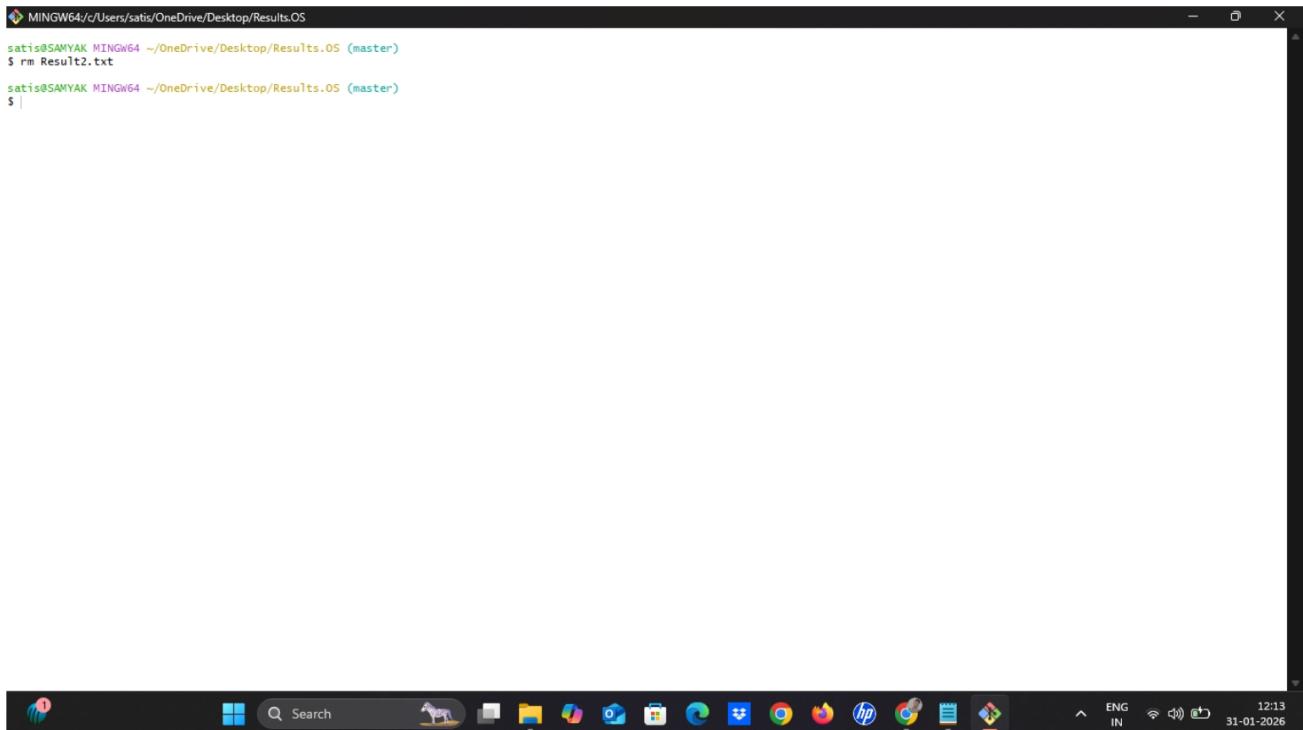
- Functionality: Prints the provided string or variables.
- Syntax: echo <arguments>
- Example: echo Hello World



```
MINGW64:/c/Users/satis/OneDrive/Desktop/Results.OS
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$ echo hello world
hello world
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$
```

8. Remove a File.

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

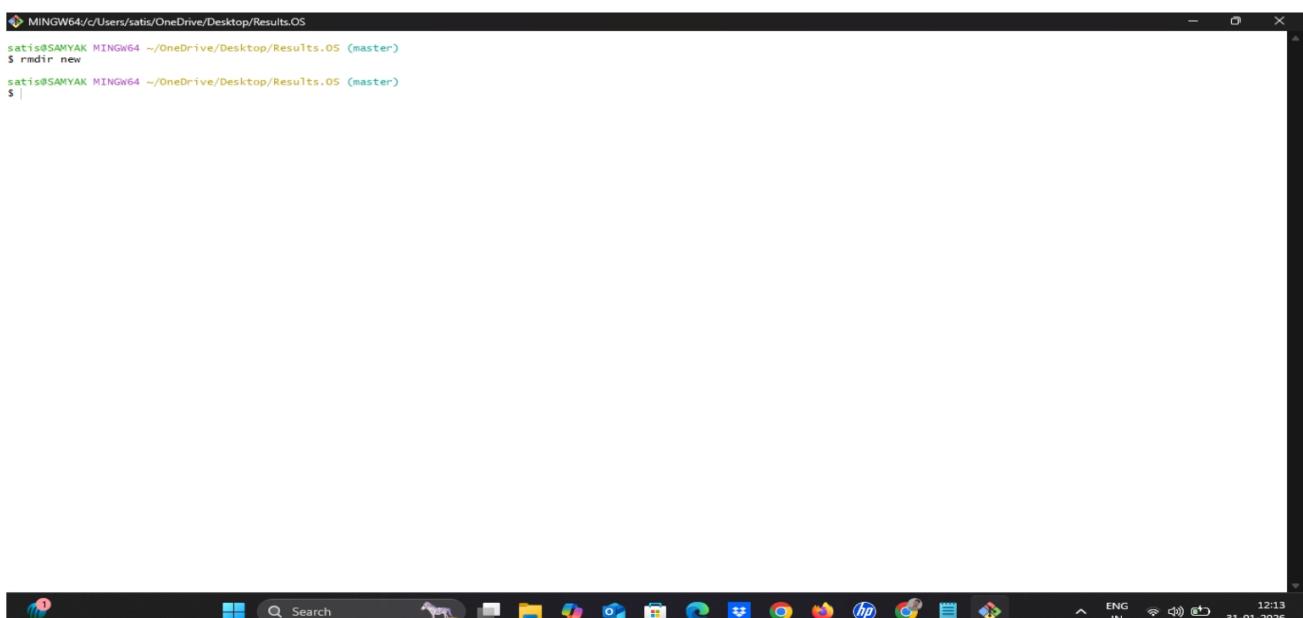


```
MINGW64:/c/Users/satis/OneDrive/Desktop/Results.OS
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$ rm Result2.txt

satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$ |
```

