Assignment on Basic Linux Commands

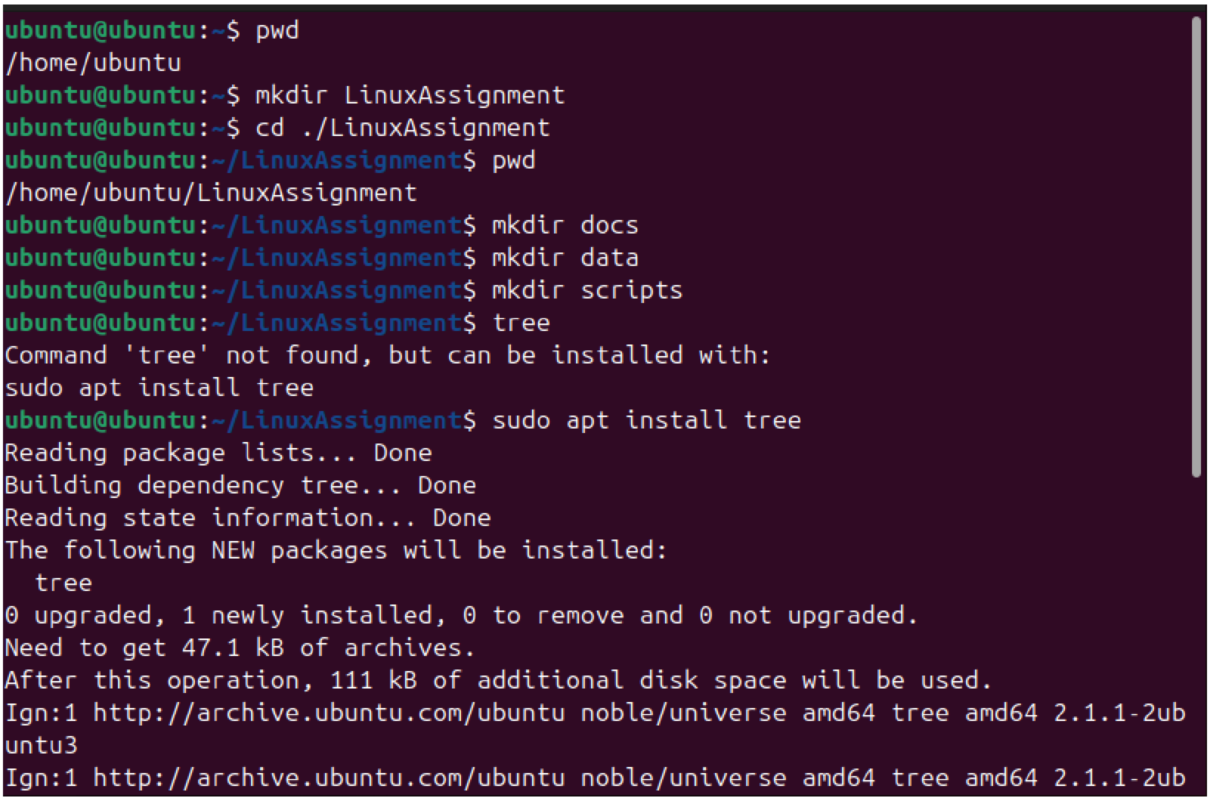
Kanishka Anand – 24122016

Objective

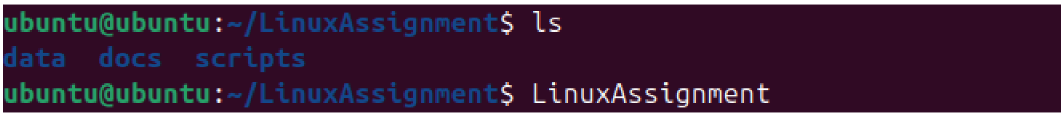
To gain practical understanding of essential Linux commands for file management, system navigation, user and process handling, and system information retrieval.

Part A: File and Directory Management

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



```
ubuntu@ubuntu:~$ pwd
/home/ubuntu
ubuntu@ubuntu:~$ mkdir LinuxAssignment
ubuntu@ubuntu:~$ cd ./LinuxAssignment
ubuntu@ubuntu:~/LinuxAssignment$ pwd
/home/ubuntu/LinuxAssignment
ubuntu@ubuntu:~/LinuxAssignment$ mkdir docs
ubuntu@ubuntu:~/LinuxAssignment$ mkdir data
ubuntu@ubuntu:~/LinuxAssignment$ mkdir scripts
ubuntu@ubuntu:~/LinuxAssignment$ tree
Command 'tree' not found, but can be installed with:
sudo apt install tree
ubuntu@ubuntu:~/LinuxAssignment$ sudo apt install tree
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 47.1 kB of archives.
After this operation, 111 kB of additional disk space will be used.
Ign:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 tree amd64 2.1.1-2ub
untu3
Ign:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 tree amd64 2.1.1-2ub
```

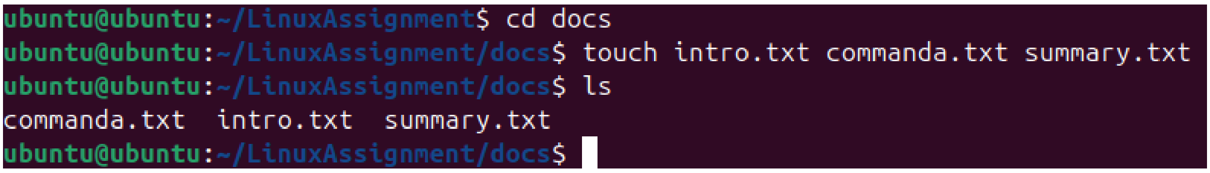


```
ubuntu@ubuntu:~/LinuxAssignment$ ls
data  docs  scripts
ubuntu@ubuntu:~/LinuxAssignment$ LinuxAssignment
```

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)

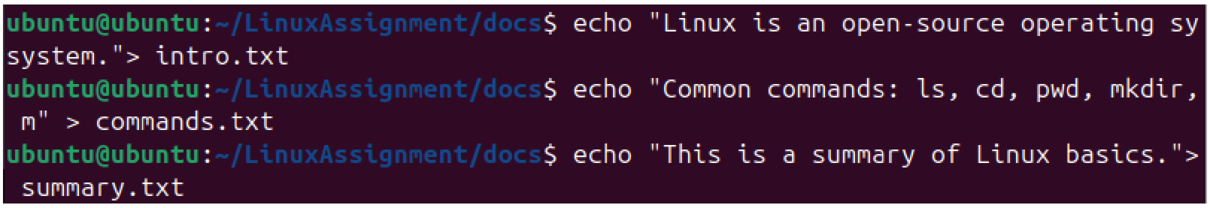


```
ubuntu@ubuntu:~/LinuxAssignment$ cd docs
ubuntu@ubuntu:~/LinuxAssignment/docs$ touch intro.txt commanda.txt summary.txt
ubuntu@ubuntu:~/LinuxAssignment/docs$ ls
commanda.txt  intro.txt  summary.txt
ubuntu@ubuntu:~/LinuxAssignment/docs$ 
```

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.

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



```
ubuntu@ubuntu:~/LinuxAssignment/docs$ echo "Linux is an open-source operating sy
system."> intro.txt
ubuntu@ubuntu:~/LinuxAssignment/docs$ echo "Common commands: ls, cd, pwd, mkdir,
 m" > commands.txt
ubuntu@ubuntu:~/LinuxAssignment/docs$ echo "This is a summary of Linux basics.">
 summary.txt
```

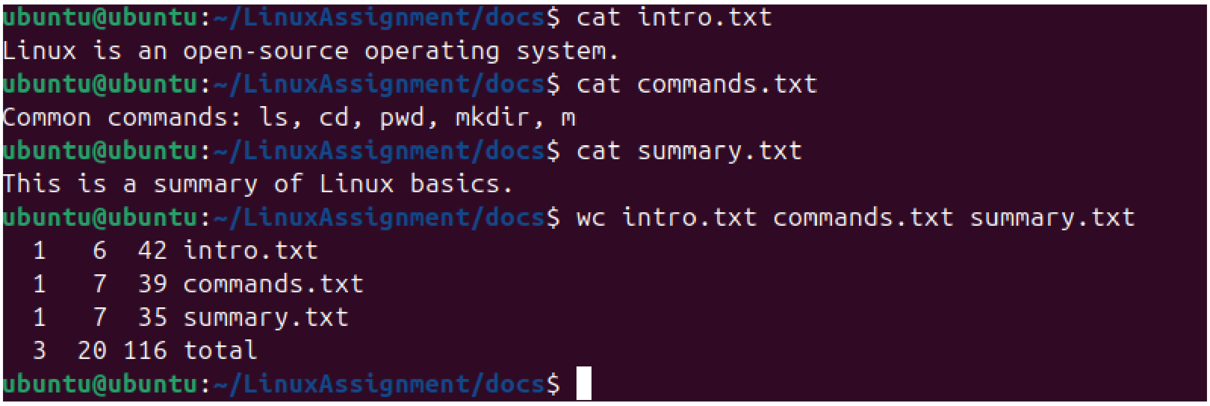
Explanation:

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)



```
ubuntu@ubuntu:~/LinuxAssignment/docs$ cat intro.txt
Linux is an open-source operating system.
ubuntu@ubuntu:~/LinuxAssignment/docs$ cat commands.txt
Common commands: ls, cd, pwd, mkdir, m
ubuntu@ubuntu:~/LinuxAssignment/docs$ cat summary.txt
This is a summary of Linux basics.
ubuntu@ubuntu:~/LinuxAssignment/docs$ wc intro.txt commands.txt summary.txt
  1   6  42 intro.txt
  1   7  39 commands.txt
  1   7  35 summary.txt
  3  20 116 total
ubuntu@ubuntu:~/LinuxAssignment/docs$ 
```

Explanation:

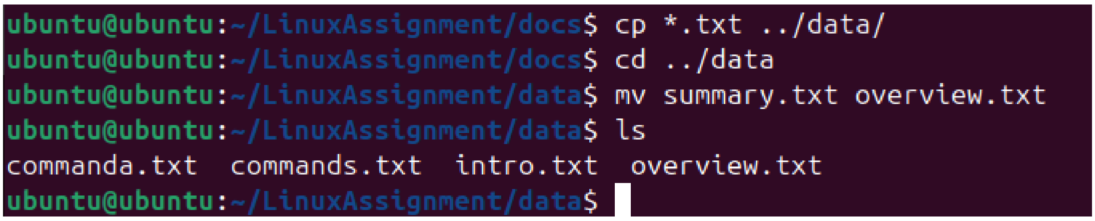
cat intro.txt - Displays the content of intro.txt

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)



```
ubuntu@ubuntu:~/LinuxAssignment/docs$ cp *.txt ../data/
ubuntu@ubuntu:~/LinuxAssignment/docs$ cd ../data
ubuntu@ubuntu:~/LinuxAssignment/data$ mv summary.txt overview.txt
ubuntu@ubuntu:~/LinuxAssignment/data$ ls
commanda.txt  commands.txt  intro.txt  overview.txt
ubuntu@ubuntu:~/LinuxAssignment/data$ 
```

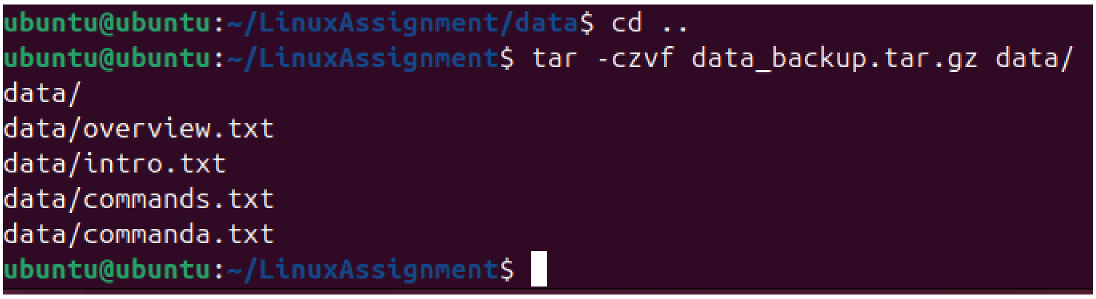
Explanation:

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)



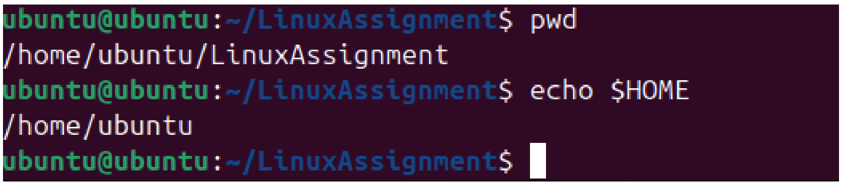
```
ubuntu@ubuntu:~/LinuxAssignment/data$ cd ..
ubuntu@ubuntu:~/LinuxAssignment$ tar -czvf data_backup.tar.gz data/
data/
data/overview.txt
data/intro.txt
data/commands.txt
data/commanda.txt
ubuntu@ubuntu:~/LinuxAssignment$ 
```

