**OPERATING SYSTEM(4ITRC2)**

**IT IV(Semester)**

*Submitted by*

**Name of Student – Akanksha singh**

**Roll Number - 06**

**Branch with Section – IT(B)**

*Submitted to*

*MRS\_JASNEET KAUR*

Assistant Professor, Department of Information Technology

Institute of Engineering and Technology

Devi Ahilya Vishwavidhyalaya, Indore (M.P.) India

**(**[www.iet.dauniv.ac.in](http://www.iet.dauniv.ac.in)**)**

**Session JAN-APRIL,2025**

**Q1. Outputs of the following commands**

**pwd**: Prints the current working directory. For example, /home/user.

 **cd**: Changes the current directory. This command does not produce output unless there's an error.

 **ls**: Lists the files and directories in the current directory. The output might be something like:

File.TXT file2.txt folder1

**mkdir**: Creates a new directory. Example: mkdir new\_folder creates a folder named new\_folder. It produces no output unless there's an error.

 **rm**: Removes files or directories. For example, rm file1.txt deletes the file. It produces no output unless there's an error.

 **touch**: Creates an empty file or updates the timestamp of a file. Example: touch newfile.txt creates a file named newfile.txt. No output is shown.

 **hostname**: Displays the name of the machine you're using. Example output: my-computer.

 **cat**: Concatenates and displays file content. Example: cat file.txt might show:

Hello, world!

 **chmod**: Changes the permissions of a file or directory. Example: chmod 755 file.txt changes the permissions, with no output unless there's an error.

 **echo**: Prints a string to the terminal. Example: echo Hello outputs:

Hello

 **grep**: Searches for a pattern in files. Example: grep Hello file.txt might output:

Hello, world!

 **fgrep**: Same as grep -F, searches for fixed strings without interpreting special characters. Behavior is similar to grep.

 **mv**: Moves or renames files or directories. Example: mv file.txt newfile.txt. No output is shown unless there's an error.

 **cp**: Copies files or directories. Example: cp file.txt backup.txt. No output unless there's an error.

 **more**: Displays file content one screen at a time. Example: more file.txt shows the content interactively.

 **less**: Like more, but more versatile for browsing file content interactively. Example: less file.txt.

**wc**: Counts the lines, words, and characters in a file. Example: wc file.txt might output:

5, 20 120 file.txt

(lines, words, characters)

**awk**: A powerful text-processing tool. Example: awk '{print $1}' file.txt prints the first word of each line in file.txt.

**sed**: A stream editor for text transformation. Example: sed 's/old/new/' file.txt replaces "old" with "new" in file.txt.

**tail**: Displays the last few lines of a file. Example: tail file.txt might show:

Line 8

Line 9

Line 10

**PART 2:**

Q1 How to navigate to a Specific Directory?

Ans: If you want to navigate to a directory named Documents in your home directory, you can use:

cd ~/Documents

The ~ symbol refers to your home directory.

If the directory is located elsewhere, provide the full or relative path. For example:

* Full path:

cd /home/username/Documents

* Relative path (assuming you're in the home directory):

cd username/Documents

2. How to see detailed information about files and directories using ls?

Ans: command

Ls -l

Output:

-rw-r--r-- 1 user group 1024 Mar 23 09:00 file1.txt

drwxr-xr-x 2 user group 4096 Mar 23 09:00 folder1

 **File type and permissions** (-rw-r--r--): Indicates whether it's a file (-) or directory (d) and the permission settings.

 **Number of links** (1): Number of hard links.

 **Owner** (user): The name of the user who owns the file.

 **Group** (group): The group associated with the file.

 **Size** (1024): The size of the file in bytes.

 **Modification date and time** (Mar 23 09:00): The last modification timestamp.

f**ile/directory name** (file1.txt, folder1): The name of the file or directory

**3. How to create multiple directories in Linux using mkdir command?**

Ans: Use the mkdir command with multiple directory names or use -p to create nested directories.

mkdir dir1 dir2 dir3

mkdir -p parent\_dir/child\_dir

**4. How to remove multiple files at once with rm?**

Ans Pass multiple filenames to rm. Example:

bash

rm file1 file2 file3

**Can rm be used to delete directories?**

Yes, use the -r option to delete directories recursively.

Code:

rm -r directory\_name

for empty directories we can use rmdir

**6. How Do You Copy Files and Directories in Linux?**

To copy a file:

cp file1.txt file2.txt

To copy a directory:

cp -r source\_directory destination\_directory

**7. How to Rename a file in Linux Using mv Command?**

Use mv to rename a file.

Code:

mv old\_filename.txt new\_filename.txt

**8. How to Move Multiple Files in Linux Using mv Command?**

List multiple files and specify the target directory.

Code:

mv file1.txt file2.txt target\_directory/

**9. How to Create Multiple Empty Files by Using touch Command in Linux?**

Specify multiple filenames with touch.

Code:

touch file1.txt file2.txt file3.txt

**10. How to View the Content of Multiple Files in Linux?**

Use cat or more with file names.

Code:

cat file1.txt file2.txt

code:

more file1.txt file2.txt

**11. How to Create a File and Add Content in Linux Using cat Command?**

Use cat with redirection.

Code:

cat > newfile.txt

**12. How to Append the Contents of One File to the End of Another File Using cat Command?**

Use >> to append.

Code:

cat source\_file >> target\_file

**13. How to Use cat Command If the File Has a Lot of Content and Can’t Fit in the Terminal?**

Combine cat with less or more.

Code:

cat file.txt | less

or

code:

cat file.txt | more

**14. How to Merge Contents of Multiple Files Using cat Command?**

Provide multiple filenames to cat.

Code:

cat file1.txt file2.txt > merged\_file.txt

**15. How to Use cat Command to Append to an Existing File?**

Use >> for appending.

Code:

cat new\_content.txt >> existing\_file.txt

**16. What is “chmod 777”, “chmod 755”, and “chmod +x” or “chmod a+x”?**

* chmod 777: Grants read, write, and execute permissions to all users.
* chmod 755: Grants full permissions to the owner and read/execute permissions to others.
* chmod +x or chmod a+x: Adds execute permission to a file for all users.

**17. How to Find the Number of Lines That Matches the Given String/Pattern?**

Use grep with the -c flag.

Code:

grep -c "pattern" file.txt

**18. How to Display the Files That Contain the Given String/Pattern?**

Use grep with file names.

Code:

grep "pattern" file1.txt file2.txt

**19. How to Show the Line Number of a File with the Line Matched?**

Use grep -n for line numbers.

Code:

grep -n "pattern" file.txt

**20. How to Match the Lines That Start with a String Using grep?**

Use ^ in your pattern.

Code:

grep "^start\_string" file.txt

**21. Can the ‘sort’ Command Be Used to Sort Files in Descending Order by Default?**

No, the default sorting is ascending. Use -r for descending order.

Code:

sort -r file.txt

**22. How Can I Sort a File Based on a Specific Column Using the ‘sort’ Command?**

Specify the column with -k. For example, to sort by the second column:

Code:

sort -k 2 file.txt