**Assignment on Basic Linux Commands**

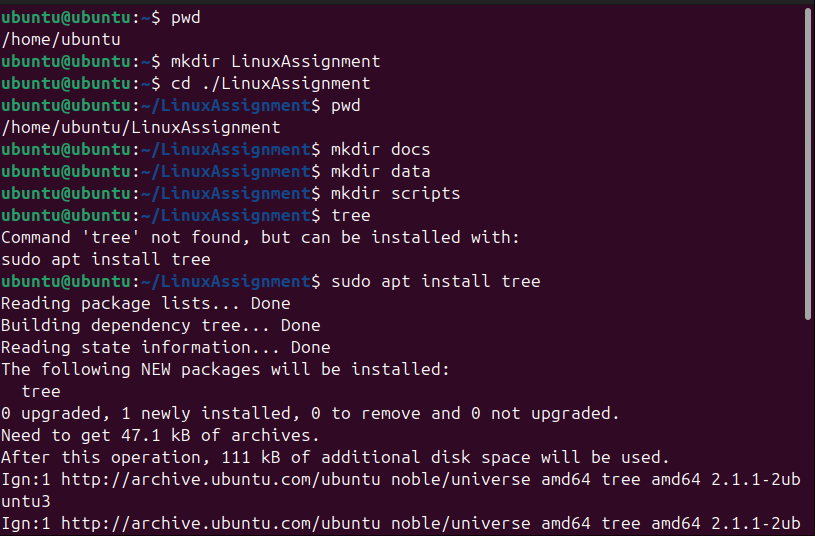
**Kanishka Anand – 24122016**

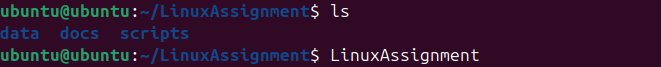
**Objective**

To gain practical understanding of essential Linux commands for file management, system navigation, user and process handling, and system information retrieval.

**Part A: File and Directory Management**

1. Create a new directory structure as shown below:  
   ~/LinuxAssignment/  
        ├── docs/  
        ├── data/  
        └── scripts/  
   (Commands: mkdir, cd, ls)





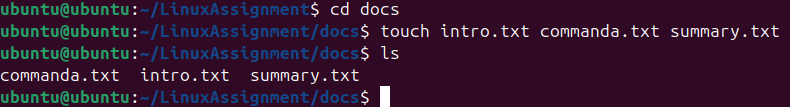
Explanation:

• **mkdir -p** creates directories and subdirectories.

• **cd** changes to the specified directory.

• **ls -R** lists directories recursively.

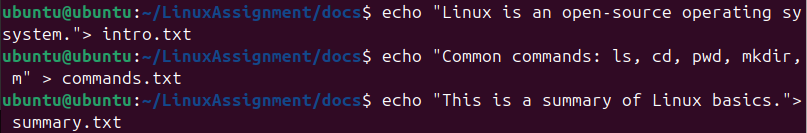
2. Inside docs/, create three files: intro.txt, commands.txt, and summary.txt. (Command: touch)



Explanation:

* **cd docs**- Changes the current directory to **docs/**.
* touch intro.txt commands.txt summary.txt - **touch** creates new, empty files named **intro.txt**, **commands.txt**, and **summary.txt**.
* **ls** - Lists the contents of **docs/**, confirming the files were created.

1. Write a few lines into each file using echo or cat.



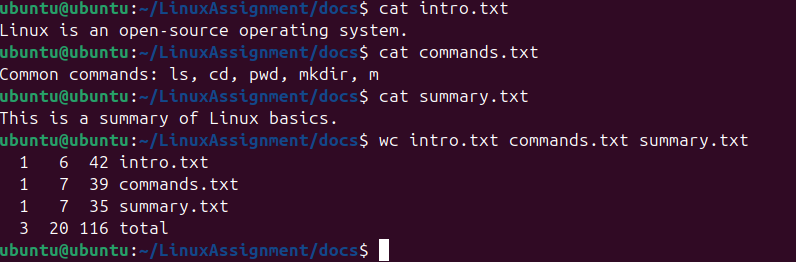
Explanation:

Writes/overwrites the specified string into **intro.txt**.

