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: Ankita Maroti Bailwad

USN: CM24008

Semester / Year: 4th/2nd

Academic Session: 2025-26

Date of Performance: 20-01-26

Date of Submission: 27-01-26

❖ **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:

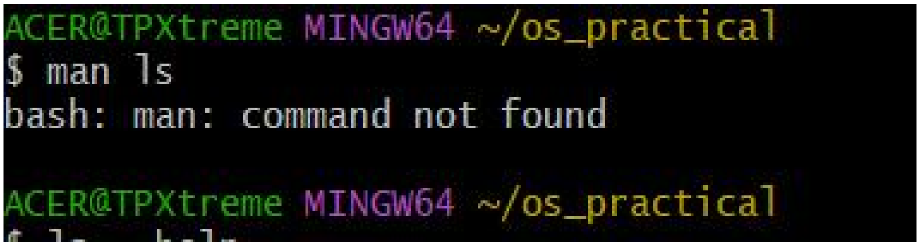
- **Operating System**: Windows 10/11, Linux (Ubuntu 20.04/CentOS 8), UNIX-based OS
- **Command-line Interface**: PowerShell or Command Prompt (Windows), Terminal (Linux/UNIX)
- **Text Editor**: Nano, Vim, or Visual Studio Code for file editing
- **Administrative Privileges**: Superuser (Linux/UNIX)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

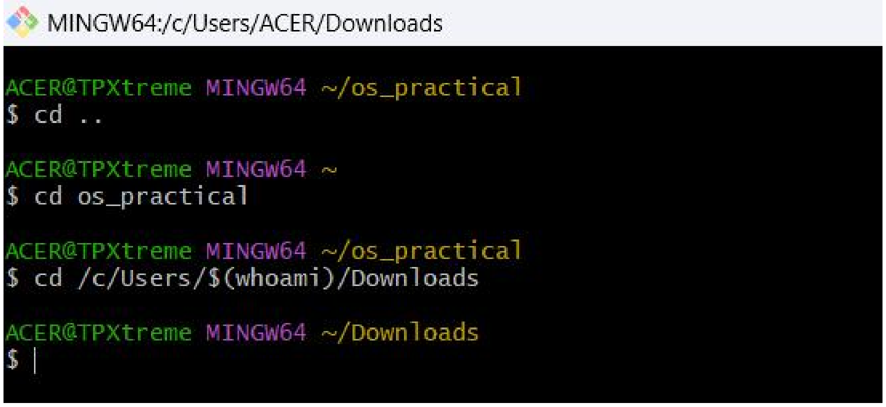
❖ Commands:

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



```
ACER@TPXtreme MINGW64 ~/os_practical
$ man ls
bash: man: command not found

ACER@TPXtreme MINGW64 ~/os_practical
$ ls --help
```

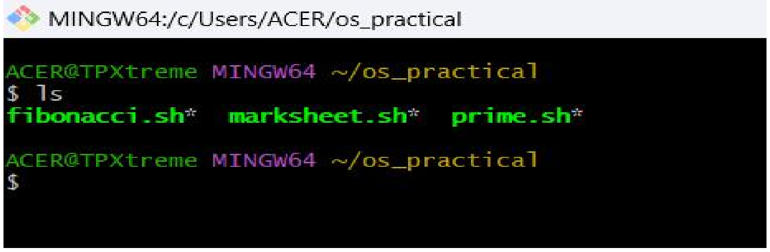
2. **Change Current Working Directory**.
   - Functionality: Changes the terminal's current working directory.
   - Syntax: cd <directory-path>
   - Example: cd /home/user/Documents.



MINGW64:/c/Users/ACER/Downloads

```
ACER@TPXtreme MINGW64 ~/os_practical
$ cd ..

ACER@TPXtreme MINGW64 ~
$ cd os_practical

ACER@TPXtreme MINGW64 ~/os_practical
$ cd /c/Users/$(whoami)/Downloads

ACER@TPXtreme MINGW64 ~/Downloads
$ |
```

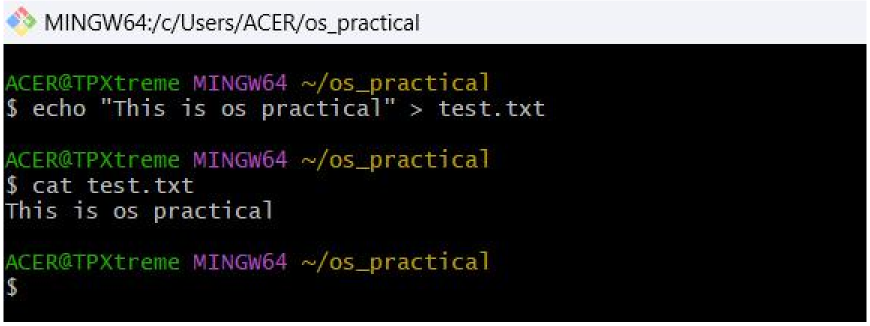
3. **List Contents of the Current Directory.**
   - Functionality: Lists all files and directories in the current location.
   - Syntax: ls
   - Example: ls



