**Practice: File and Directory answers**

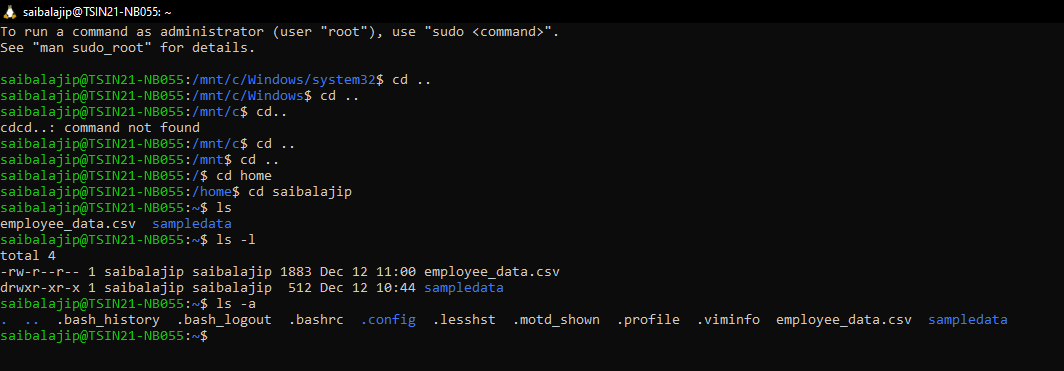
Here are detailed answers with screenshots and practical examples for the Linux command-based questions:

ls – List directory contents 1. What is the difference between ls , ls -l , and ls -a ? Provide examples.

ls : Lists the files and directories in the current directory.

ls -l : Provides detailed information about each file (permissions, owner, group, size, and timestamp).

Example: ls -a : Displays all files, including hidden files (those starting with a dot . ).



2. How can you display the contents of a directory sorted by file size using ls?

Use ls -lS to sort files by size. Add -h for human-readable sizes.

Example:

A computer screen with white text

Description automatically generated

3. Explain the purpose of the ls -i option and its practical use.?

ls -i shows the inode number of each file, which is useful for managing hard links or troubleshooting filesystem issues.

Example:

A black screen with white text

Description automatically generated

4. How can you use ls to only list files with a specific extension (e.g., .txt )?

Use wildcard matching:

A black background with green text

Description automatically generated

5. How do you display the human-readable file sizes in the output of ls ?

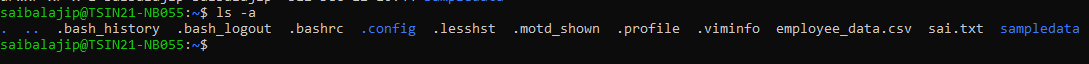
Use the -h option:

A black screen with white text

Description automatically generated

6. How can you use ls to display hidden files in a directory?

Use ls -a :



7. What is the difference between ls -R and ls -d \*/ ?

ls -R : Recursively lists all subdirectories and their contents.

ls -d \*/ : Lists only the directories in the current path (no contents).

Example:

A computer screen with white text and green letters

Description automatically generated

8. How would you use ls to display file types (e.g., directories, regular files, symbolic links)?

Use the -F option, which adds a / for directories, @ for symbolic links, and \* for executable files:

A black background with green and white text

Description automatically generated

cd – Change directories

9. What does cd .. do? How is it different from cd ../.. ?

cd .. moves up one directory level.

cd ../.. moves up two directory levels.

Example:

A screenshot of a computer

Description automatically generated

10. Explain the behavior of cd - . Why is it useful?

cd - switches to the previous directory. Useful for toggling between two directories.

Example:

A black screen with green and blue text

Description automatically generated

11. What happens when you run cd without any arguments?

It takes you to your home directory:



12. How can you navigate to a directory with spaces in its name using cd ?

Quote the directory name or escape the spaces with a backslash:

A screen shot of a computer screen

Description automatically generated

13. Describe a situation where using an absolute path in cd is better than using a relative path.

Absolute paths are better when running scripts or when the current directory is unpredictable.



14. What is the difference between cd ~ and cd $HOME ? Are they always equivalent?

Both take you to the home directory, and they are usually equivalent unless $HOME is overridden.

15. How is the output of pwd different from pwd -P ? Provide examples.

pwd : Shows the logical current directory, including symbolic links.

pwd -P : Resolves the symbolic links to show the physical path.

Example:

A screenshot of a computer code

Description automatically generated

16. What is the significance of pwd in shell scripting?

It helps scripts dynamically get the current directory for file operations.

Example:

17. Why might the pwd output differ after using cd into a symbolic link? Explain with an example.

Logical pwd shows the symbolic link path; physical pwd -P resolves it.

Example:

ln -s /real/path /shortcut

cd /shortcut

pwd

pwd -P

18. How can you store the output of pwd into a shell variable and use it later in a script? Example: mkdir – Create directories

19. How can you create a directory along with its parent directories in one command?

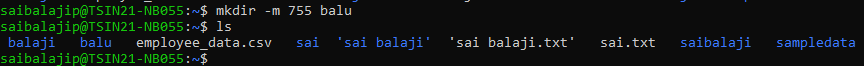
Use -p :

20. What happens if you try to create a directory that already exists using mkdir ? How can you suppress the error message?

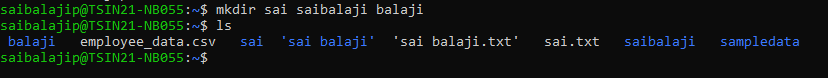
Use -p to suppress the error:

21. Explain the purpose of the -m option in mkdir . How would you use it to set specific permissions on a directory during creation?

Use -m to set permissions during creation:



22. How can you use mkdir to create multiple directories at once (e.g., dir1, dir2, dir3)? Example: 1



rm – Remove files or directories

23. Explain the difference between rm filename , rm -r directory , and rm -rf directory .

rm filename : Deletes a file.

rm -r directory : Deletes a directory and its contents recursively.

rm -rf directory : Forces deletion without confirmation.

24. Why should you be cautious when using rm -rf ?

Provide a real-world example of unintended consequences. Mistakenly running rm -rf / can delete the entire filesystem.

Example:

rm -rf /

25. How can you use rm interactively to confirm file deletion?

Use -i :

rm -i file.txt

26. How would you remove all files with a specific extension (e.g., .log ) in the current directory using rm ?

Example:

rm \*.log

27. What is the difference between rm -i and rm -f ? When would you use each?

rm -i : Asks for confirmation before deleting.

rm -f : Deletes without confirmation.

28. How can you ensure that rm does not accidentally delete files by enabling a safety mechanism?

Use an alias: