Linux Command Practice

File Operations

1

```
touch project_ideas.txt
```

This command created an empty file named project_ideas.txt in my current directory.

2

```
mkdir reports
```

This command created a new directory named reports in my current directory.

3

```
mv project_ideas.txt reports/
```

This moved the file project_ideas.txt into the reports directory.

4

```
mv project_ideas.txt initial_ideas.txt
```

This command failed because project_ideas.txt had already been moved into the reports directory, so it no longer existed in the current directory.

5

```
touch important_notes.md
```

This command created an empty file named important_notes.md in my current directory.

6

```
echo "# Important Reminders\\n- Backup the server regularly.\\n- Monitor system resources.\\n- Keep software updated." > important_notes.md
```

This overwrote important_notes.md with the provided text, using \\n to create a multi-line file containing a header and three bullet points.

7

```
head -n 2 important_notes.md
```

This displayed the first two lines of the file.

Output observed:
Important Reminders
- Backup the server regularly.

8
`tail -n 2 important_notes.md`
This displayed the last two lines of the file.
Output observed:
- Monitor system resources.
- Keep software updated.

9
`less important_notes.md`
This opened the file in a scrollable view using the less pager. I could navigate using arrow keys and exit by pressing q.

10
`echo "New project list coming soon!" > project_ideas.txt`
This created (or overwrote) a file named project_ideas.txt with the single line:
New project list coming soon!

11
`reports$ cat project_ideas.txt`
This displayed the content of the project_ideas.txt file located in the reports directory.
Output observed:
New project list coming soon!

Navigation & Directory Management

12
`cd ..`
This command moved up one level to the parent directory.

13

`pwd`

This displayed the full path of the current working directory.

14

`ls -lh`

This listed the contents of the current directory in long format with human-readable file sizes.

15

`ls -a`

This showed all files and directories, including hidden ones, in the current directory.

16

`mkdir reports`

This created a directory named reports in the current directory.

17

`cd reports`

This changed the current directory to the reports directory.

18

`cd -`

This returned to the previous directory I was in before the last cd command.

19

`mkdir reports/archive`

This created a subdirectory named archive inside the reports directory.

20

`rmdir reports/archive`

This removed the empty archive directory from inside the reports directory.

Package Management (APT - Assuming a Debian-based system like Ubuntu)

21

`whoami`

This displayed the username of the currently logged-in user.

22

`groups`

This listed all the groups the current user belongs to.

23

`ls`

This listed the files and directories in the current working directory.

24

`ls -lh important_notes.md`

This showed the details of important_notes.md with human-readable file size.

25

`chmod 755 important_notes.md`

This changed the file permissions to:

- Owner: read, write, execute
- Group: read, execute
- Others: read, execute

26

`ls -l important_notes.md`

This confirmed the new permission settings of the file after the chmod command.

27

`chmod g+w,o+w important_notes.md`

This added write permissions for both group and others.

28

`ls -l important_notes.md`

This showed the updated permissions reflecting write access for group and others.

System Information

29

`ps`

This displayed a snapshot of the current active processes associated with the user's terminal.

30

`top`

This launched an interactive display showing real-time system summary information and a list of tasks currently managed by the kernel.

31

`htop`

This launched an enhanced version of top (if installed), with a user-friendly interface to monitor system processes and usage.

32

`ps aux | grep nano`

This searched for any running 'nano' processes by displaying all processes and filtering the results to show only those related to 'nano'.
It also shows the process ID (PID), CPU/memory usage, and other details if nano is running.

33

`uname -a`

This displayed detailed system information, including the kernel name, version, machine type, and operating system.

34

`hostname`

This showed the system's hostname (i.e., the name assigned to the machine on the network).

35

`uptime`

This displayed how long the system has been running, along with the current time, number of users, and load average.

36

`free -m`

This showed the system's memory usage in megabytes, including total, used, free, and buffers/cache.

37

`df -h`

This displayed disk space usage of all mounted filesystems in human-readable format (e.g., GB, MB).

Help & Documentation

38

`man man`

This opened the manual page for the `man` command itself, explaining how to use the man pages and navigate them.

39

`whatis man`

This displayed a brief one-line description of the `man` command, typically pulled from the manual database.