MINGW64:/c/Users/ACER/os_practical

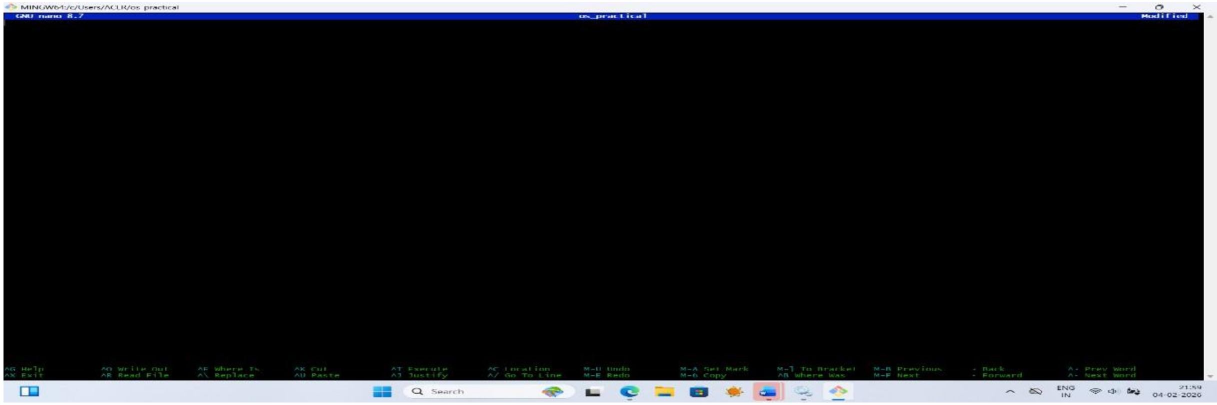
```
ACER@TPXtreme MINGW64 ~/os_practical
$ ls
fibonacci.sh*  marksheet.sh*  prime.sh*

ACER@TPXtreme MINGW64 ~/os_practical
$
```

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



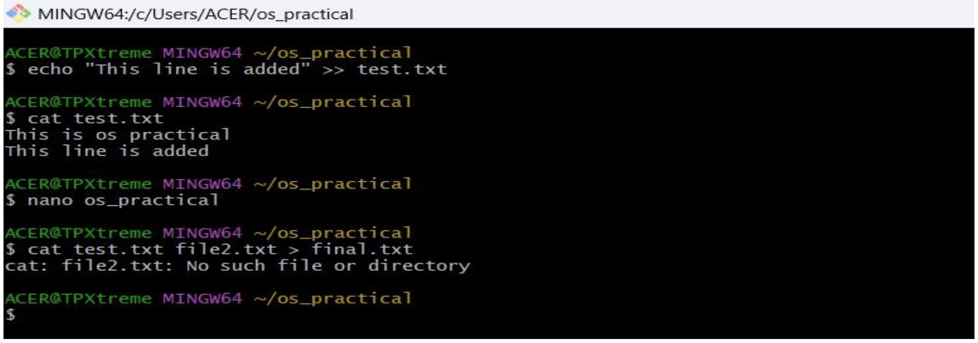
MINGW64:/c/Users/ACER/os_practical

```
ACER@TPXtreme MINGW64 ~/os_practical
$ echo "This is os practical" > test.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ cat test.txt
This is os practical

ACER@TPXtreme MINGW64 ~/os_practical
$
```



MINGW64:/c/Users/ACER/os_practical

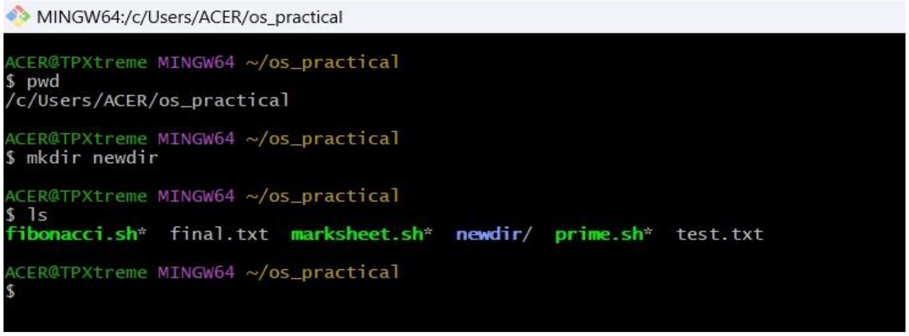
```
  GNU nano 8.7                         os_practical                         Modified

^G Help    ^O Write Out  ^F Where Is  ^K Cut    ^T Execute  ^C Location    M-U Undo  M-A Set Mark  M-] To Bracket  M-B Previous  ◂ Back     ^◂ Prev Word
^X Exit    ^R Read File  ^\ Replace   ^U Paste  ^J Justify  ^/ Go To Line  M-E Redo  M-6 Copy      ^B Where Was    M-F Next      ▸ Forward  ^▸ Next Word
```



MINGW64:/c/Users/ACER/os_practical

```
ACER@TPXtreme MINGW64 ~/os_practical
$ echo "This line is added" >> test.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ cat test.txt
This is os practical
This line is added

ACER@TPXtreme MINGW64 ~/os_practical
$ nano os_practical

ACER@TPXtreme MINGW64 ~/os_practical
$ cat test.txt file2.txt > final.txt
cat: file2.txt: No such file or directory

ACER@TPXtreme MINGW64 ~/os_practical
$
```

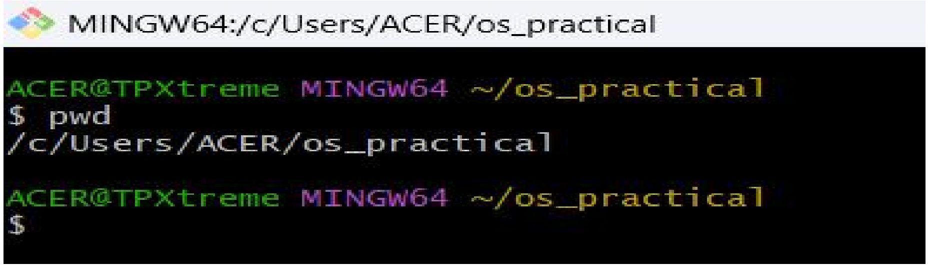
5. **Create a New Directory**.
   - Functionality: Creates a new directory at the specified path.
   - Syntax: mkdir <directory-name
   - Example: mkdir newdir



MINGW64:/c/Users/ACER/os_practical

```
ACER@TPXtreme MINGW64 ~/os_practical
$ pwd
/c/Users/ACER/os_practical

ACER@TPXtreme MINGW64 ~/os_practical
$ mkdir newdir

ACER@TPXtreme MINGW64 ~/os_practical
$ ls
fibonacci.sh*  final.txt  marksheet.sh*  newdir/  prime.sh*  test.txt

ACER@TPXtreme MINGW64 ~/os_practical
$
```

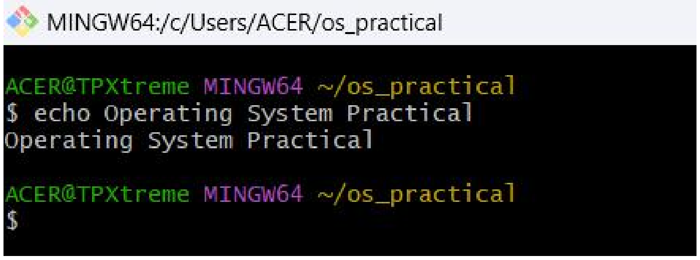
6. **Display Current Working Directory.**
   - Functionality: Prints the current directory path.
   - Syntax: pwd
   - Example: pwd



MINGW64:/c/Users/ACER/os_practical

```
ACER@TPXtreme MINGW64 ~/os_practical
$ pwd
/c/Users/ACER/os_practical

ACER@TPXtreme MINGW64 ~/os_practical
$
```

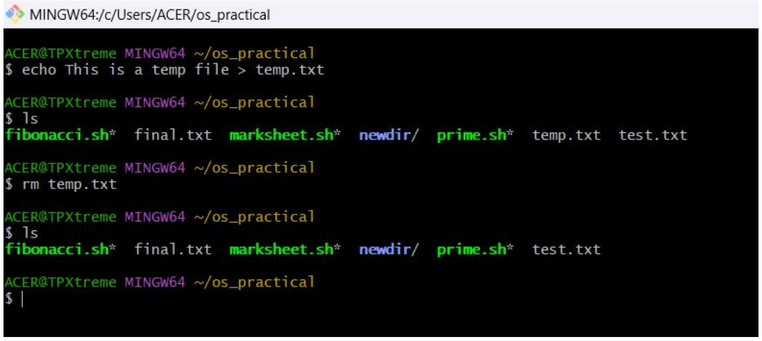
7. **Write Arguments to Standard Output.**
   - Functionality: Prints the provided string or variables.
   - Syntax: echo <arguments>
   - Example: echo Hello World