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)



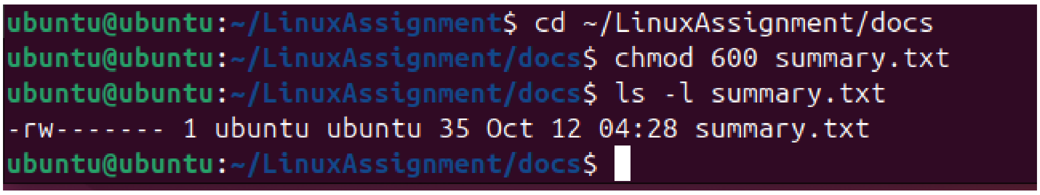
```
ubuntu@ubuntu:~/LinuxAssignment$ pwd
/home/ubuntu/LinuxAssignment
ubuntu@ubuntu:~/LinuxAssignment$ echo $HOME
/home/ubuntu
ubuntu@ubuntu:~/LinuxAssignment$ 
```

Explanation:

Pwd - Displays the current working directory.

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

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



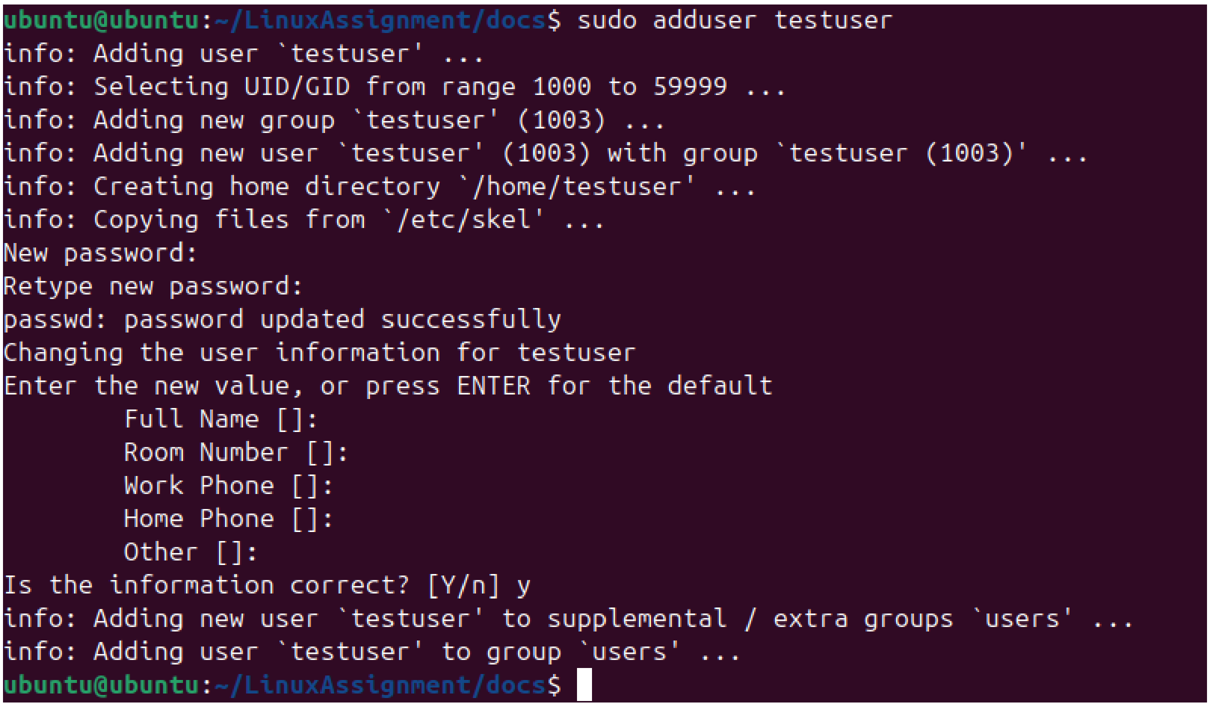
```
ubuntu@ubuntu:~/LinuxAssignment$ cd ~/LinuxAssignment/docs
ubuntu@ubuntu:~/LinuxAssignment/docs$ chmod 600 summary.txt
ubuntu@ubuntu:~/LinuxAssignment/docs$ ls -l summary.txt
-rw------- 1 ubuntu ubuntu 35 Oct 12 04:28 summary.txt
ubuntu@ubuntu:~/LinuxAssignment/docs$ 
```

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.

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

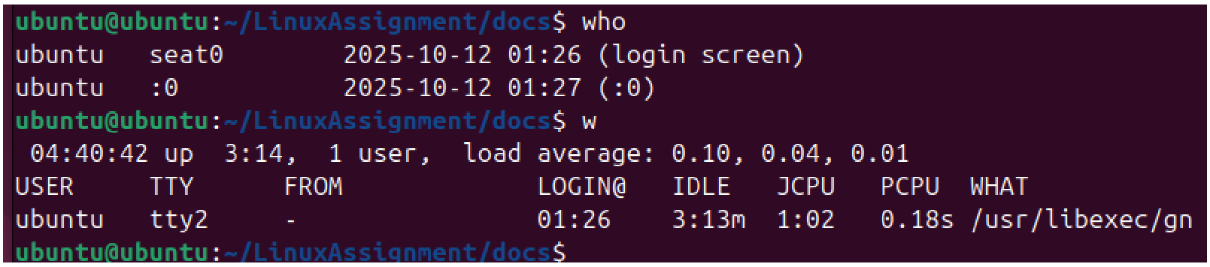


```
ubuntu@ubuntu:~/LinuxAssignment/docs$ sudo adduser testuser
info: Adding user `testuser' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `testuser' (1003) ...
info: Adding new user `testuser' (1003) with group `testuser (1003)' ...
info: Creating home directory `/home/testuser' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for testuser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
info: Adding new user `testuser' to supplemental / extra groups `users' ...
info: Adding user `testuser' to group `users' ...
ubuntu@ubuntu:~/LinuxAssignment/docs$ 
```

Explanation:

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

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



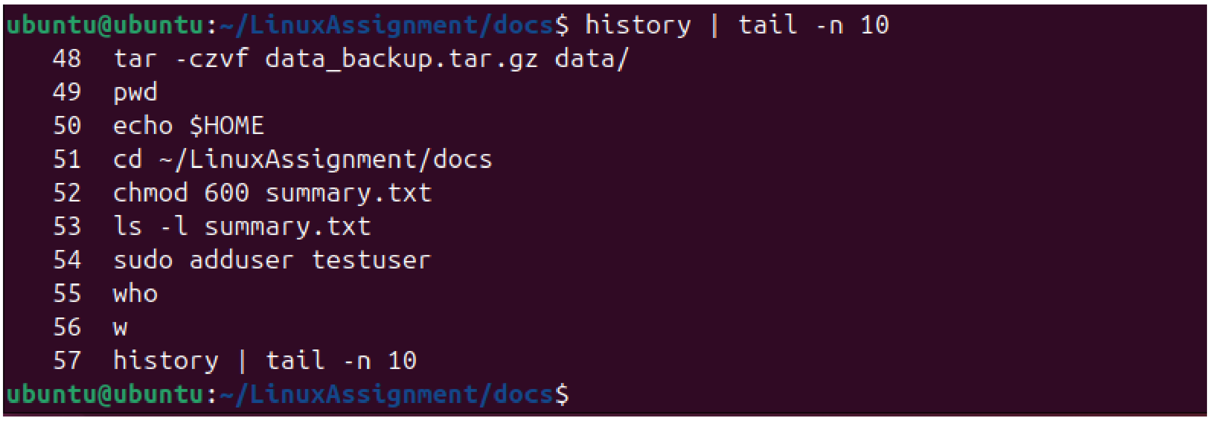
```
ubuntu@ubuntu:~/LinuxAssignment/docs$ who
ubuntu   seat0        2025-10-12 01:26 (login screen)
ubuntu   :0           2025-10-12 01:27 (:0)
ubuntu@ubuntu:~/LinuxAssignment/docs$ w
 04:40:42 up  3:14,  1 user,  load average: 0.10, 0.04, 0.01
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU  WHAT
ubuntu   tty2     -                01:26    3:13m  1:02   0.18s /usr/libexec/gn
ubuntu@ubuntu:~/LinuxAssignment/docs$
```

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)



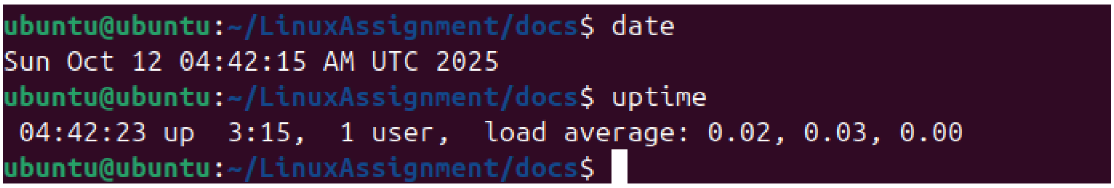
```
ubuntu@ubuntu:~/LinuxAssignment/docs$ history | tail -n 10
   48  tar -czvf data_backup.tar.gz data/
   49  pwd
   50  echo $HOME
   51  cd ~/LinuxAssignment/docs
   52  chmod 600 summary.txt
   53  ls -l summary.txt
   54  sudo adduser testuser
   55  who
   56  w
   57  history | tail -n 10
ubuntu@ubuntu:~/LinuxAssignment/docs$
```

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)



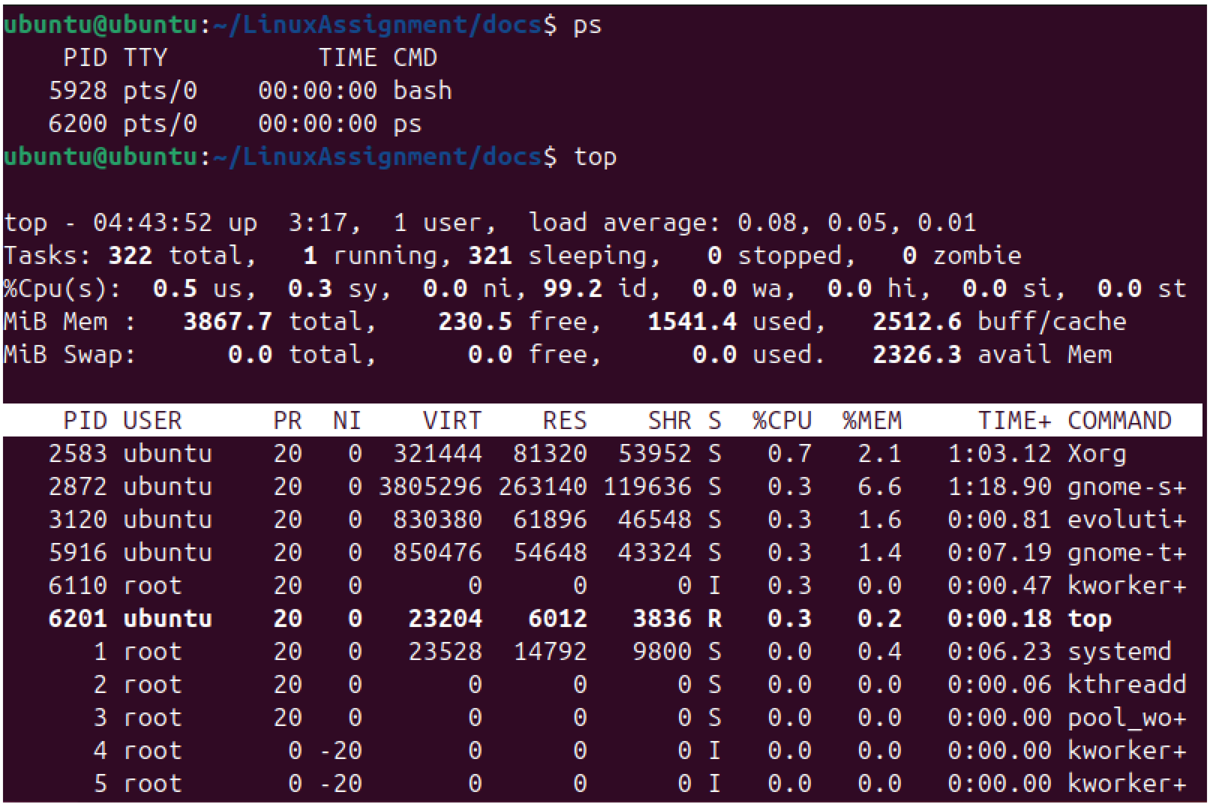
```
ubuntu@ubuntu:~/LinuxAssignment/docs$ date
Sun Oct 12 04:42:15 AM UTC 2025
ubuntu@ubuntu:~/LinuxAssignment/docs$ uptime
 04:42:23 up  3:15,  1 user,  load average: 0.02, 0.03, 0.00
ubuntu@ubuntu:~/LinuxAssignment/docs$
```