9. Delete a Directory.

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

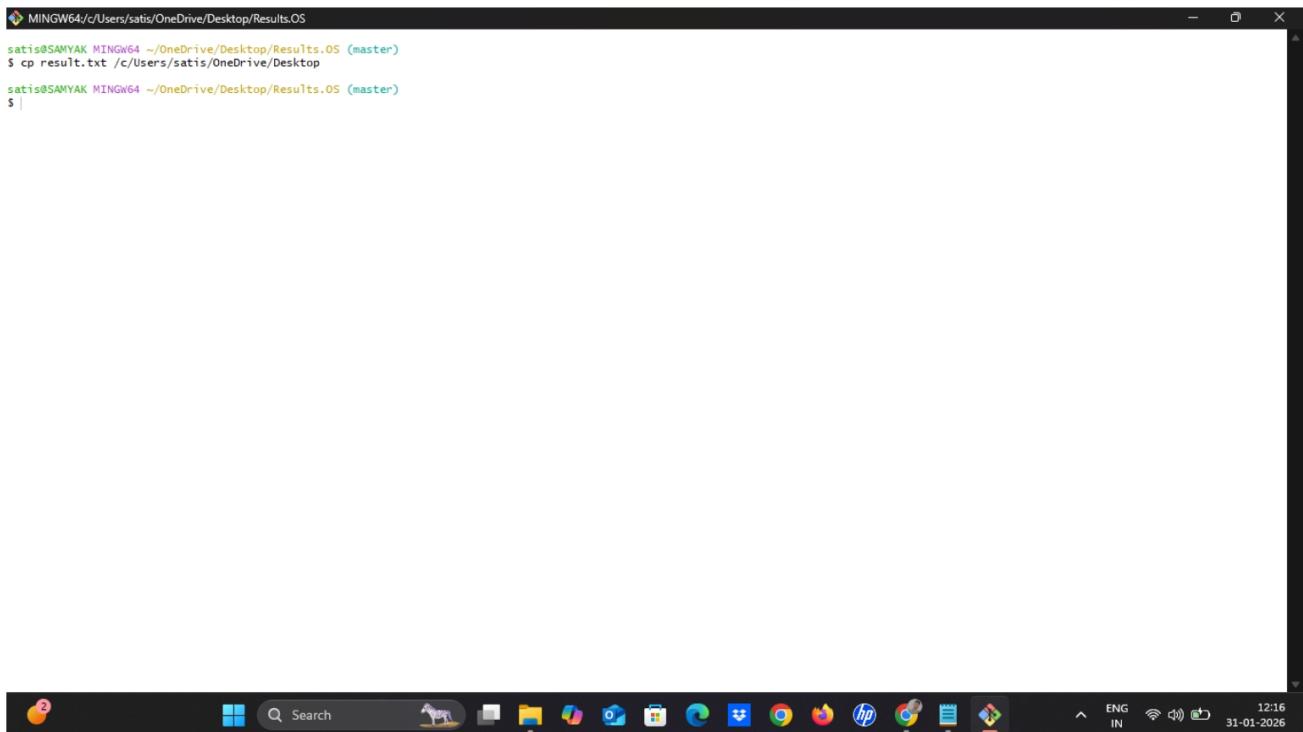


```
MINGW64:/c/Users/satis/OneDrive/Desktop/Results.OS
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$ rmdir new

satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$ |
```

10. Copy a File or Directory.

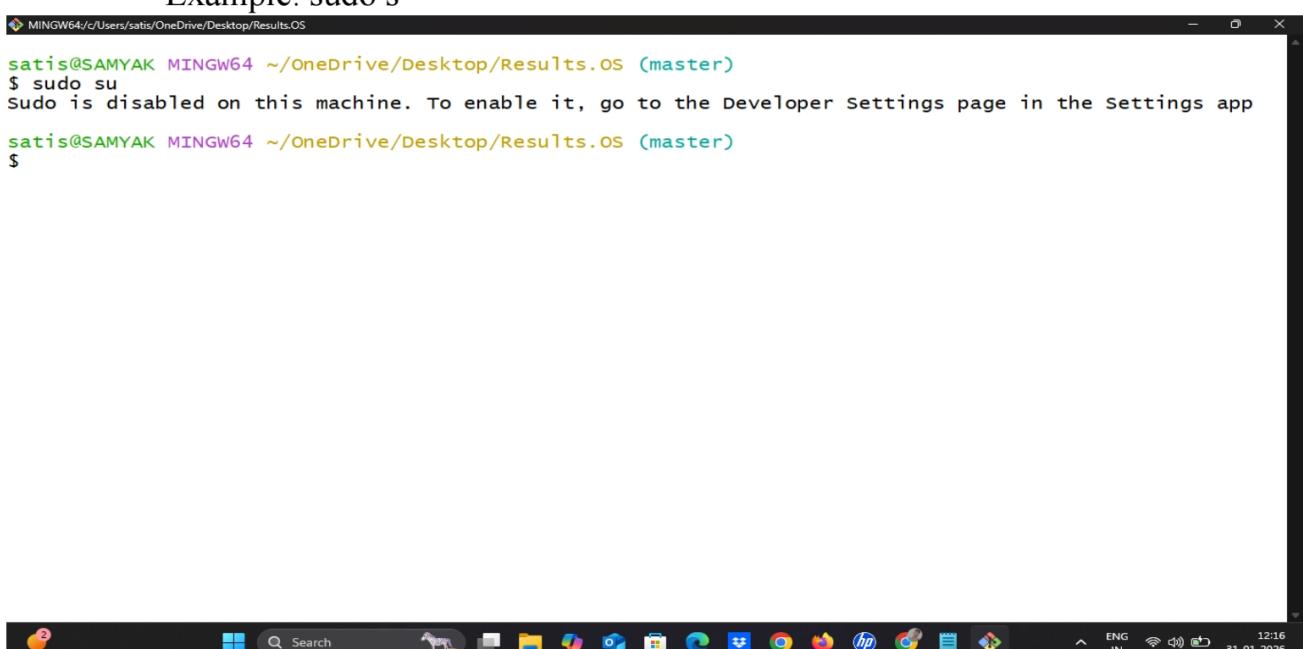
- Functionality: Copies a file or directory to a destination.
- Syntax: cp <source> <destination>
- Example: cp file.txt backup/



```
MINGW64:/c/Users/satis/OneDrive/Desktop/Results.OS
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$ cp result.txt ./Users/satis/OneDrive/Desktop
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$ |
```