MINGW64:/c/Users/ACER/os_practical

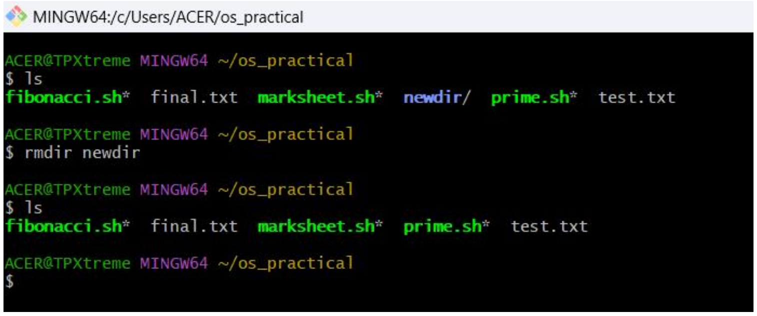
```
ACER@TPXtreme MINGW64 ~/os_practical
$ echo Operating System Practical
Operating System Practical

ACER@TPXtreme MINGW64 ~/os_practical
$
```

8. **Remove a File.**
   - Functionality: Deletes a specified file.
   - Syntax: rm <filename>
   - Example: rm file.txt



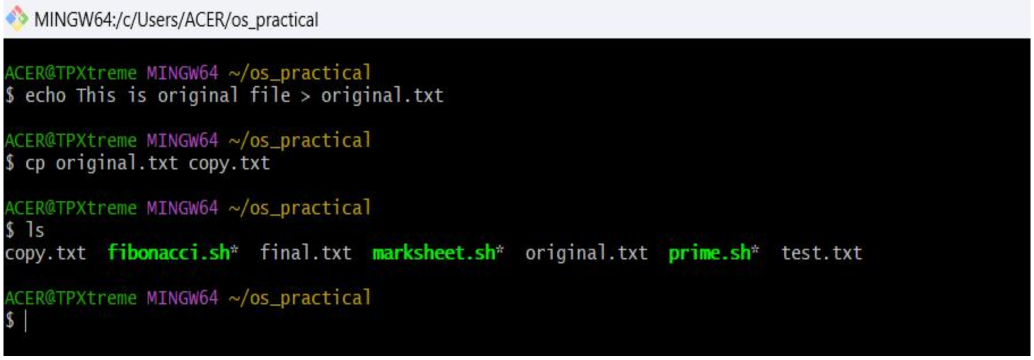
```
ACER@TPXtreme MINGW64 ~/os_practical
$ echo This is a temp file > temp.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ ls
fibonacci.sh*  final.txt  marksheet.sh*  newdir/  prime.sh*  temp.txt  test.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ rm temp.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ ls
fibonacci.sh*  final.txt  marksheet.sh*  newdir/  prime.sh*  test.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ |
```

9. **Delete a Directory.**
   - Functionality: Removes an empty directory.
   - Syntax: rmdir <directory-name>
   - Example: rmdir olddir



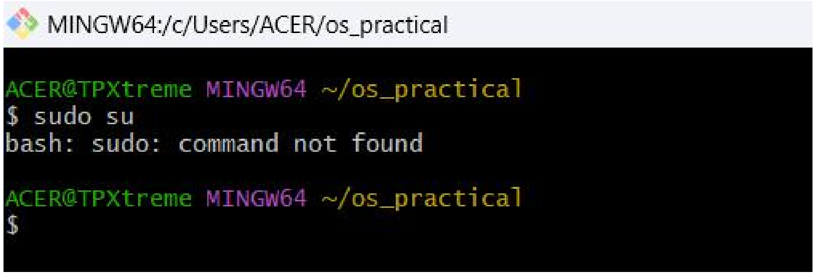
```
ACER@TPXtreme MINGW64 ~/os_practical
$ ls
fibonacci.sh*  final.txt  marksheet.sh*  newdir/  prime.sh*  test.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ rmdir newdir

ACER@TPXtreme MINGW64 ~/os_practical
$ ls
fibonacci.sh*  final.txt  marksheet.sh*  prime.sh*  test.txt

ACER@TPXtreme MINGW64 ~/os_practical
$
```