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)

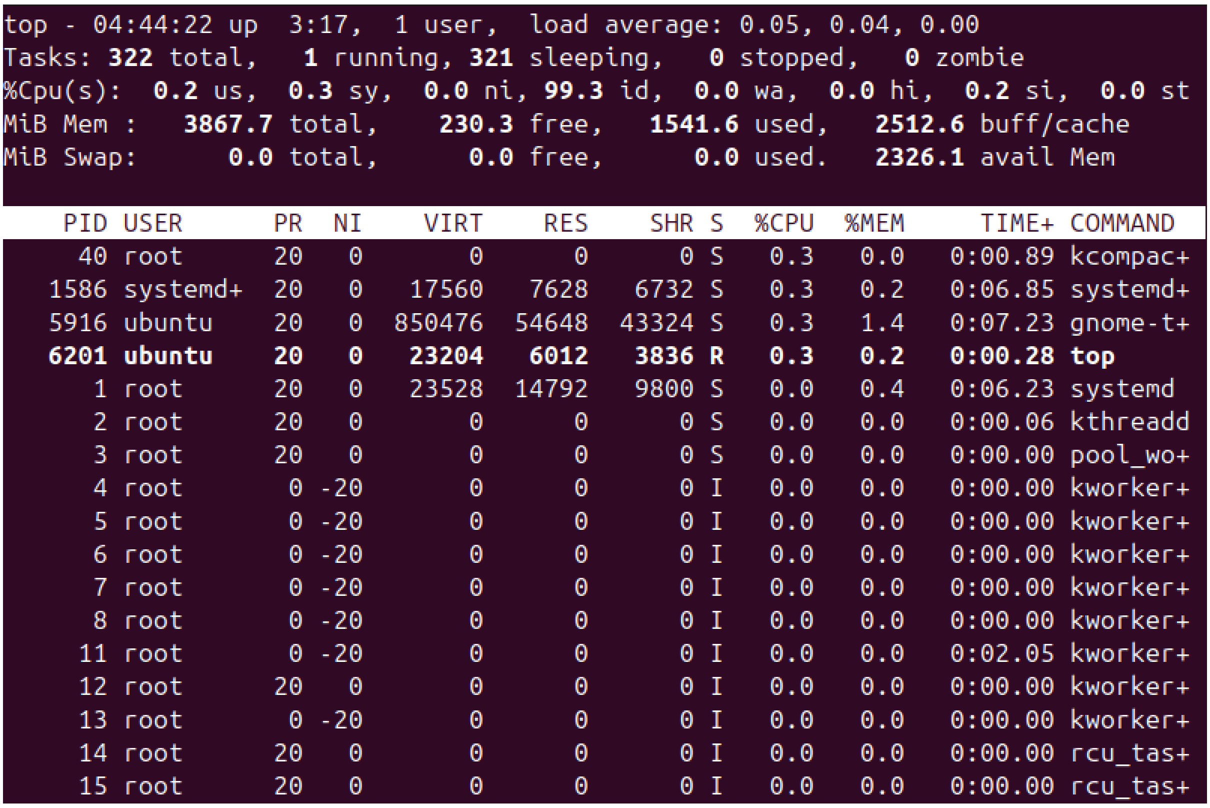


```
ubuntu@ubuntu:~/LinuxAssignment/docs$ ps
    PID TTY          TIME CMD
   5928 pts/0    00:00:00 bash
   6200 pts/0    00:00:00 ps
ubuntu@ubuntu:~/LinuxAssignment/docs$ top

top - 04:43:52 up  3:17,  1 user,  load average: 0.08, 0.05, 0.01
Tasks: 322 total,   1 running, 321 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.5 us,  0.3 sy,  0.0 ni, 99.2 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   3867.7 total,    230.5 free,   1541.4 used,   2512.6 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.   2326.3 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   2583 ubuntu    20   0  321444  81320  53952 S   0.7   2.1   1:03.12 Xorg
   2872 ubuntu    20   0 3805296 263140 119636 S   0.3   6.6   1:18.90 gnome-s+
   3120 ubuntu    20   0  830380  61896  46548 S   0.3   1.6   0:00.81 evoluti+
   5916 ubuntu    20   0  850476  54648  43324 S   0.3   1.4   0:07.19 gnome-t+
   6110 root      20   0       0      0      0 I   0.3   0.0   0:00.47 kworker+
   6201 ubuntu    20   0   23204   6012   3836 R   0.3   0.2   0:00.18 top
      1 root      20   0   23528  14792   9800 S   0.0   0.4   0:06.23 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.06 kthreadd
      3 root      20   0       0      0      0 S   0.0   0.0   0:00.00 pool_wo+
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
```

Explanation:

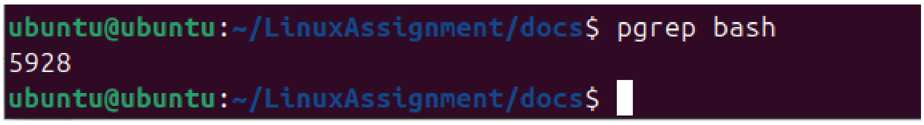
Ps - Displays a snapshot of the current processes.