Writes/overwrites the specified string into **commands.txt.**

Writes/overwrites the specified string into **summary.txt.**

4. Display file contents and word count for each file. (Commands: cat, wc)



Explanation:

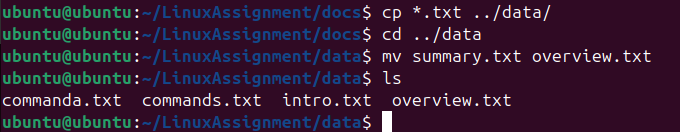
**cat intro.txt** - Displays the content of intro.txt

**cat commands.txt** - Displays the content of commands.txt

**cat summary.txt** - Displays the content of summary.txt

**wc intro.txt commands.txt summary.txt** - Displays the word count, line count, and byte count for the files.

5. Copy all files from docs/ to data/ and rename one file. (Commands: cp, mv)



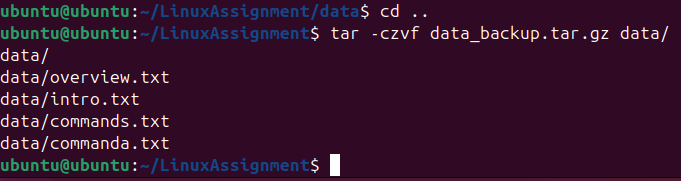
Explanation:

**cp \*.txt ../data/** - Copies all files ending in .txt to the data/ subdirectory in the parent directory (..).

**cd ../data** - Changes the directory to data/.

**mv summary.txt overview.txt** - Renames summary.txt to overview.txt.

6. Compress the data/ folder into a .tar.gz file. (Commands: tar, gzip)

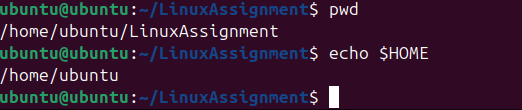


Explanation:

**tar -czvf data\_backup.tar.gz data/** - Compresses the data/ folder into a gzipped tar archive (.tar.gz).

**Part B: System Navigation & Permissions**

1. Display the current working directory and path to your home directory. (Commands: pwd, echo $HOME)

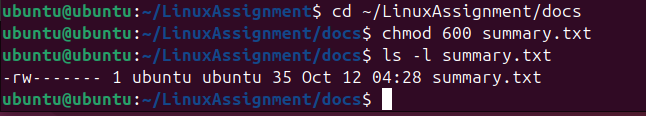


Explanation:

**Pwd** - Displays the current working directory.

**echo $HOME** - Displays the path to the home directory.

1. Change file permissions for summary.txt so that only the owner can read/write it. (Command: chmod 600 summary.txt)

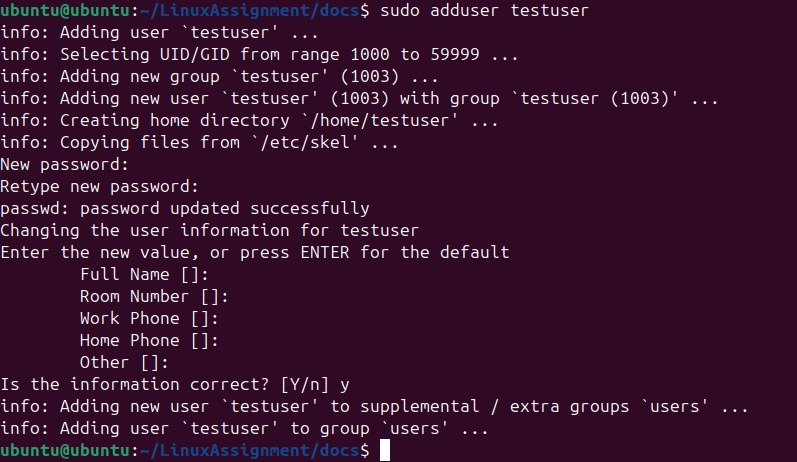


Explanation:

**chmod 600 summary.txt -** Changes file permissions so only the owner can read/write.

**ls -l summary.txt -** Lists the file details in long format to show the new permissions.

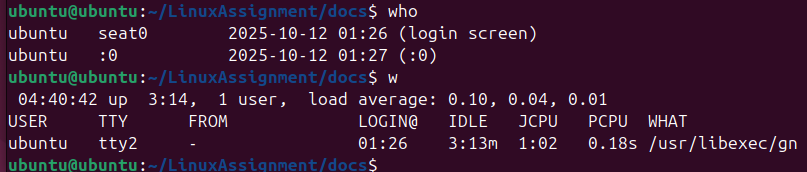
1. Create a new user (simulation using sudo adduser testuser – if not allowed, explain the process).