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



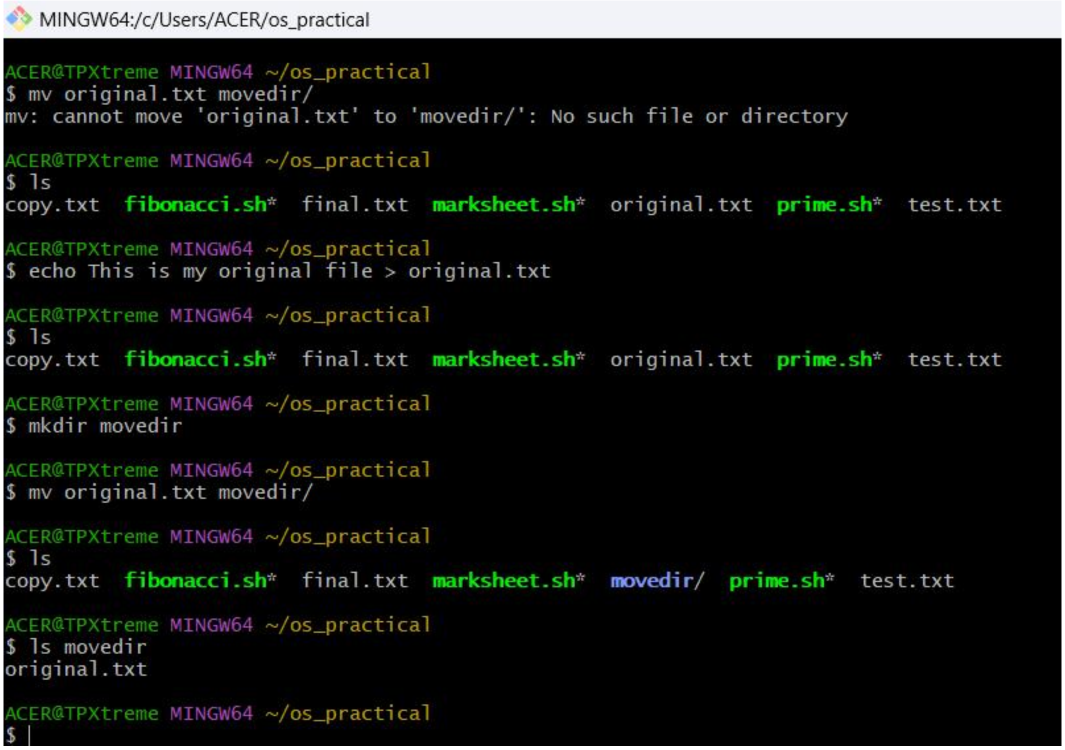
```
ACER@TPXtreme MINGW64 ~/os_practical
$ echo This is original file > original.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ cp original.txt copy.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ ls
copy.txt  fibonacci.sh*  final.txt  marksheet.sh*  original.txt  prime.sh*  test.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ |
```

11. **Switch to Root User.**
    - Functionality: Gains root privileges temporarily.
    - Syntax: sudo su
    - Example: sudo s



```
ACER@TPXtreme MINGW64 ~/os_practical
$ sudo su
bash: sudo: command not found

ACER@TPXtreme MINGW64 ~/os_practical
$
```

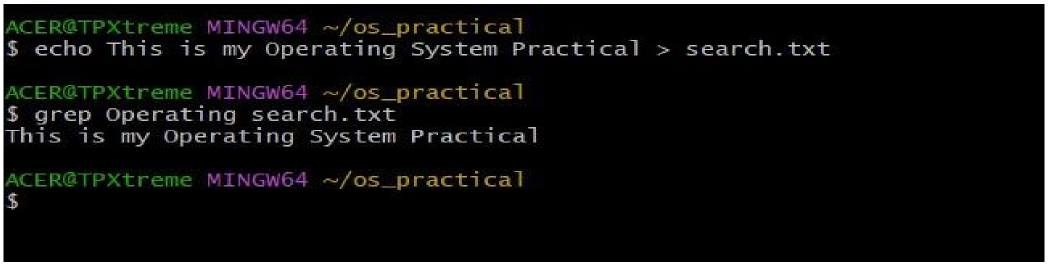
12. **Move Files or Directories.**
    - Functionality: Moves or renames files and directories.
    -Syntax: mv <source> <destination>
    - Example: mv file.txt newdir/



```
ACER@TPXtreme MINGW64 ~/os_practical
$ mv original.txt movedir/
mv: cannot move 'original.txt' to 'movedir/': No such file or directory

ACER@TPXtreme MINGW64 ~/os_practical
$ ls
copy.txt  fibonacci.sh*  final.txt  marksheet.sh*  original.txt  prime.sh*  test.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ echo This is my original file > original.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ ls
copy.txt  fibonacci.sh*  final.txt  marksheet.sh*  original.txt  prime.sh*  test.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ mkdir movedir

ACER@TPXtreme MINGW64 ~/os_practical
$ mv original.txt movedir/

ACER@TPXtreme MINGW64 ~/os_practical
$ ls
copy.txt  fibonacci.sh*  final.txt  marksheet.sh*  movedir/  prime.sh*  test.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ ls movedir
original.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ |
```

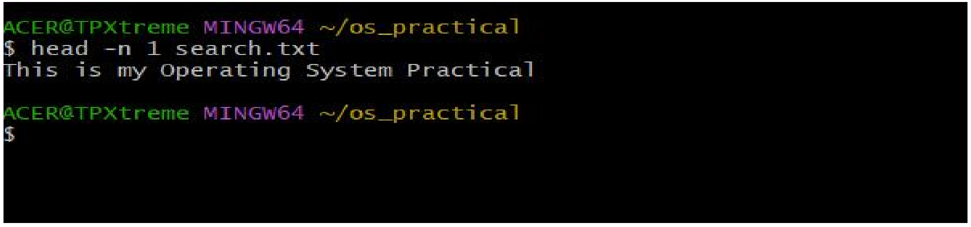
13. **Search for a String in a File.**
    - Functionality: Searches for a specific word or pattern in a file.
    - Syntax: grep "<string>" <file>