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



```
top - 04:44:22 up  3:17,  1 user,  load average: 0.05, 0.04, 0.00
Tasks: 322 total,   1 running, 321 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.2 us,  0.3 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.2 si,  0.0 st
MiB Mem :   3867.7 total,    230.3 free,   1541.6 used,   2512.6 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.   2326.1 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
     40 root      20   0       0      0      0 S   0.3   0.0   0:00.89 kcompac+
   1586 systemd+  20   0   17560   7628   6732 S   0.3   0.2   0:06.85 systemd+
   5916 ubuntu    20   0  850476  54648  43324 S   0.3   1.4   0:07.23 gnome-t+
   6201 ubuntu    20   0   23204   6012   3836 R   0.3   0.2   0:00.28 top
      1 root      20   0   23528  14792   9800 S   0.0   0.4   0:06.23 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.06 kthreadd
      3 root      20   0       0      0      0 S   0.0   0.0   0:00.00 pool_wo+
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
      7 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
     11 root       0 -20       0      0      0 I   0.0   0.0   0:02.05 kworker+
     12 root      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker+
     13 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker+
     14 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tas+
     15 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tas+
```

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

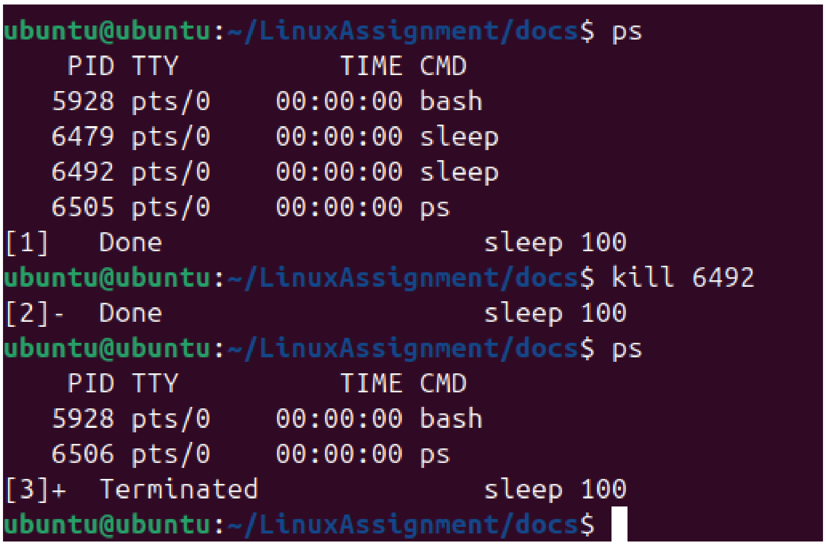


```
ubuntu@ubuntu:~/LinuxAssignment/docs$ pgrep bash
5928
ubuntu@ubuntu:~/LinuxAssignment/docs$ 
```

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)



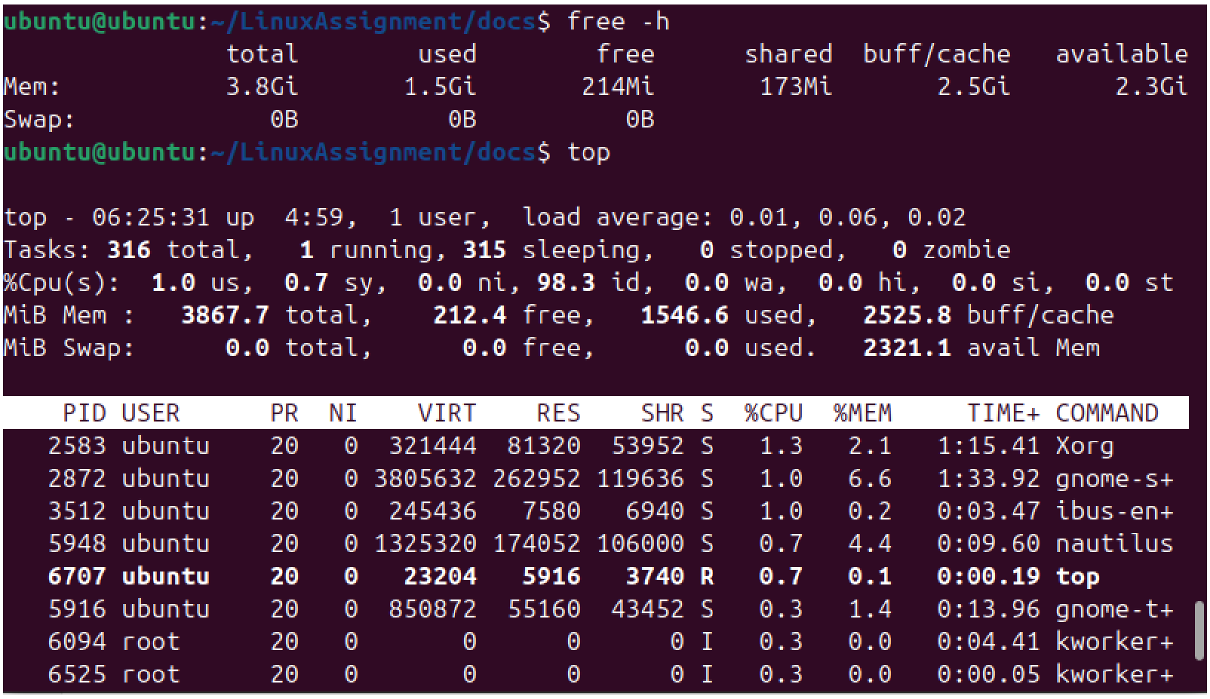
```
ubuntu@ubuntu:~/LinuxAssignment/docs$ ps
    PID TTY          TIME CMD
   5928 pts/0    00:00:00 bash
   6479 pts/0    00:00:00 sleep
   6492 pts/0    00:00:00 sleep
   6505 pts/0    00:00:00 ps
[1]   Done                    sleep 100
ubuntu@ubuntu:~/LinuxAssignment/docs$ kill 6492
[2]-  Done                    sleep 100
ubuntu@ubuntu:~/LinuxAssignment/docs$ ps
    PID TTY          TIME CMD
   5928 pts/0    00:00:00 bash
   6506 pts/0    00:00:00 ps
[3]+  Terminated              sleep 100
ubuntu@ubuntu:~/LinuxAssignment/docs$ 
```

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)



```
ubuntu@ubuntu:~/LinuxAssignment/docs$ free -h
               total        used        free      shared  buff/cache   available
Mem:           3.8Gi       1.5Gi       214Mi       173Mi       2.5Gi       2.3Gi
Swap:             0B          0B          0B
ubuntu@ubuntu:~/LinuxAssignment/docs$ top

top - 06:25:31 up  4:59,  1 user,  load average: 0.01, 0.06, 0.02
Tasks: 316 total,   1 running, 315 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.0 us,  0.7 sy,  0.0 ni, 98.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   3867.7 total,    212.4 free,   1546.6 used,   2525.8 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.   2321.1 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   2583 ubuntu    20   0  321444  81320  53952 S   1.3   2.1   1:15.41 Xorg
   2872 ubuntu    20   0 3805632 262952 119636 S   1.0   6.6   1:33.92 gnome-s+
   3512 ubuntu    20   0  245436   7580   6940 S   1.0   0.2   0:03.47 ibus-en+
   5948 ubuntu    20   0 1325320 174052 106000 S   0.7   4.4   0:09.60 nautilus
   6707 ubuntu    20   0   23204   5916   3740 R   0.7   0.1   0:00.19 top
   5916 ubuntu    20   0  850872  55160  43452 S   0.3   1.4   0:13.96 gnome-t+
   6094 root      20   0       0      0      0 I   0.3   0.0   0:04.41 kworker+
   6525 root      20   0       0      0      0 I   0.3   0.0   0:00.05 kworker+
```