11. Switch to Root User.

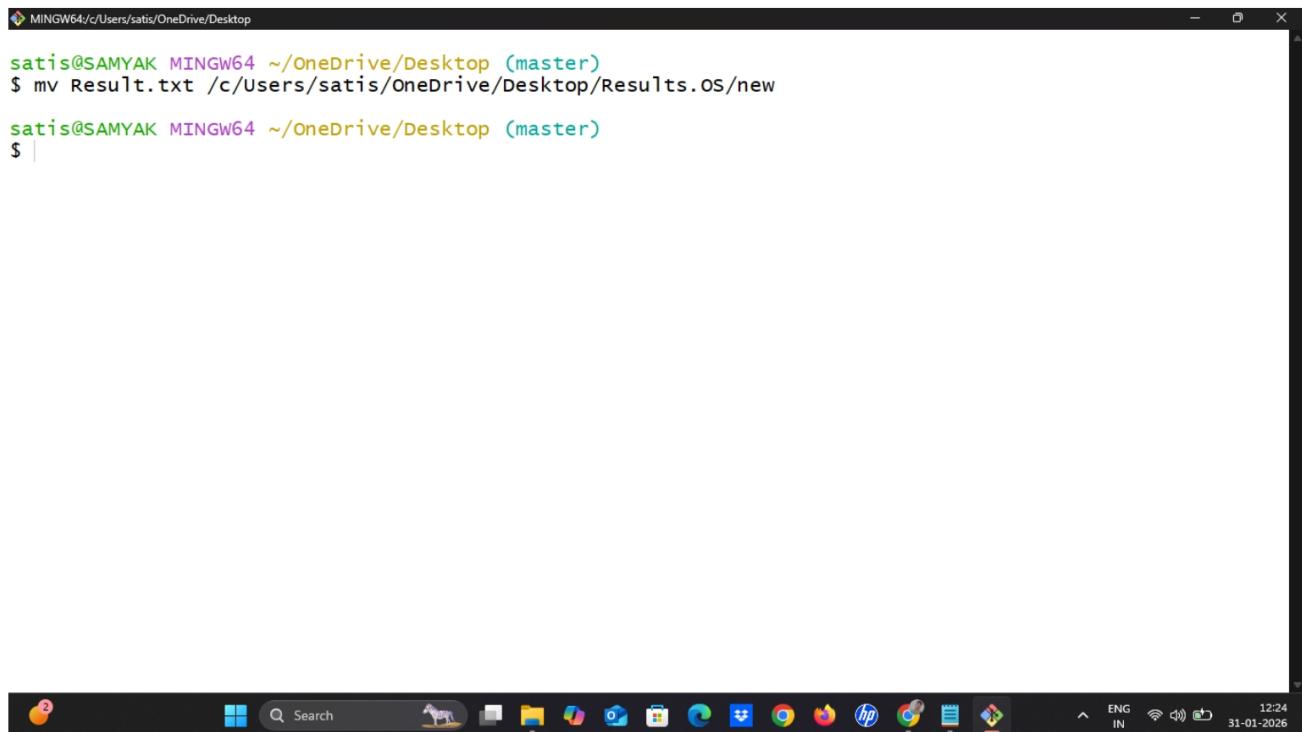
- Functionality: Gains root privileges temporarily.
- Syntax: sudo su
- Example: sudo s



```
MINGW64:/c/Users/satis/OneDrive/Desktop/Results.OS
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$ sudo su
Sudo is disabled on this machine. To enable it, go to the Developer Settings page in the Settings app
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS (master)
$
```

12. Move Files or Directories.

- Functionality: Moves or renames files and directories.
- Syntax: mv <source> <destination>
- Example: mv file.txt newdir/

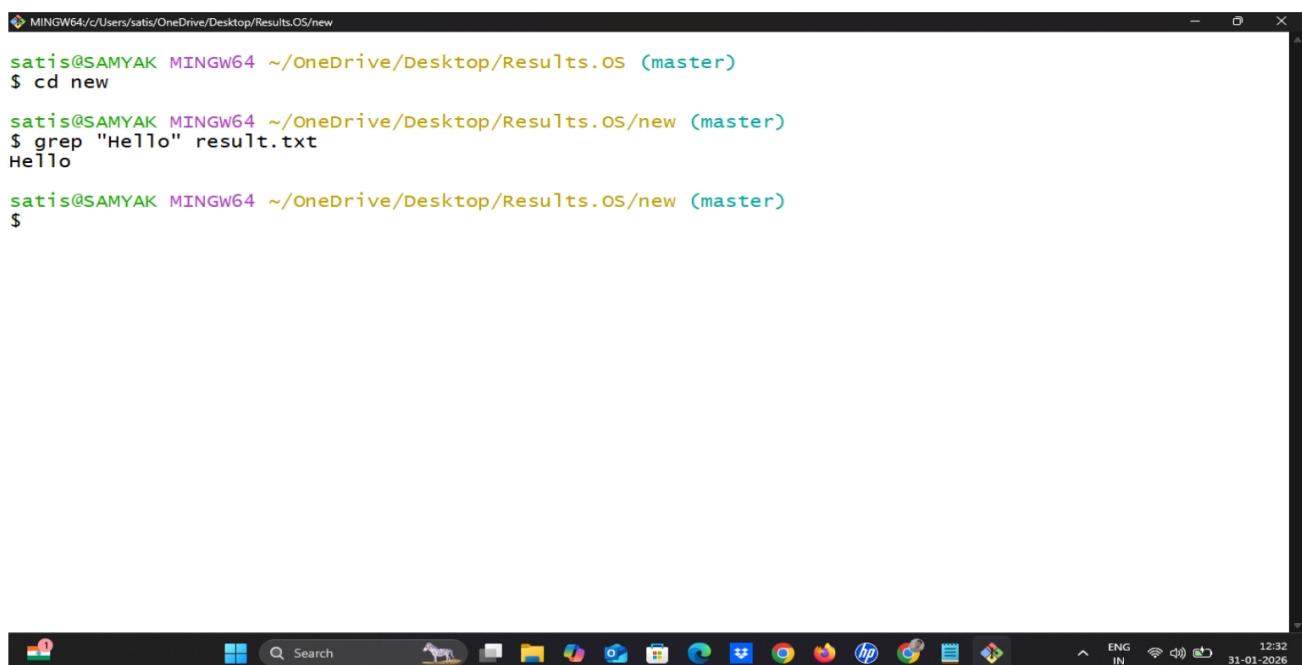


```
satis@SAMYAK MINGW64 ~/OneDrive/Desktop (master)
$ mv Result.txt /c/Users/satis/OneDrive/Desktop/Results.os/new

satis@SAMYAK MINGW64 ~/OneDrive/Desktop (master)
$ |
```

13. 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



```
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new
$ cd new

satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$ grep "Hello" result.txt
Hello

satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$
```

14. Print Top N Lines of a File.

- Functionality: Displays the first N lines of a file.
- Syntax: head -n <N> <file>
- Example: ‘head -n 10 file.txt’

```
MINGW64/c/User/satis/OneDrive/Desktop/Results.OS/new
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$ head -n 5 result.txt
Hello
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$ |
```