Explanation:

**sudo adduser testuser** - Creates a new user account named **testuser**.

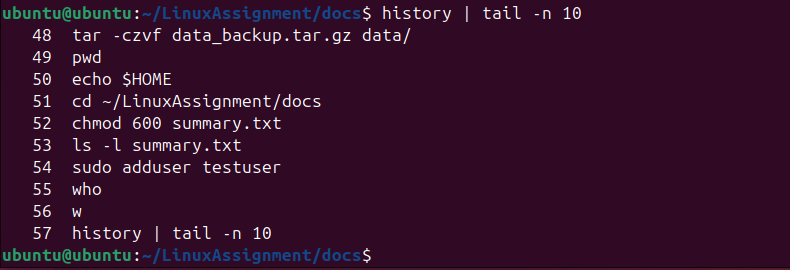
4. Display the list of users currently logged in. (Commands: who, w)



Explanation:

**Who** - Displays a list of users currently logged in.

**W** - Displays a list of users currently logged in and what they are doing.

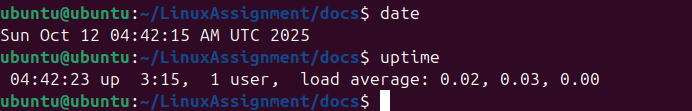
5. Display the 10 most recent commands you executed. (Command: history | tail -n 10)

Explanation:

Displays the 10 most recent commands executed.

**Part C: Process and System Monitoring**

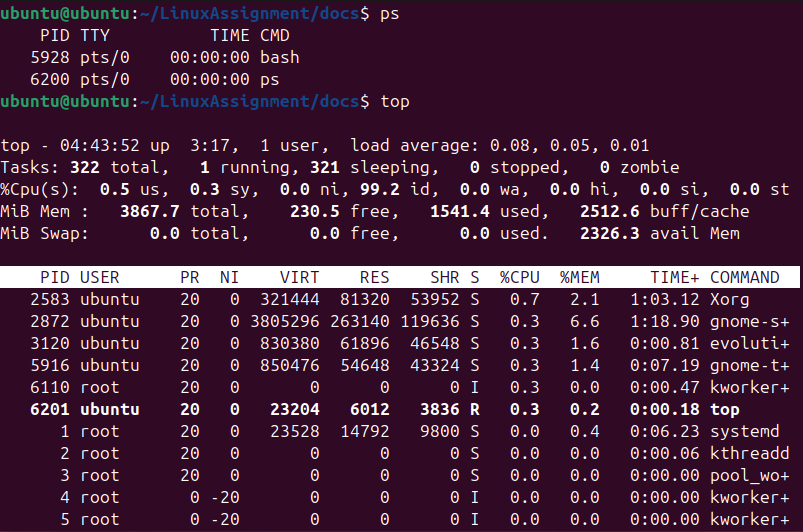
1. Display current date, time, and system uptime. (Commands: date, uptime)



Explanation:

**Date** - Displays the current system date and time.

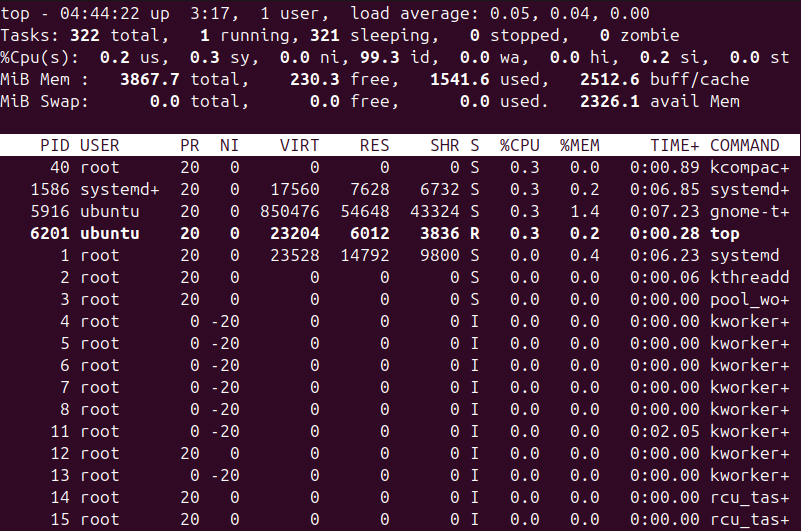
**Uptime** - Displays how long the system has been running.

