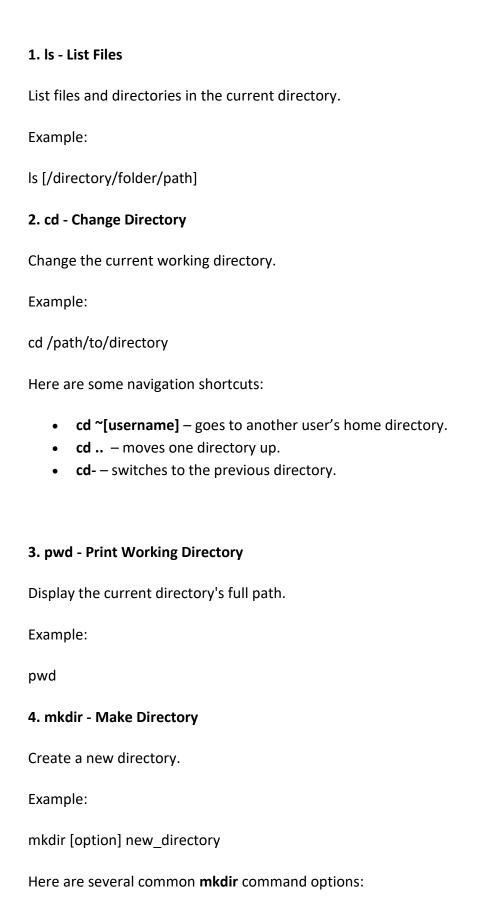
Assignment 1: Linux basic commands



- -p creates a directory between two existing folders. For example, mkdir p
 Music/2023/Songs creates a new 2023 directory.
- -m sets the folder permissions. For instance, enter mkdir -m777 directory to create a directory with read, write, and execute permissions for all users.
- -v prints a message for each created directory.

5. touch - Create Empty File

Create a new empty file.

Example:

touch new file.txt

6. cp - Copy Files and Directories

Copy files or directories from one location to another.

Example:

• Copying one file from the current directory to another folder. Specify the file name and target path:

cp filename.txt /home/username/Documents

• Duplicating an entire directory. Pass the -R flag followed by the source and destination directory:

cp -R /home/username/Documents /home/username/Documents backup

7. rm - Remove Files and Directories

Remove files or directories (use with caution).

Example:

rm [option] unwanted_file.txt

To modify the command, add the following options:

- -i prompts a confirmation before deletion.
- -f allows file removal without a confirmation.
- -r deletes files and directories recursively.

Note: Use the **rm** command with caution since deletion is irreversible. Avoid using the **-r** and **-f** options since they may wipe all your files. Always add the **-i** option to avoid accidental deletion.

8. rmdir - Remove Directories

Use the **rmdir** command to delete an empty directory in Linux.

rmdir [option] directory_name

If the folder contains a subdirectory, the command will return an error. To force delete a non-empty directory, use the **-p** option.

9. touch - Create Empty File

Create a new empty file.

Example:

touch new file.txt

10. my - Move or Rename Files

Move or rename files and directories.

Example (move):

mv filename.txt /home/username/Documents

Example (rename):

mv old_filename.txt new_filename.txt

11. cat - Concatenate and Display File Content

Display the contents of a file.

Example:

cat filename.txt

There are various ways to use the **cat** command:

• cat > filen.txt - creates a new file.

- cat file1.txt file2.txt > file3.txt merges file1.txt with file2.txt and stores the output in file3.txt.
- tac file.txt displays content in reverse order.

12. grep - Search Text

The **global regular expression** or **grep** command lets you find a word by searching the content of a file

content of a file. Example: grep blue notepad.txt 13. head and tail - Display Beginning/End of File Display the first or last lines of a file. Example (head): head file.txt # Display first 5 lines Example (tail): tail file.txt # Display last 5 lines 14. chmod - Change File Permissions Change file permissions (read, write, execute). Example: chmod 755 script.sh Read(1), write(2), and execute(4) for owner (1+2+4=7); read(1) and execute(4) for group(1+4=5) and others(1+4=5)

15. chown - Change File Owner

Change the owner of a file.

Example:

chown new_owner:group file.txt

16. du - Disk Usage of Directories

Display disk usage of directories and files.

Example:

du -sh /path/to/directory

The **du** command has several options, such as:

- -s shows the specified folder's total size.
- -m provides folder and file information in MB.
- -k displays information in KB.
- -h informs the displayed folders and files' last modification date.

17. zip, unzip - Compress and Decompress Files

The **zip** command lets you compress items into a **ZIP** file with the optimal compression ratio. Here's the syntax:

zip [options] zipfile file1 file2....

For example, this command compresses **note.txt** into **zipfile.zip** in the current working directory:

zip archive.zip note.txt

Use the **unzip** command to extract the compressed file. Here's the syntax:

unzip file_name.zip

18. ps - Process Status

List running processes.

Example:

ps [option]

The **ps** command accepts several options, including:

- -T displays all processes associated with the current shell session.
- -u username lists processes associated with a specific user.
- -A shows all the running processes.

19. kill - Terminate Processes

Terminate processes by their process ID (PID)
Example:
kill [PID]

20. uname - Unix Name

The **uname** or **unix name** command prints information about your machine, including its hardware, system name, and Linux kernel.

Example:

uname [option]

While you can use it without an option, add the following to modify the command:

- -a prints all the system information.
- -s outputs the kernel name.
- -n shows the system's node hostname.

21. sudo - Run with special privilege

Superuser do or **sudo** is one of the most basic commands in Linux. It runs your command with administrative or root permissions.

Example:

sudo useradd username

You can also add an option, such as:

- -k invalidates the timestamp file.
- -g executes commands as a specified group name or ID.
- -h runs commands on the host.