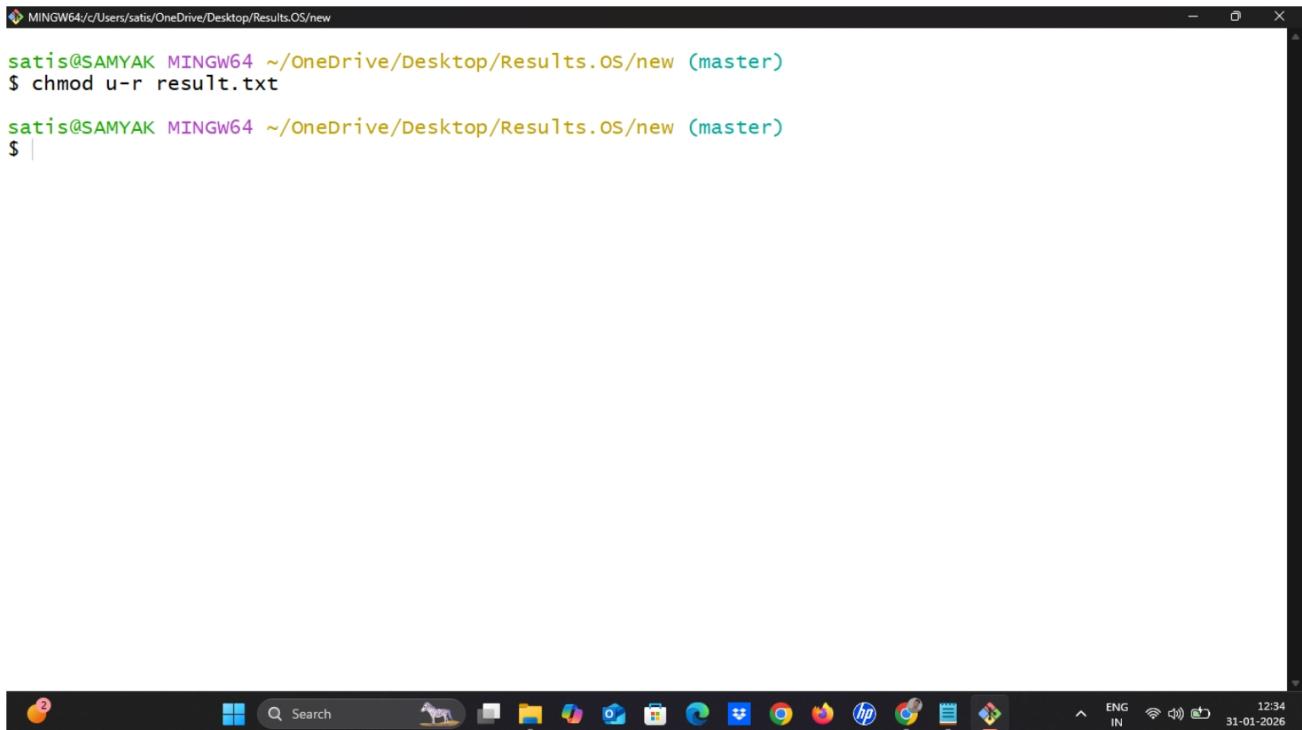
15. Print Last N Lines of a File.

- Functionality: Displays the last N lines of a file.
- Syntax: tail -n <N> <file>
- Example: ‘tail -n 10 file.txt’

```
MINGW64/c/Users/satis/OneDrive/Desktop/Results.OS/new
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$ tail -n 5 result.txt
Hello
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$ |
```

16. Remove Read Permission from Owner.

- Functionality: Revokes the owner's read permission for a file.
- Syntax: chmod u-r <filename>
- Example: chmod u-r file.txt

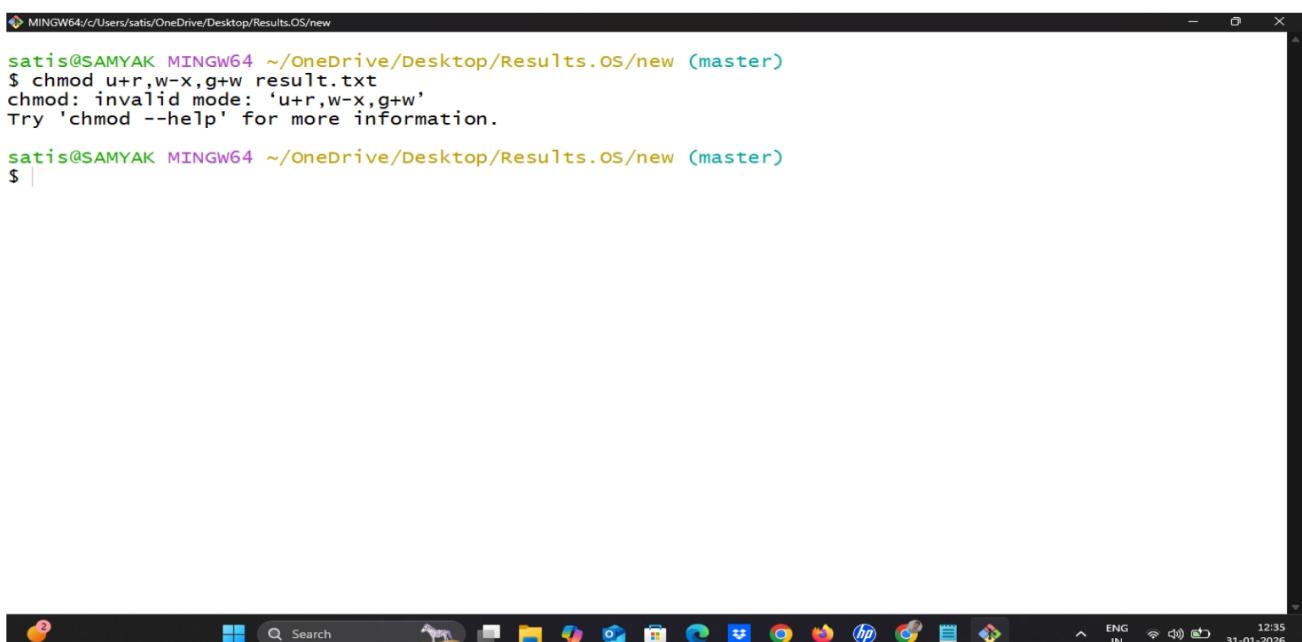


```
MINGW64/c/Users/satis/OneDrive/Desktop/Results.OS/new
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$ chmod u-r result.txt

satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$ |
```

17. Change Specific Permissions.

- Functionality: Sets or removes specific file permissions.
- Syntax: chmod u+r,w-x,g+w <filename>
- Example: chmod u+r,w-x,g+w file.txt

