Assignment 2

**Part-1:**

**Output of the following commands**

**1. pwd**  
*Command:* pwd  
*Output:* Prints the present working directory. Example:

/home/user

**2. cd**  
*Command:* cd <directory>  
*Output:* Changes the current directory to the specified one. No output if successful. Example:

cd Documents

**3. ls**  
*Command:* ls  
*Output:* Lists files and directories in the current directory. Example:

file1.txt file2.txt folder1 folder2

**4. mkdir**  
*Command:* mkdir new\_directory  
*Output:* Creates a new directory. No output if successful.

**5. rm**  
*Command:* rm file.txt  
*Output:* Removes the specified file. No output if successful.

**6. touch**  
*Command:* touch newfile.txt  
*Output:* Creates an empty file or updates the timestamp of an existing file. No output if successful.

**7. hostname**  
*Command:* hostname  
*Output:* Displays the system’s hostname. Example:

my-computer

**8. cat**  
*Command:* cat file.txt  
*Output:* Displays the contents of the specified file.

**9. chmod**  
*Command:* chmod 777 file.txt  
*Output:* Changes the file permissions. No output if successful.

**10. echo**  
*Command:* echo "Hello, World!"  
*Output:*

Hello, World!

**11. grep**  
*Command:* grep "text" file.txt  
*Output:* Displays lines in the file that contain the specified text.

**12. fgrep**  
*Command:* fgrep "text" file.txt  
*Output:* Similar to grep, but treats the pattern as a fixed string instead of a regular expression.

**13. mv**  
*Command:* mv oldfile.txt newfile.txt  
*Output:* Renames or moves a file. No output if successful.

**14. cp**  
*Command:* cp source.txt destination.txt  
*Output:* Copies a file. No output if successful.

**15. more**  
*Command:* more file.txt  
*Output:* Displays file contents page by page.

**16. less**  
*Command:* less file.txt  
*Output:* Similar to more, but allows both forward and backward navigation.

**17. wc**  
*Command:* wc file.txt  
*Output:* Displays the word, line, and character count of the file.

**18. awk**  
*Command:* awk '{print $1}' file.txt  
*Output:* Extracts and prints the first column of text.

**19. sed**  
*Command:* sed 's/old/new/g' file.txt  
*Output:* Replaces all occurrences of "old" with "new" in the file.

**20. tail**  
*Command:* tail -n 10 file.txt  
*Output:* Displays the last 10 lines of the file.

**Part-2:**

**Answer to the following questions :**

**1. How to navigate to a specific directory?**

**Theory:**  
The cd (change directory) command is used to move between directories in Linux. You can navigate to any directory by specifying its absolute or relative path.

**Command:**

bash

cd /path/to/directory

**Example:**

bash

cd /home/user/Documents

* cd .. → Moves up one directory.
* cd ~ → Moves to the home directory.

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

**Theory:**  
The ls (list) command is used to view the contents of a directory. The -l flag provides detailed information, including permissions, owner, size, and modification date.

**Command:**

bash

ls -l

**Example:**

bash

ls -la # Includes hidden files

* ls -lh → Shows file sizes in human-readable format.
* ls -lt → Sorts files by modification time.

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

**Theory:**  
The mkdir command is used to create directories. You can specify multiple directories at once or use the -p flag to create nested directories.

**Command:**

bash

mkdir dir1 dir2 dir3

**Example:**

bash

CopyEdit

mkdir -p parent/child/grandchild

* -p → Creates parent directories if they don’t exist.

**4. How to copy multiple files at once?**

**Theory:**  
The cp (copy) command is used to duplicate files. You can copy multiple files to a directory by specifying their names followed by the destination path.

**Command:**

bash

cp file1 file2 file3 /destination/path/

**Example:**

bash

cp \*.txt /home/user/backup/

* cp -r directory\_name /destination/path/ → Copies a directory recursively.

**5. How to delete directories in Linux?**

**Theory:**  
The rm (remove) command is used to delete files and directories. To delete a directory and its contents, use the -r (recursive) option.

**Command:**

bash

rm -r directory\_name

**Example:**

bash

rm -rf folder\_name

* -r → Removes directories and their contents.
* -f → Forces deletion without confirmation.

