

Assignment : 2

Task : 2

2. File Maintenance Commands:

a. Task: Create a backup of a directory - Use the cp command to create a backup copy of an entire directory, preserving its structure.

Certainly! To create a backup of a directory using the cp command in Linux, you can use the following command:

cp -r source_directory destination_backup_directory

Here's a breakdown of the command:

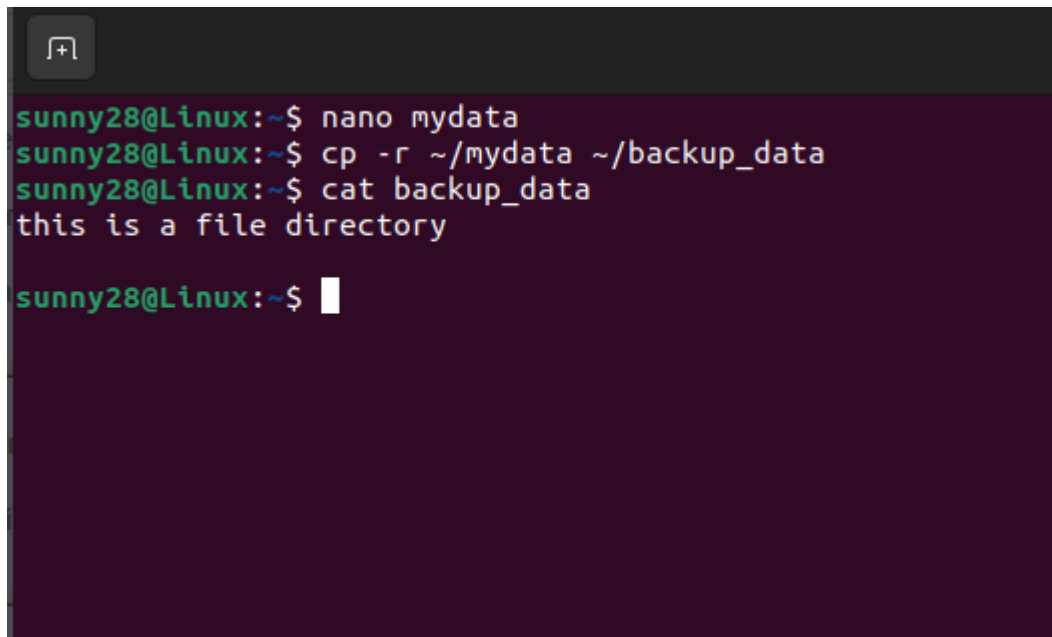
cp: This is the command for copying files and directories.

-r: This option stands for recursive, allowing the cp command to copy directories and their contents recursively.

source_directory: Replace this with the path of the directory you want to back up.

destination_backup_directory: Replace this with the path where you want to store the backup copy.

For example, if you want to back up a directory named "mydata" located in your home directory to a backup directory named "backup_data" in the same location, you would run:

A terminal window with a dark background and a light blue icon in the top-left corner. The terminal shows a series of commands and their outputs. The first command is 'nano mydata', which is followed by a prompt. The second command is 'cp -r ~/mydata ~/backup_data', followed by another prompt. The third command is 'cat backup_data', which outputs the text 'this is a file directory'. The final line shows a prompt with a cursor, indicating the terminal is ready for the next command.

```
sunny28@Linux:~$ nano mydata
sunny28@Linux:~$ cp -r ~/mydata ~/backup_data
sunny28@Linux:~$ cat backup_data
this is a file directory

sunny28@Linux:~$
```

This command copies the entire "mydata" directory and its contents to the "backup_data" directory. Make sure you have the necessary permissions to read the source directory and write to the destination directory.

b. Task: Delete files that match a specific pattern - Use the `rm` command with wildcards to delete files that match a certain pattern (e.g., all `.tmp` files in a directory).

Certainly! To delete files that match a specific pattern using the `rm` command with wildcards, you can use the following command:

`rm path/to/files/*.pattern`

Here's a breakdown of the command:

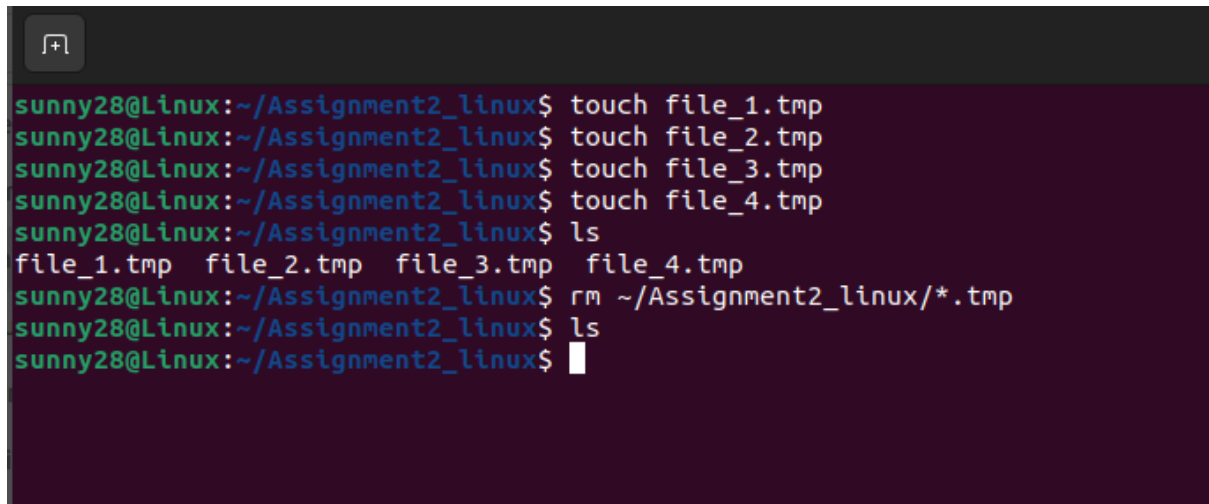
`rm`: This is the command for removing or deleting files.

`path/to/files/`: Replace this with the path to the directory where the files are located.

`*.pattern`: Replace "pattern" with the specific pattern you want to match. The asterisk (*) is a wildcard that matches any sequence of characters.

For example, if you want to delete all `.tmp` files in a directory named "myfiles" located in your home directory, you would run:

rm ~/Assignment2_linux/*.tmp

A terminal window with a dark purple background and a window control icon in the top-left corner. The terminal shows a series of commands and their outputs. The user creates four temporary files (file_1.tmp to file_4.tmp) using the 'touch' command. Then, they list the files with 'ls', which shows all four files. Finally, they run 'rm ~/Assignment2_linux/*.tmp' to delete all files, followed by another 'ls' command which shows no output, indicating the files have been successfully removed.

```
sunny28@Linux:~/Assignment2_linux$ touch file_1.tmp
sunny28@Linux:~/Assignment2_linux$ touch file_2.tmp
sunny28@Linux:~/Assignment2_linux$ touch file_3.tmp
sunny28@Linux:~/Assignment2_linux$ touch file_4.tmp
sunny28@Linux:~/Assignment2_linux$ ls
file_1.tmp  file_2.tmp  file_3.tmp  file_4.tmp
sunny28@Linux:~/Assignment2_linux$ rm ~/Assignment2_linux/*.tmp
sunny28@Linux:~/Assignment2_linux$ ls
sunny28@Linux:~/Assignment2_linux$
```

This prompts you for confirmation before deleting each file.