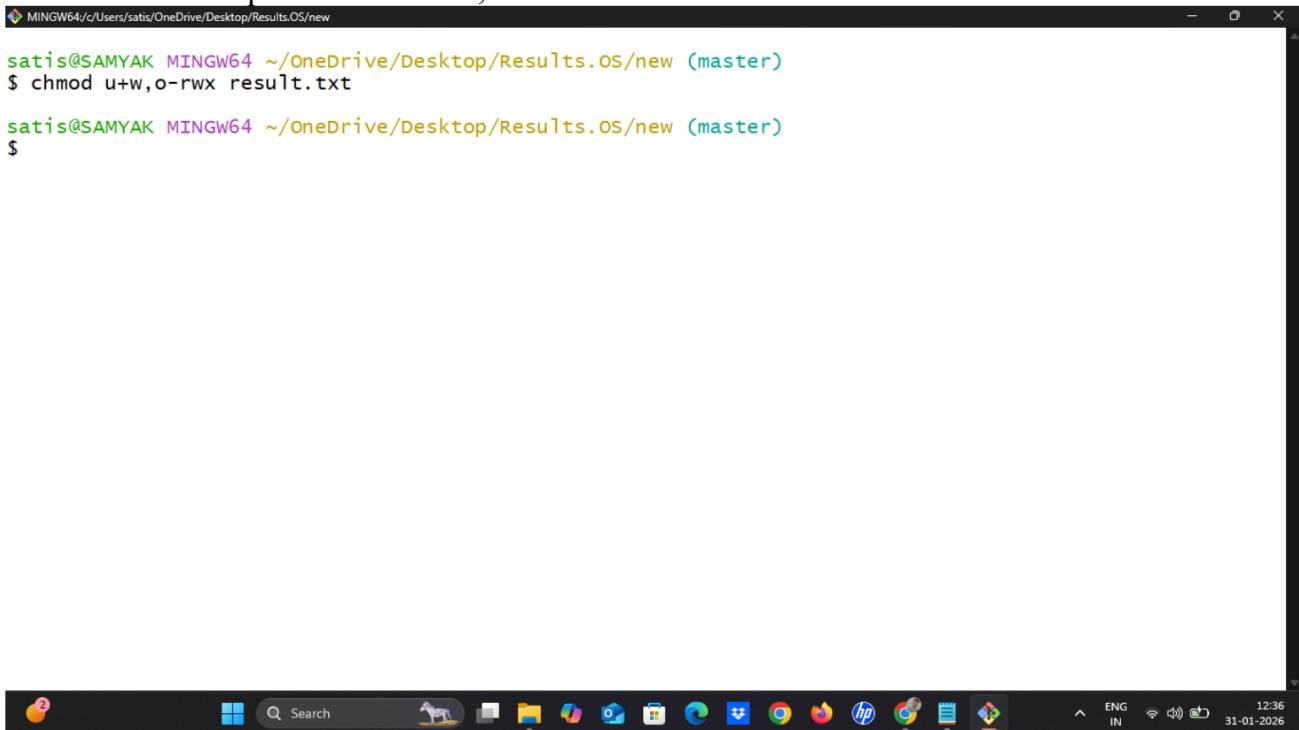


```
MINGW64/c/Users/satis/OneDrive/Desktop/Results.OS/new
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$ chmod u+r,w-x,g+w result.txt
chmod: invalid mode: 'u+r,w-x,g+w'
Try 'chmod --help' for more information.

satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$ |
```

18. Add Write Permission to Owner, None to Others.

- Functionality: Allows write access for the owner only.
- Syntax: chmod u+w,o-rwx <filename>
- Example: chmod u+w,o-rwx file.txt

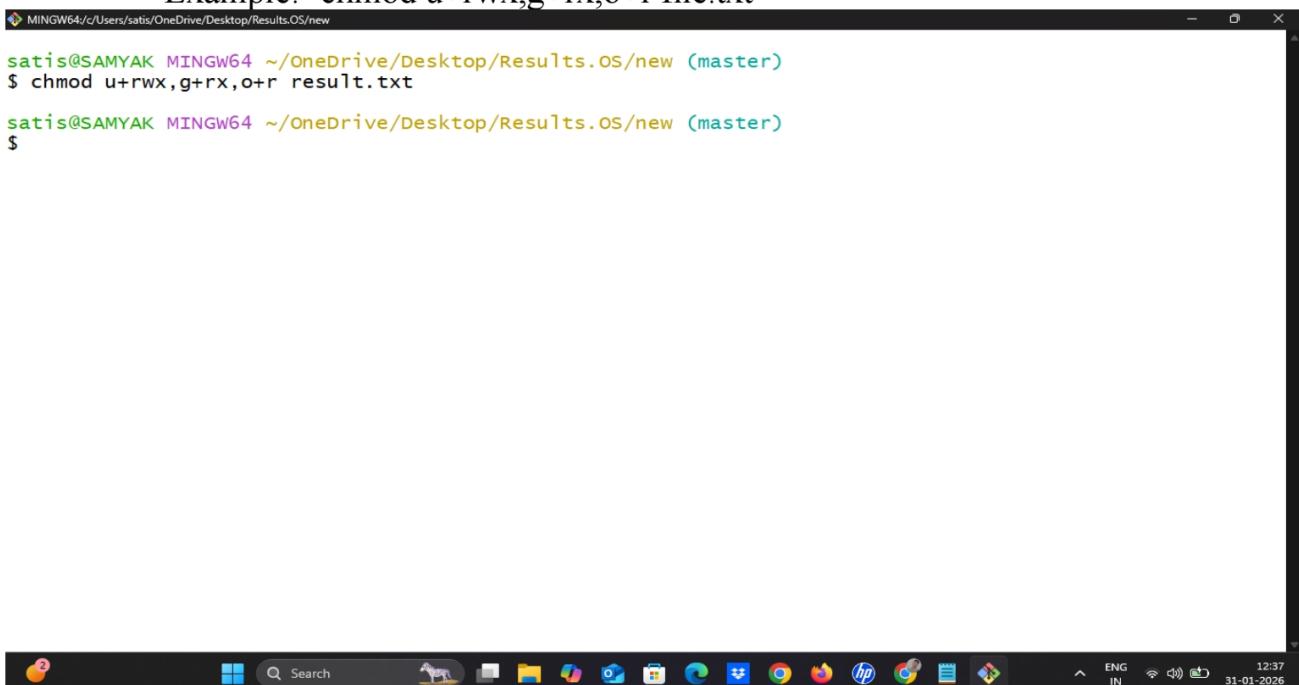


```
MINGW64/c/Users/satis/OneDrive/Desktop/Results.OS/new (master)
$ chmod u+w,o-rwx result.txt

satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$
```

19. Assign Permissions to Users.

- Functionality: Modifies file access for users, groups, and others.
- Syntax: chmod u+rwx,g+rx,o+r <filename>
- Example: 'chmod u+rwx,g+rx,o+r file.txt'