**6. How to copy files and directories in Linux?**

**Theory:**  
The cp command is used for copying files and directories. The -r flag is used to copy directories recursively.

**Command:**

bash

cp file1.txt /destination/path/

**Example:**

bash

cp -r my\_folder /backup/location/

* cp -i → Prompts before overwriting files.

**7. How to rename a file in Linux using the mv command?**

**Theory:**  
The mv (move) command is used to rename or move files and directories. To rename a file, specify the current name and the new name.

**Command:**

bash

mv old\_filename new\_filename

**Example:**

bash

mv report.txt summary.txt

* mv -i → Asks before overwriting a file.

**8. How to move multiple files in Linux using the mv command?**

**Theory:**  
The mv command can move multiple files at once by listing them before the destination directory.

**Command:**

bash

mv file1 file2 file3 /destination/path/

**Example:**

bash

mv \*.pdf /home/user/Documents/

* Moves all .pdf files to the "Documents" folder.

**9. How to create multiple empty files using the touch command in Linux?**

**Theory:**  
The touch command is used to create empty files. You can specify multiple file names to create multiple files simultaneously.

**Command:**

bash

touch file1 file2 file3

**Example:**

bash

touch report1.txt report2.txt report3.txt

* If the file already exists, touch updates its timestamp.

**10. How to view the contents of multiple files in Linux?**

**Theory:**  
The cat (concatenate) command displays the contents of files. You can specify multiple file names to view them together.

**Command:**

bash

cat file1 file2

**Example:**

bash

cat notes.txt summary.txt

* less filename → Views large files one page at a time.

**11. How to create a file and add content using the cat command?**

**Theory:**  
The cat command can create a file and allow users to add content interactively. The > operator is used to write content to a new file.

**Command:**

bash

cat > filename

**Example:**

bash

cat > my\_notes.txt

* After typing content, press **Ctrl + D** to save the file.

**12. How to append the contents of one file to another file using the cat command?**

**Theory:**  
The cat command with the >> operator appends the content of one file to another.

**Command:**

bash

cat source\_file >> destination\_file

**Example:**

bash

cat notes.txt >> summary.txt

* The contents of notes.txt are added to the end of summary.txt.

**13. How to use cat command if the file has a lot of content and can't fit in the terminal?**

If a file has too much content, you can use cat with a pager like less or more to scroll through the content easily:

cat filename | less

or

cat filename | more

Alternatively, you can directly use less instead of cat for efficiency:

less filename

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

To concatenate multiple files into a single output, use:

cat file1 file2 file3

To merge and save into a new file:

cat file1 file2 > merged\_file

To append to an existing file:

cat file1 file2 >> existing\_file

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

To append content of one file to another:

cat source\_file >> target\_file

To manually type and append text to a file:

cat >> filename

Press Ctrl + D to save and exit.

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

* **chmod 777 filename** → Gives read (r), write (w), and execute (x) permissions to everyone (not secure).
* **chmod 755 filename** → Owner has full access, while others have read and execute permissions (common for scripts).
* **chmod +x filename** → Makes the file executable for the user running the command.
* **chmod a +x filename** → Makes the file executable for all users.

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

Use grep with the -c flag to count matching lines:

grep -c "pattern" filename

Example:

grep -c "error" logfile.txt

Alternatively, using wc -l:

grep "pattern" filename | wc -l

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

Use grep -l to list filenames that contain the pattern:

grep -l "pattern" \*.txt

Example:

grep -l "error" /var/log/\*

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

Use grep -n:

grep -n "pattern" filename

Example:

grep -n "ERROR" logs.txt

This will show output in the format:

pgsql

23: ERROR found in the system

where 23 is the line number.

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

Use the ^ symbol to match the beginning of a line:

grep "^pattern" filename

Example:

grep "^ERROR" logfile.txt

This will only return lines that start with "ERROR".

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

No, sort sorts in ascending order by default. To sort in descending order, use the -r (reverse) flag:

sort -r filename

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

Use -k to specify the column number. Example, sorting by the second column:

sort -k2 filename

For numeric sorting based on a column:

sort -k2 -n filename

For descending order sorting based on a column:

sort -k2 -nr filename