- Example: grep "error" log.txt



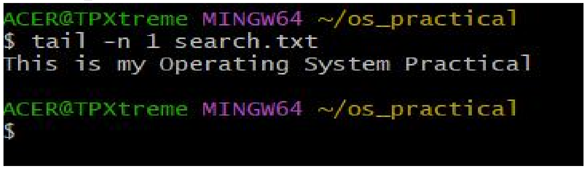
```
ACER@TPXtreme MINGW64 ~/os_practical
$ echo This is my Operating System Practical > search.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ grep Operating search.txt
This is my Operating System Practical

ACER@TPXtreme MINGW64 ~/os_practical
$
```

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’



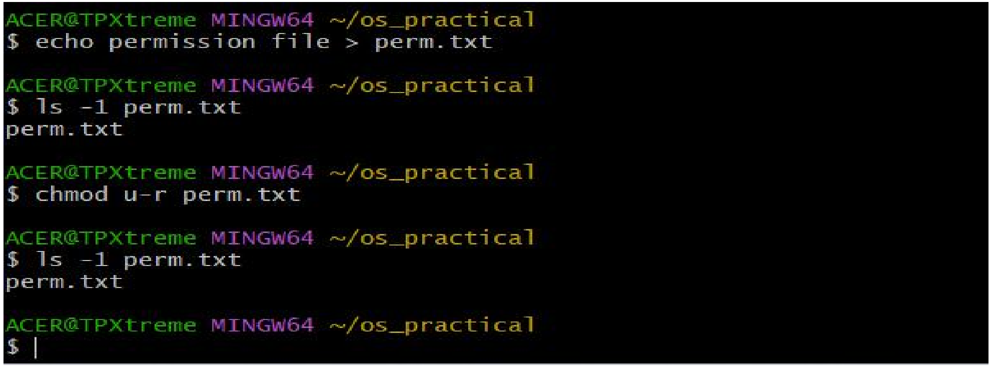
```
ACER@TPXtreme MINGW64 ~/os_practical
$ head -n 1 search.txt
This is my Operating System Practical

ACER@TPXtreme MINGW64 ~/os_practical
$
```

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



```
ACER@TPXtreme MINGW64 ~/os_practical
$ tail -n 1 search.txt
This is my Operating System Practical

ACER@TPXtreme MINGW64 ~/os_practical
$
```

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



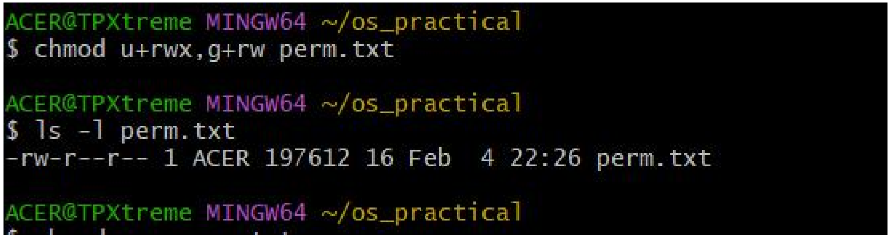
```
ACER@TPXtreme MINGW64 ~/os_practical
$ echo permission file > perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ ls -1 perm.txt
perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ chmod u-r perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ ls -1 perm.txt
perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ |
```

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



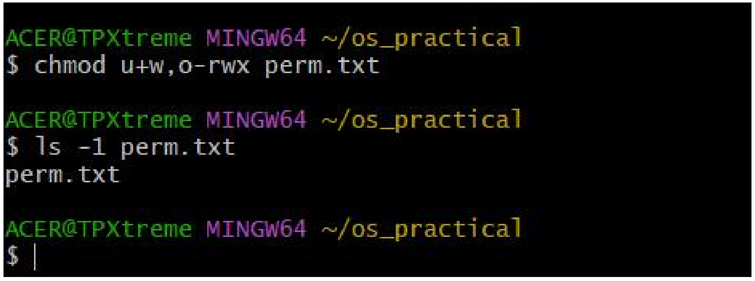
```
ACER@TPXtreme MINGW64 ~/os_practical
$ chmod u+rwx,g+rw perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ ls -l perm.txt
-rw-r--r-- 1 ACER 197612 16 Feb  4 22:26 perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
```