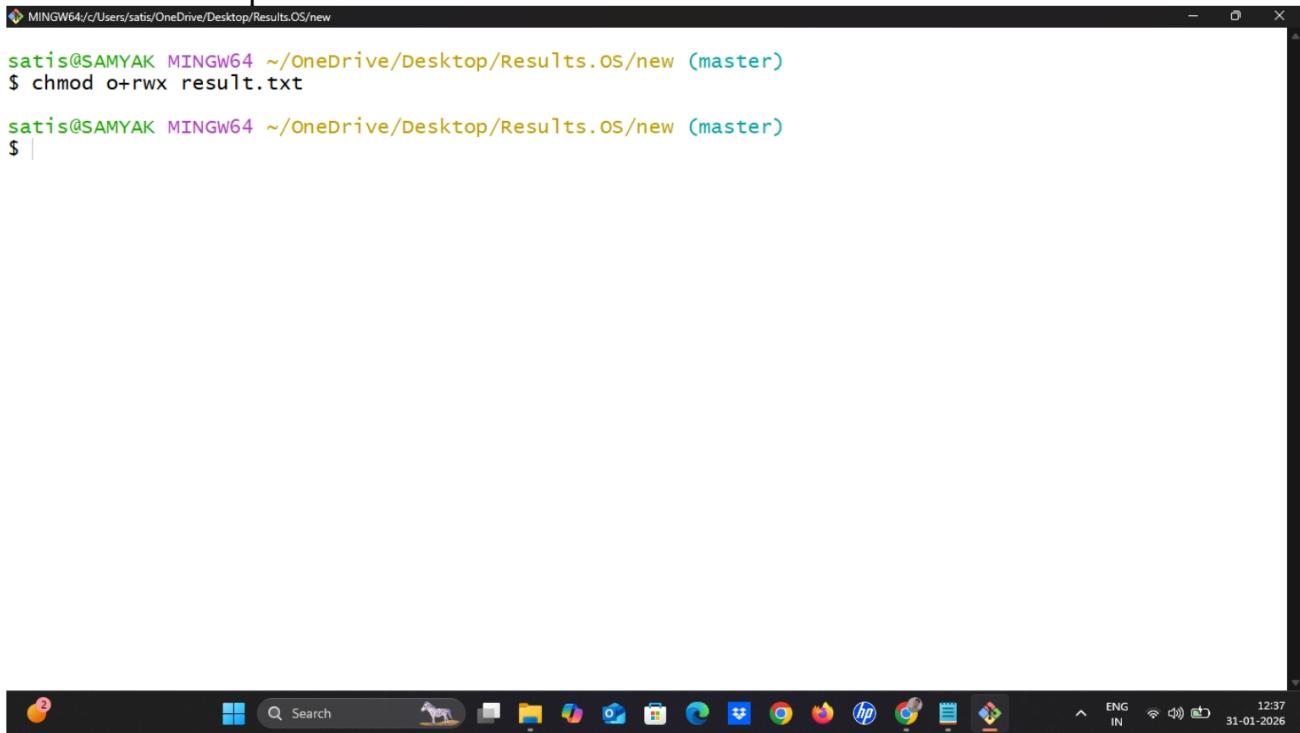


```
MINGW64/c/Users/satis/OneDrive/Desktop/Results.OS/new (master)
$ chmod u+rwx,g+rx,o+r result.txt

satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$
```

20. Assign R/W/X to Others.

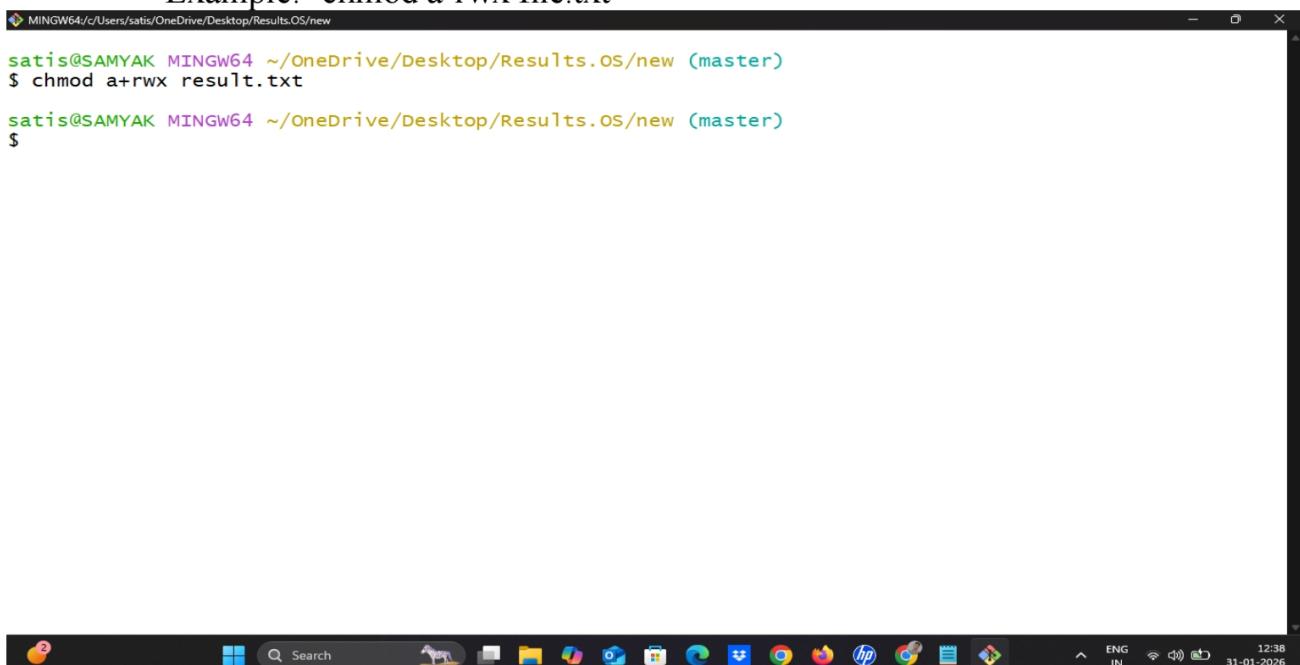
- Functionality: Gives read, write, and execute permissions to others.
- Syntax: chmod o+rwx <filename>
- Example: chmod o+rwx file.txt



```
MINGW64/c/Users/satis/OneDrive/Desktop/Results.OS/new
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$ chmod o+rwx result.txt
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$ |
```

21. Remove All Permissions from All Users.

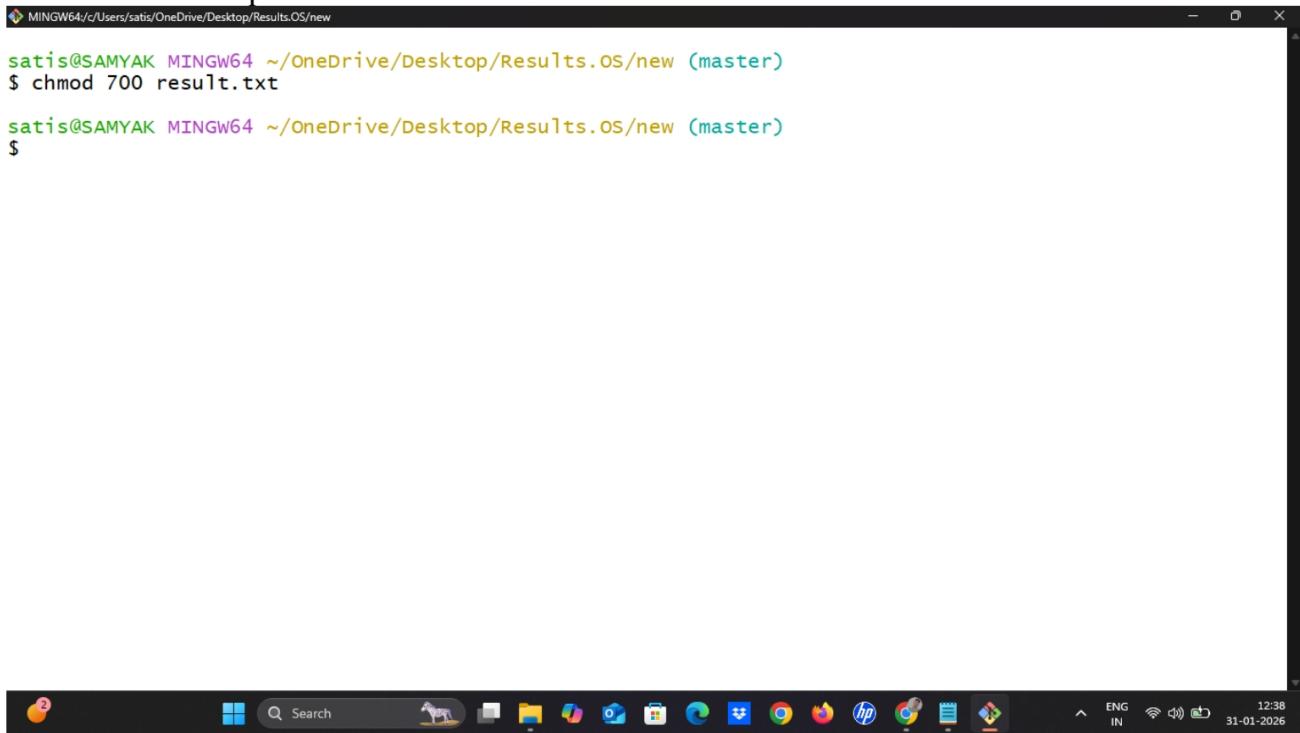
- Functionality: Clears all permissions on a file.
- Syntax: 'chmod a-rwx <filename>
- Example: 'chmod a-rwx file.txt



```
MINGW64/c/Users/satis/OneDrive/Desktop/Results.OS/new
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$ chmod a+rwx result.txt
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$ |
```

22. Remove Read Permission Using Absolute Mode.

- Functionality: Uses numeric mode to restrict read access.
- Syntax: chmod 700 <filename>
- Example: chmod 700 file.txt

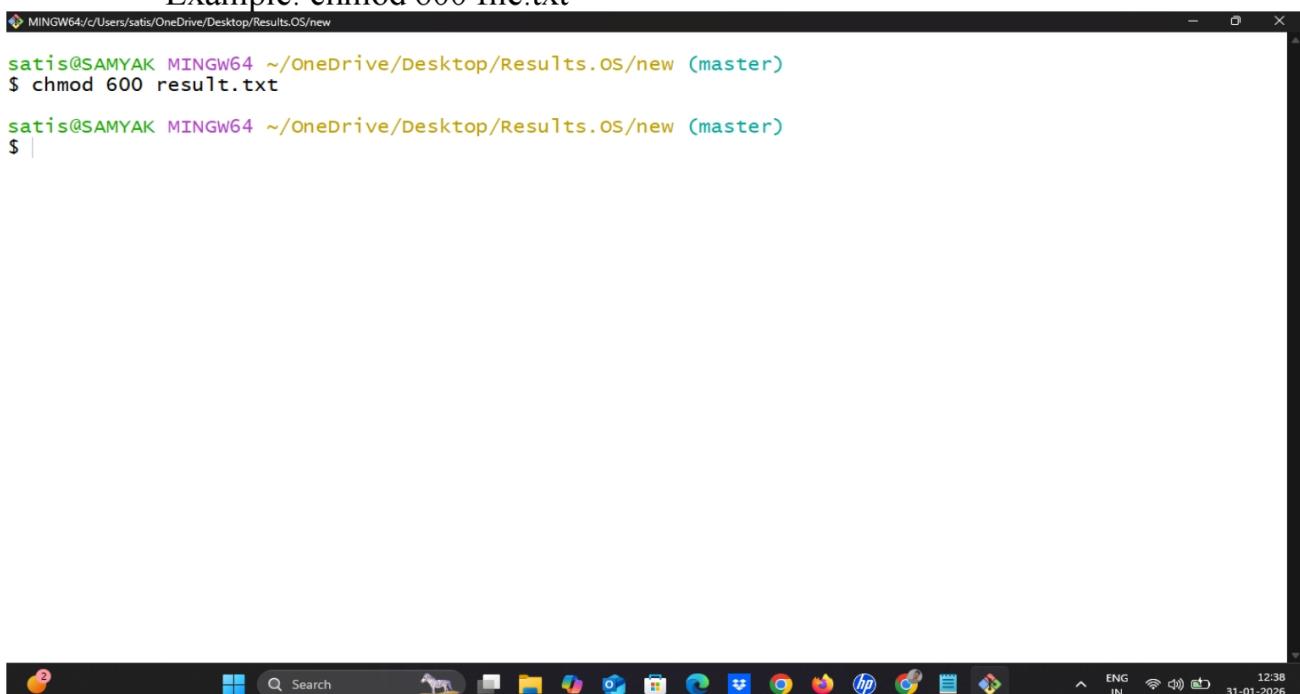


```
MINGW64:/c/Users/satis/OneDrive/Desktop/Results.OS/new
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$ chmod 700 result.txt

satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$
```

23. Set R/W for Owner, None for Group/Other.

- Functionality: Assigns permissions in numeric mode.
- Syntax: chmod 600 <filename>
- Example: chmod 600 file.txt'

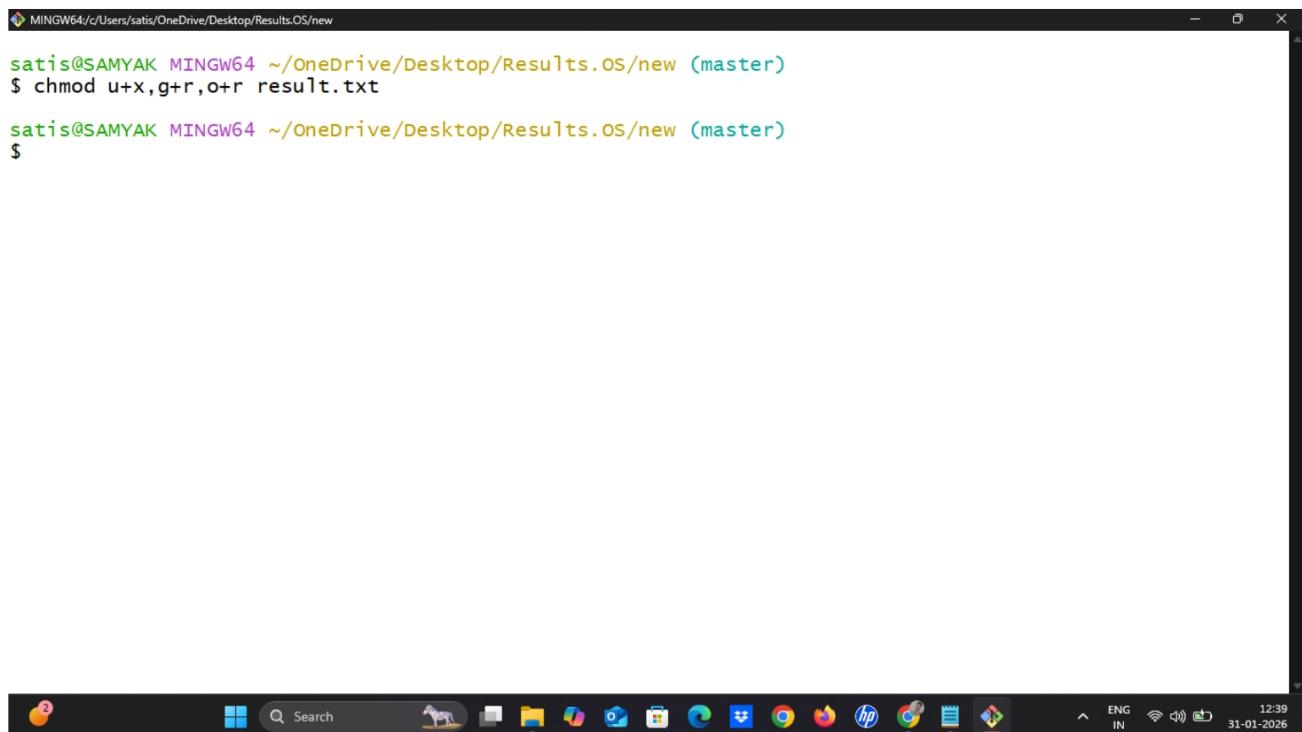


```
MINGW64:/c/Users/satis/OneDrive/Desktop/Results.OS/new
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$ chmod 600 result.txt

satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$ |
```