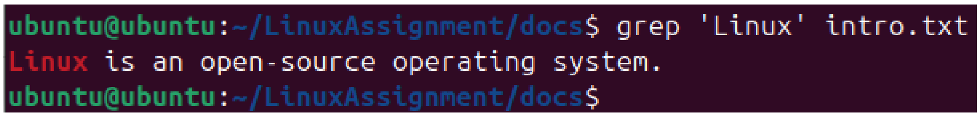
Explanation:

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)

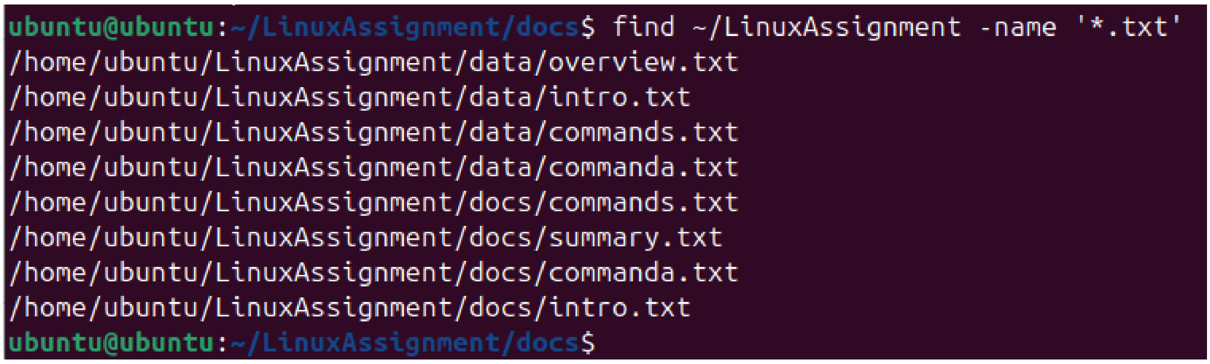


```
ubuntu@ubuntu:~/LinuxAssignment/docs$ grep 'Linux' intro.txt
Linux is an open-source operating system.
ubuntu@ubuntu:~/LinuxAssignment/docs$
```

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')

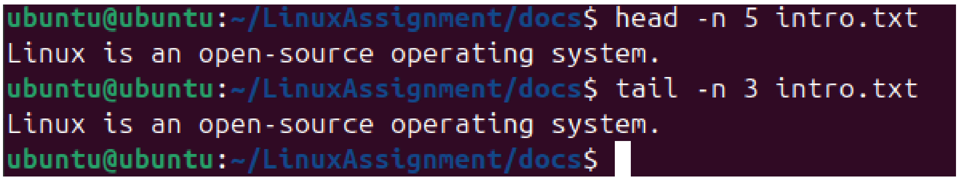


```
ubuntu@ubuntu:~/LinuxAssignment/docs$ find ~/LinuxAssignment -name '*.txt'
/home/ubuntu/LinuxAssignment/data/overview.txt
/home/ubuntu/LinuxAssignment/data/intro.txt
/home/ubuntu/LinuxAssignment/data/commands.txt
/home/ubuntu/LinuxAssignment/data/commanda.txt
/home/ubuntu/LinuxAssignment/docs/commands.txt
/home/ubuntu/LinuxAssignment/docs/summary.txt
/home/ubuntu/LinuxAssignment/docs/commanda.txt
/home/ubuntu/LinuxAssignment/docs/intro.txt
ubuntu@ubuntu:~/LinuxAssignment/docs$
```

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)



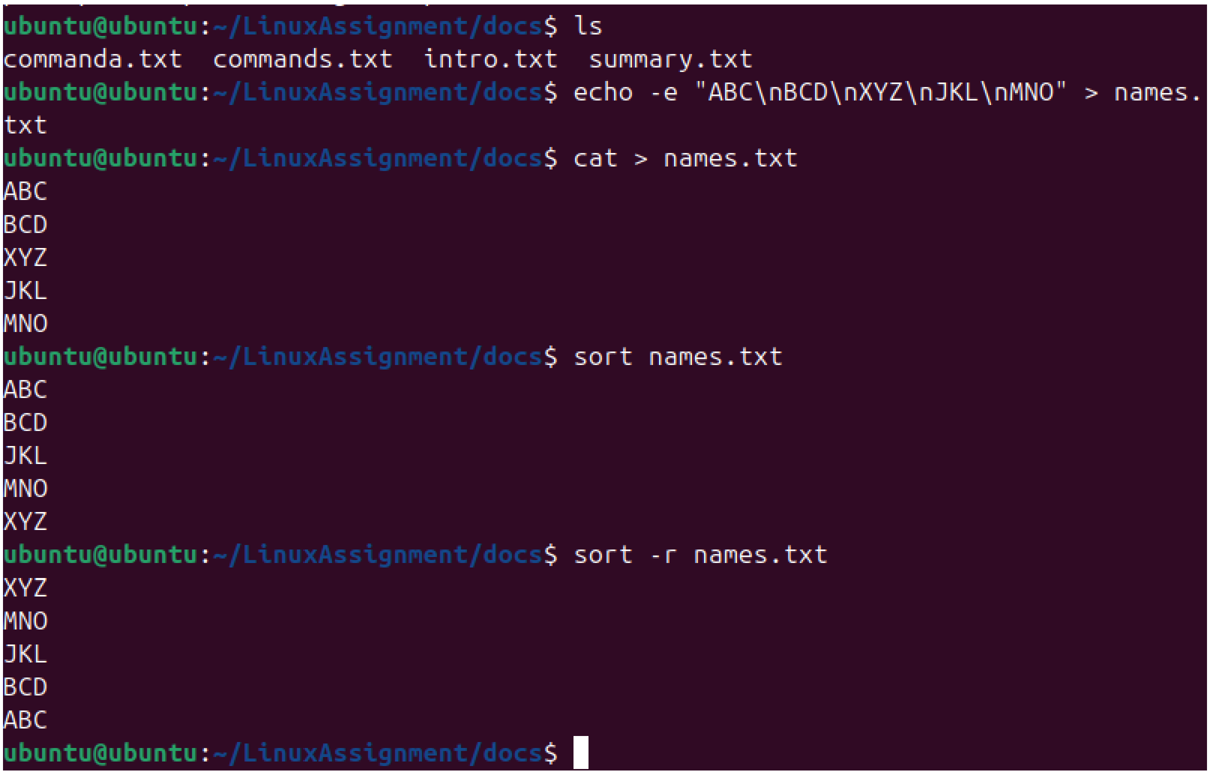
```
ubuntu@ubuntu:~/LinuxAssignment/docs$ head -n 5 intro.txt
Linux is an open-source operating system.
ubuntu@ubuntu:~/LinuxAssignment/docs$ tail -n 3 intro.txt
Linux is an open-source operating system.
ubuntu@ubuntu:~/LinuxAssignment/docs$
```