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



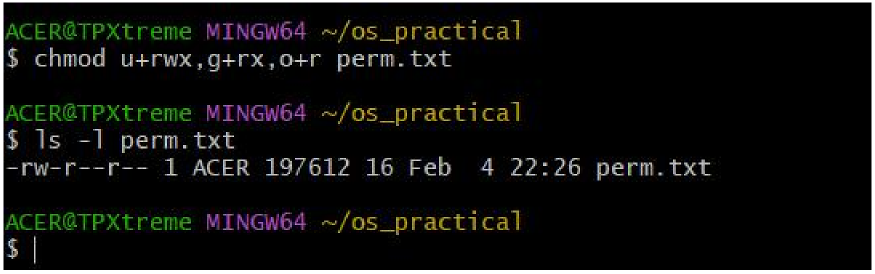
```
ACER@TPXtreme MINGW64 ~/os_practical
$ chmod u+w,o-rwx perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ ls -1 perm.txt
perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ |
```

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



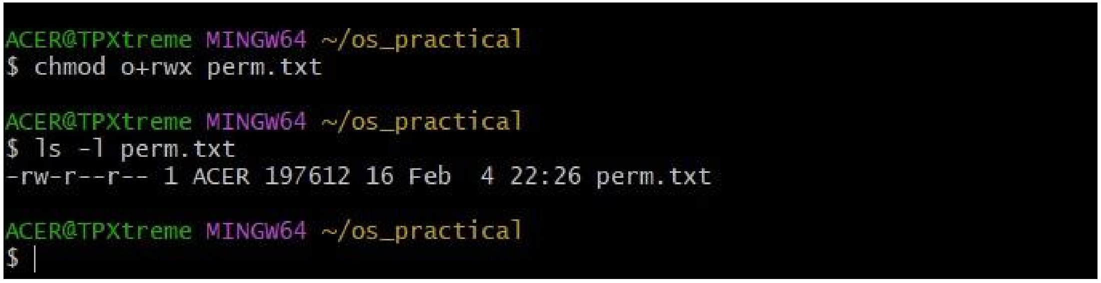
```
ACER@TPXtreme MINGW64 ~/os_practical
$ chmod u+rwx,g+rx,o+r perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ ls -l perm.txt
-rw-r--r-- 1 ACER 197612 16 Feb  4 22:26 perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ |
```

20. **Assign R/W/X to Others.**

- Functionality: Gives read, write, and execute permissions to others.

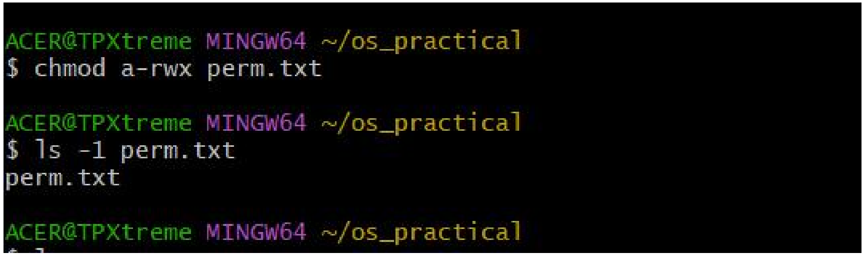
- Syntax: chmod o+rwx <filename>

- Example: chmod o+rwx file.txt



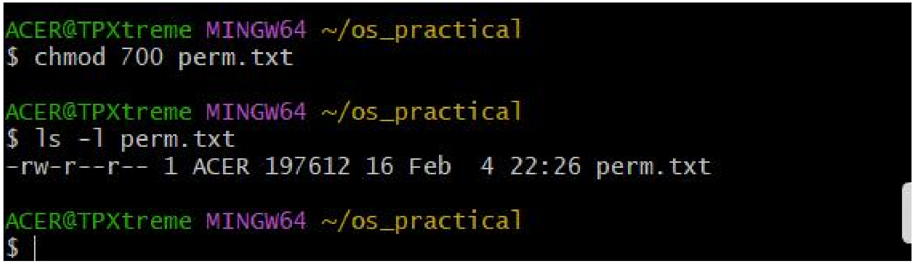
```
ACER@TPXtreme MINGW64 ~/os_practical
$ chmod o+rwx perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ ls -l perm.txt
-rw-r--r-- 1 ACER 197612 16 Feb  4 22:26 perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ |
```

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



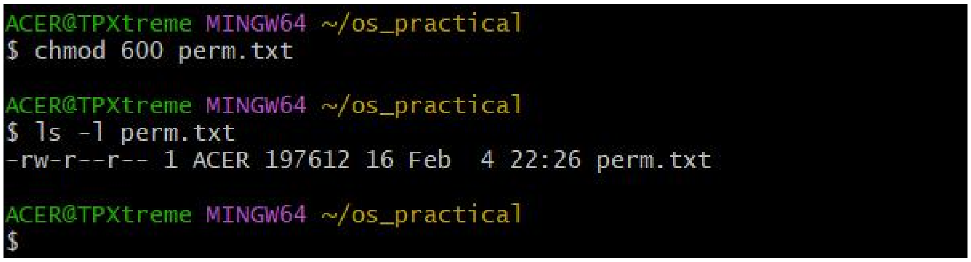
```
ACER@TPXtreme MINGW64 ~/os_practical
$ chmod a-rwx perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ ls -1 perm.txt
perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
```