24. 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

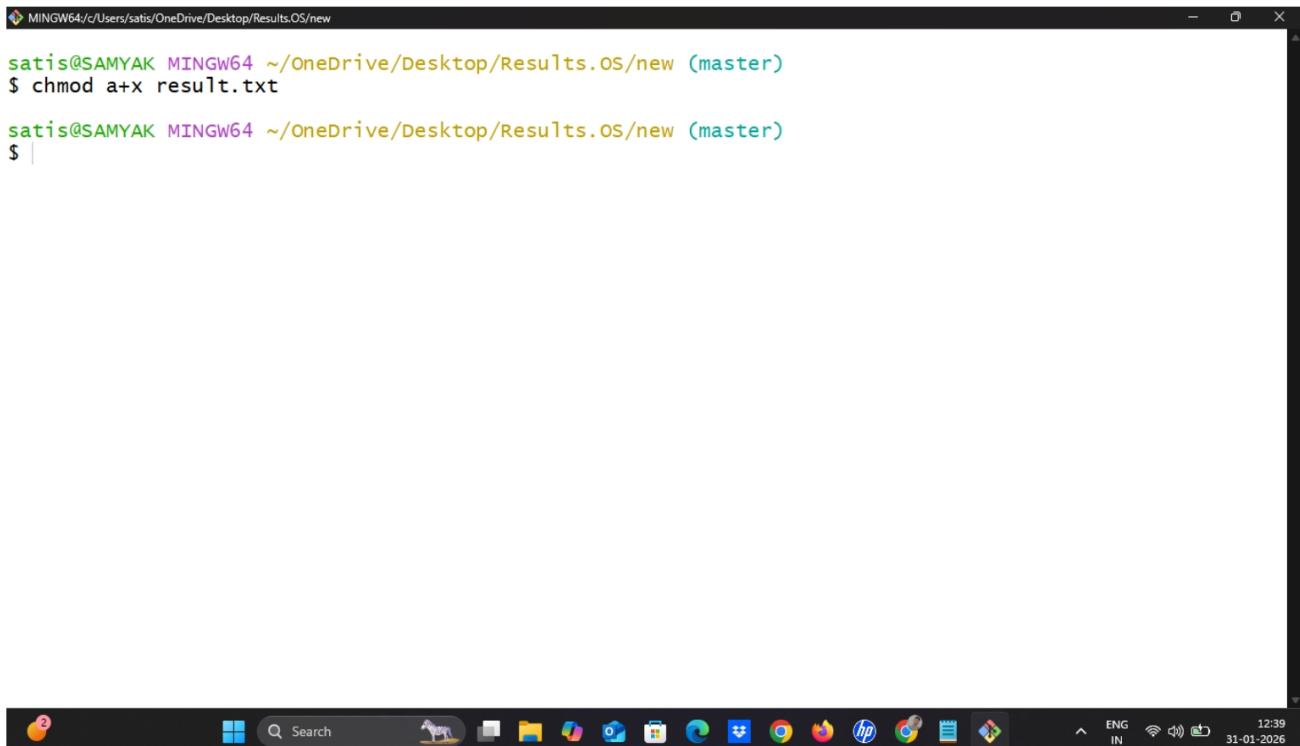


A screenshot of a Windows desktop environment. At the top, there is a taskbar with various icons for applications like File Explorer, Search, Task View, and others. The system tray shows the date (31-01-2026) and time (12:39). In the center, a terminal window titled 'MINGW64/c/Users/satis/OneDrive/Desktop/Results.OS/new' is open. It contains the following command history:

```
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$ chmod u+x,g+r,o+r result.txt
satis@SAMYAK MINGW64 ~/OneDrive/Desktop/Results.OS/new (master)
$
```

25. Add Execute Permission to All Users.

- Functionality: Enables execution by everyone.
- Syntax: chmod a+x <filename>
- Example: chmod a+x script.sh



The screenshot shows a Windows desktop environment with a terminal window open. The terminal window title is 'MINGW64:/c/Users/satis/OneDrive/Desktop/Results.OS/new'. The command entered is '\$ chmod a+x result.txt'. The terminal window has a dark background with light-colored text. The desktop taskbar at the bottom shows various icons for applications like File Explorer, Edge, and Google Chrome. The system tray indicates the date as 31-01-2026 and the time as 12:39.

❖ **Conclusion:** In conclusion, understanding and using essential operating system commands like ‘ls’, ‘cd’, ‘cp’, ‘mv’, and ‘chmod’ enables efficient file management, navigation, and permission control. Tools like ‘grep’, ‘head’, and ‘tail’ enhance data processing. Mastery of these commands improves system administration, task automation, and overall system security and performance.

❖ Discussion Questions:

1. What is the significance of the pwd command in a Linux environment?

Answer: The pwd (print working directory) command displays the absolute path of the current working directory. It helps verify the user's present location in the file system. Syntax: pwd.

2. **Explain the function of the cp command and its common options.**
Answer: The cp command copies files or directories. Syntax: cp <source> <destination>. Options like -r copy directories recursively, and -i prompts before overwriting.
3. **How does chmod 700 affect file permissions, and what does each digit represent?**
Answer: chmod 700 grants full permissions (read, write, execute) to the owner and no permissions to others. The digits represent permissions for the owner, group, and others, respectively.
4. **Describe the difference between head and tail commands in Linux.**
Answer: The head command displays the first N lines of a file, while tail shows the last N lines. Both accept the -n option to specify the number of lines.
5. **What is the purpose of the grep command, and how is it used with regular expressions?**
Answer: The grep command searches for patterns within files using regular expressions. Syntax: grep <pattern> <file>. It supports options like -i for case-insensitive search and -v to invert the match.

❖ **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(DS)
Sem: 4 / 2025-26