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)



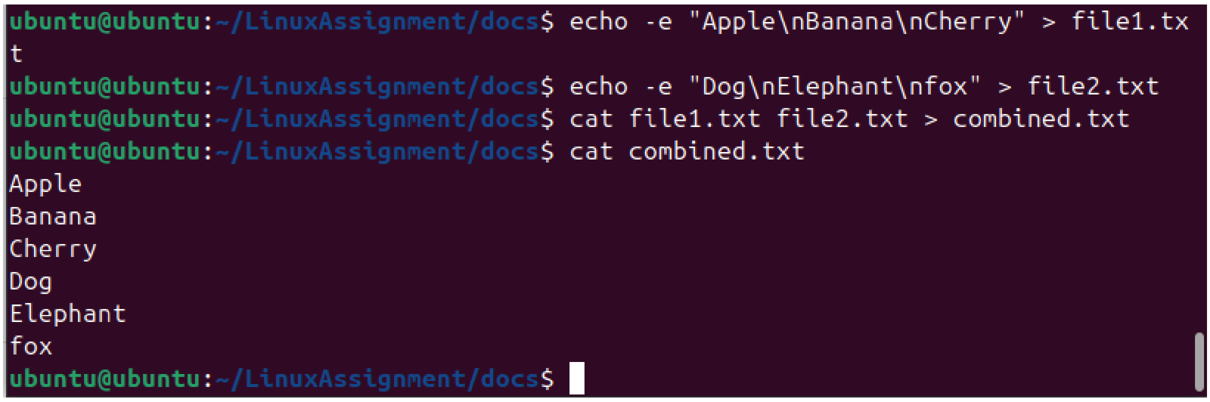
```
ubuntu@ubuntu:~/LinuxAssignment/docs$ ls
commanda.txt  commands.txt  intro.txt  summary.txt
ubuntu@ubuntu:~/LinuxAssignment/docs$ echo -e "ABC\nBCD\nXYZ\nJKL\nMNO" > names.
txt
ubuntu@ubuntu:~/LinuxAssignment/docs$ cat > names.txt
ABC
BCD
XYZ
JKL
MNO
ubuntu@ubuntu:~/LinuxAssignment/docs$ sort names.txt
ABC
BCD
JKL
MNO
XYZ
ubuntu@ubuntu:~/LinuxAssignment/docs$ sort -r names.txt
XYZ
MNO
JKL
BCD
ABC
ubuntu@ubuntu:~/LinuxAssignment/docs$ 
```

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)



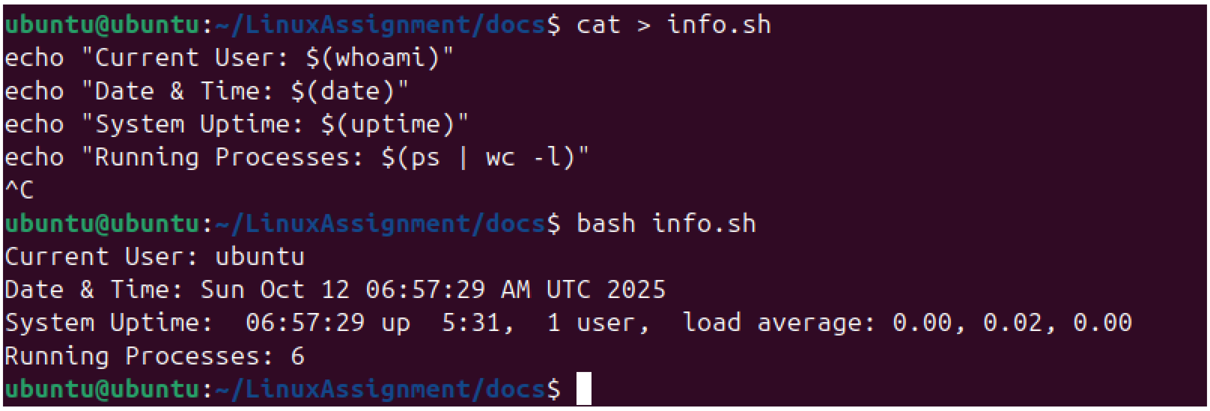
```
ubuntu@ubuntu:~/LinuxAssignment/docs$ echo -e "Apple\nBanana\nCherry" > file1.tx
t
ubuntu@ubuntu:~/LinuxAssignment/docs$ echo -e "Dog\nElephant\nfox" > file2.txt
ubuntu@ubuntu:~/LinuxAssignment/docs$ cat file1.txt file2.txt > combined.txt
ubuntu@ubuntu:~/LinuxAssignment/docs$ cat combined.txt
Apple
Banana
Cherry
Dog
Elephant
fox
ubuntu@ubuntu:~/LinuxAssignment/docs$ 
```

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)



```
ubuntu@ubuntu:~/LinuxAssignment/docs$ cat > info.sh
echo "Current User: $(whoami)"
echo "Date & Time: $(date)"
echo "System Uptime: $(uptime)"
echo "Running Processes: $(ps | wc -l)"
^C
ubuntu@ubuntu:~/LinuxAssignment/docs$ bash info.sh
Current User: ubuntu
Date & Time: Sun Oct 12 06:57:29 AM UTC 2025
System Uptime:  06:57:29 up  5:31,  1 user,  load average: 0.00, 0.02, 0.00
Running Processes: 6
ubuntu@ubuntu:~/LinuxAssignment/docs$ 
```

Explanation:

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

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