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



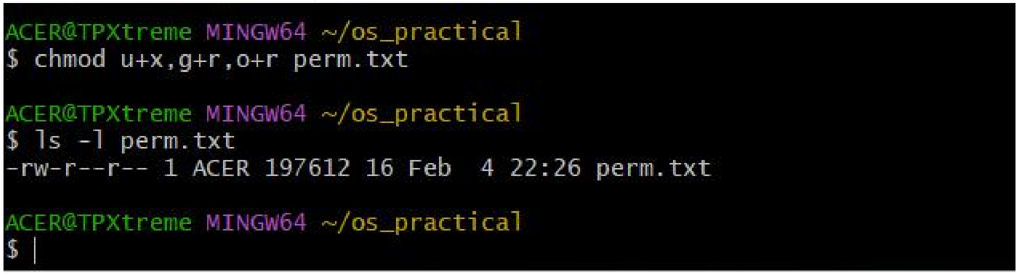
```
ACER@TPXtreme MINGW64 ~/os_practical
$ chmod 700 perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ ls -l perm.txt
-rw-r--r-- 1 ACER 197612 16 Feb  4 22:26 perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ |
```

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’



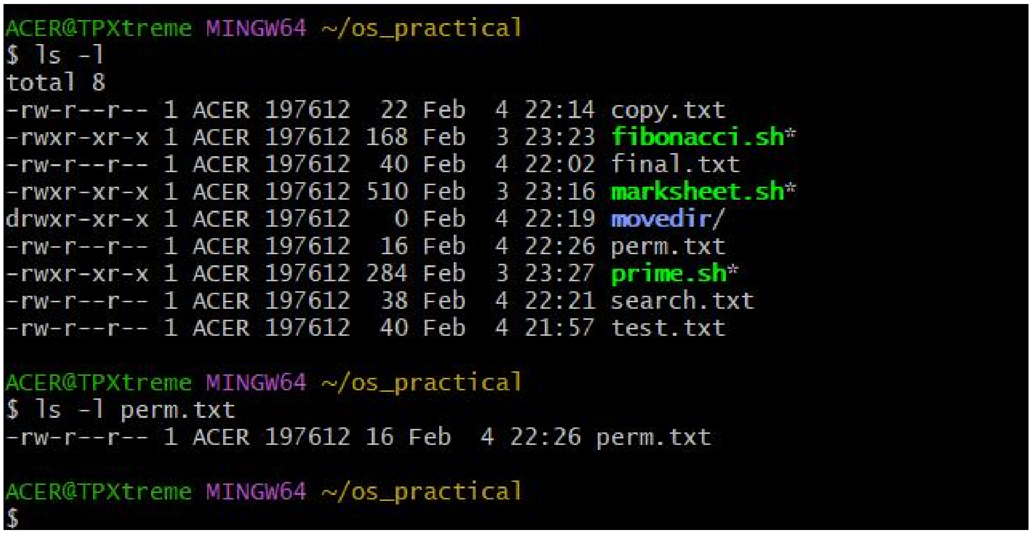
```
ACER@TPXtreme MINGW64 ~/os_practical
$ chmod 600 perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ ls -l perm.txt
-rw-r--r-- 1 ACER 197612 16 Feb  4 22:26 perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$
```

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



```
ACER@TPXtreme MINGW64 ~/os_practical
$ chmod u+x,g+r,o+r perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ ls -l perm.txt
-rw-r--r-- 1 ACER 197612 16 Feb  4 22:26 perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ |
```

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



```
ACER@TPXtreme MINGW64 ~/os_practical
$ ls -l
total 8
-rw-r--r-- 1 ACER 197612  22 Feb  4 22:14 copy.txt
-rwxr-xr-x 1 ACER 197612 168 Feb  3 23:23 fibonacci.sh*
-rw-r--r-- 1 ACER 197612  40 Feb  4 22:02 final.txt
-rwxr-xr-x 1 ACER 197612 510 Feb  3 23:16 marksheet.sh*
drwxr-xr-x 1 ACER 197612   0 Feb  4 22:19 movedir/
-rw-r--r-- 1 ACER 197612  16 Feb  4 22:26 perm.txt
-rwxr-xr-x 1 ACER 197612 284 Feb  3 23:27 prime.sh*
-rw-r--r-- 1 ACER 197612  38 Feb  4 22:21 search.txt
-rw-r--r-- 1 ACER 197612  40 Feb  4 21:57 test.txt

ACER@TPXtreme MINGW64 ~/os_practical
$ ls -l perm.txt
-rw-r--r-- 1 ACER 197612 16 Feb  4 22:26 perm.txt

ACER@TPXtreme MINGW64 ~/os_practical
$
```

- **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?**
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:<u>**27/01/2026**</u>

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