2. Show currently running processes. (Commands: ps, top, htop)

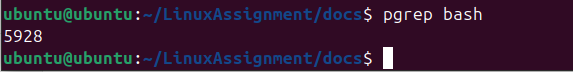
Explanation:

**Ps** - Displays a snapshot of the current processes.

**Top** - Displays a dynamic, real-time view of running processes and system resources.



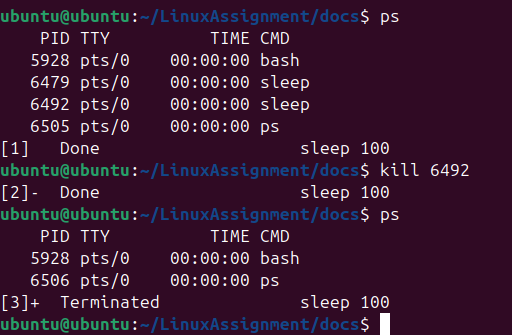
3. Find the process ID (PID) of any running process (e.g., “bash” or “firefox”). (Command: pgrep or ps -ef | grep)



Explanation:

**pgrep bash** - Finds and prints the Process ID (PID) of the **bash** process.

4. Kill a dummy process using its PID (can use sleep 100 & to create one). (Command: kill)

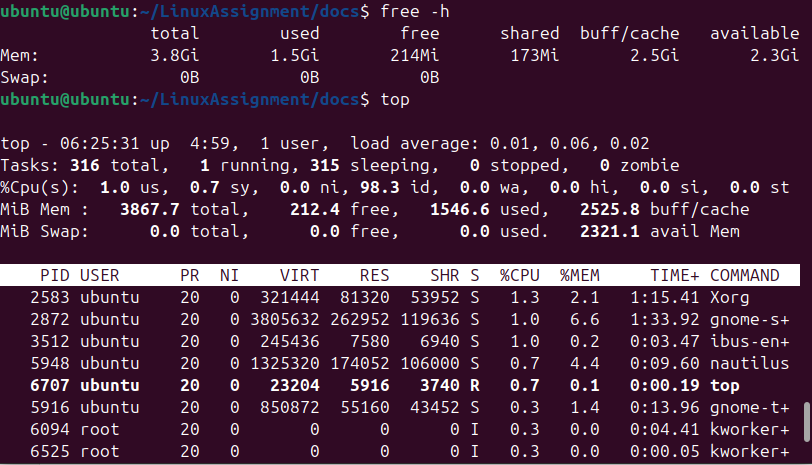


Explanation:

**sleep 100 & -** Starts a **sleep** process in the background (used to create a dummy process).

**Ps -** Displays a snapshot of the current processes.

5. Check the memory and CPU usage of the system. (Commands: free -h, top)



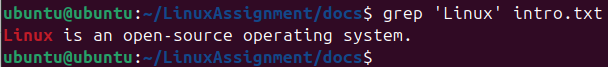
Explanation:

**free -h** - Displays used and free memory/swap in a human-readable format

**top** - Displays a dynamic, real-time view of running processes and system resources.

**Part D: Searching, Filtering & Redirection**

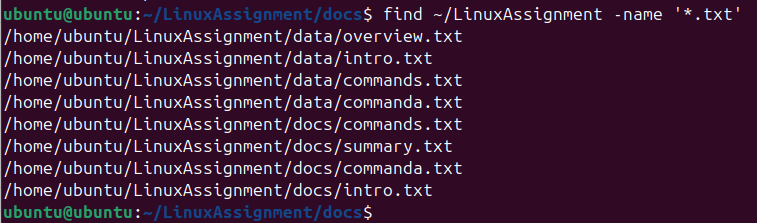
1. Use grep to find occurrences of the word “Linux” in any text file. (Command: grep 'Linux' filename)



Explanation:

**grep 'Linux' intro.txt -** Searches for the word **"Linux"** in intro.txt.

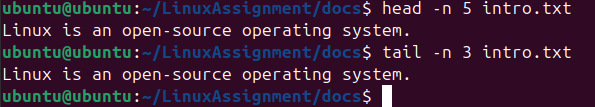
2. Use find to locate all .txt files in your LinuxAssignment directory. (Command: find ~/LinuxAssignment -name '\*.txt')



Explanation:

**find ~/LinuxAssignment -name '\*.txt' -** Locates all files ending in .txt within the LinuxAssignment directory.

3. Display the first 5 lines of a text file and then the last 3 lines. (Commands: head, tail)

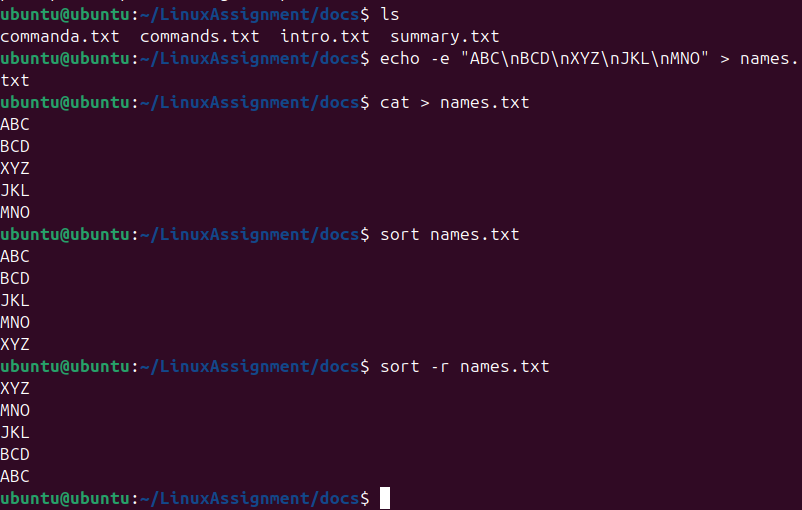


Explanation:

**head -n 5 intro.txt -** Displays the first 5 lines of intro.txt

**tail -n 3 intro.txt -** Displays the last 3 lines of intro.txt

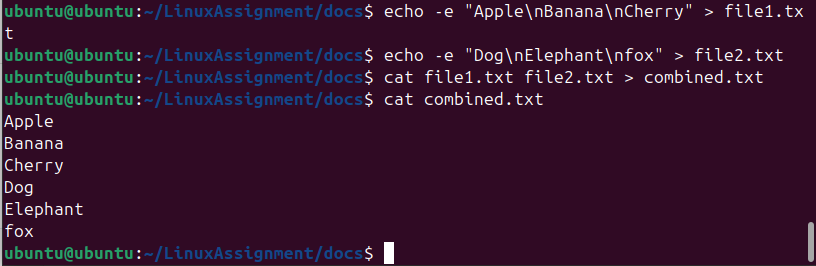
4. Sort a list of names stored in a file in ascending and descending order. (Command: sort, sort -r)

 Explanation:

**sort names.txt** - Sorts the lines in names.txt in ascending order.

**sort -r names.txt** - Sorts the lines in names.txt in **r**everse (descending) order.

5. Combine outputs of two text files into a single file using redirection. (Command: cat file1 file2 > combined.txt)

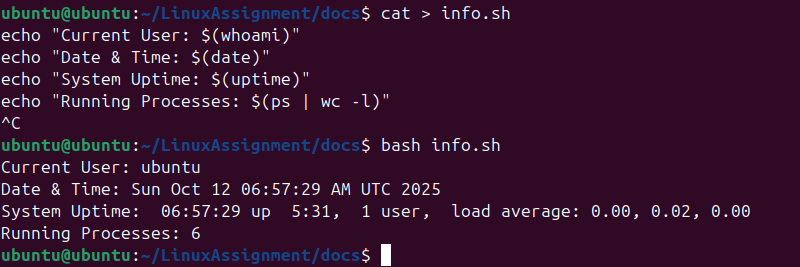


Explanation:

**cat file1.txt file2.txt > combined.txt** - Concatenates the contents of file1.txt and file2.txt and redirects the output to combined.txt

**Bonus Task**

Write a simple bash script (info.sh) that prints the following system information: - Current username - Date and time - System uptime - Number of running processes (Use commands: whoami, date, uptime, ps | wc -l)



Explanation:

**bash info.sh** - Executes the script **info.sh** using the **bash** interpreter.

**cat > info.sh** - Creates the script file info.sh for input.