Execute various LINUX commands for:

i. **Information Maintenance:**
wc, clear, cal, who, date, pwd

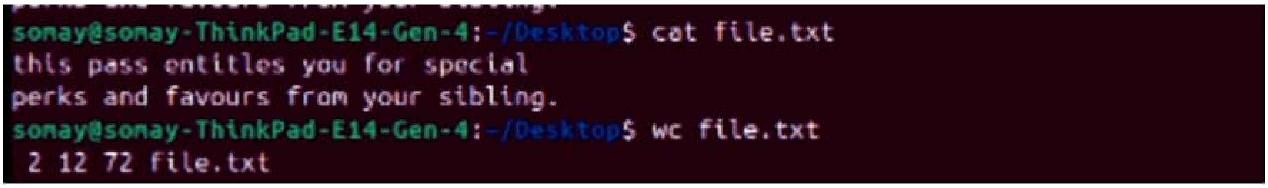
ii. **File Management:**
cat, cp, rm, mv, cmp, comm, diff, find, grep, awk

iii. **Directory Management:**
cd, mkdir, rmdir, ls

Information Maintenance:

1. wc - This command counts the number of lines, words, and bytes in a file.

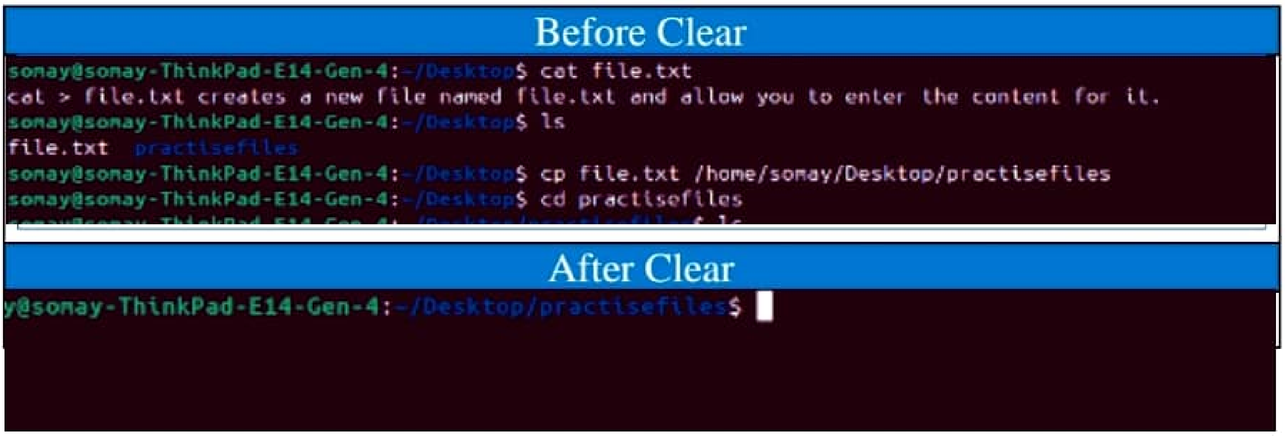
For example, wc file.txt will display the number of lines, words, and bytes in file.txt



```
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ cat file.txt
this pass entitles you for special
perks and favours from your sibling.
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ wc file.txt
 2 12 72 file.txt
```

2. clear - This command clears the terminal screen.

For example, clear will erase all the previous output and commands from the terminal



Before Clear

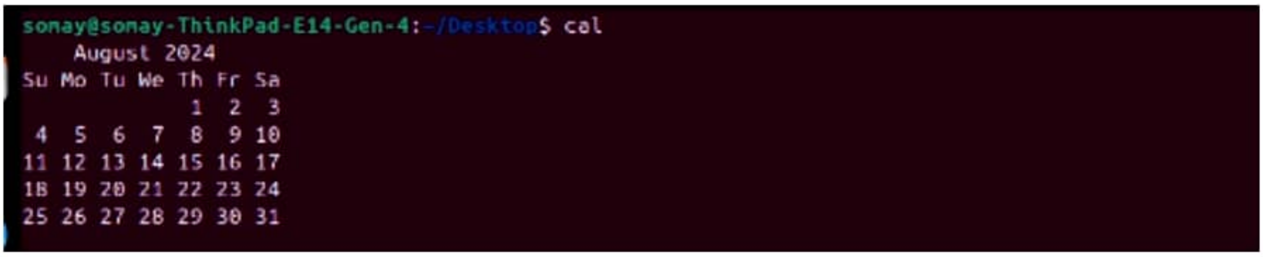
```
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ cat file.txt
cat > file.txt creates a new file named file.txt and allow you to enter the content for it.
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ ls
file.txt  practisefiles
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ cp file.txt /home/sonay/Desktop/practisefiles
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ cd practisefiles
```

After Clear

```
y@sonay-ThinkPad-E14-Gen-4:~/Desktop/practisefiles$ 
```

3. **cal -** This command displays a calendar of the current month or a specified month and year.

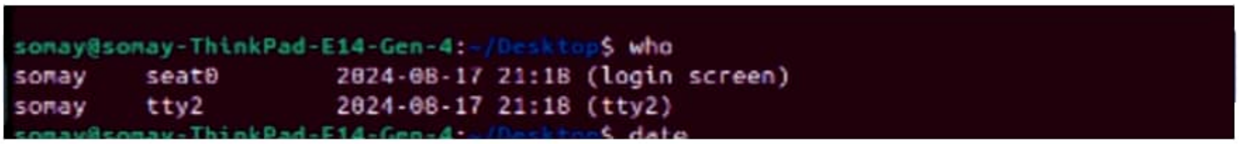
   For example, cal 10 2023 will display the calendar of October 2023.



```
somay@somay-ThinkPad-E14-Gen-4:~/Desktop$ cal
    August 2024
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

4. **who -** This command shows the information about the users who are currently logged in to the system.

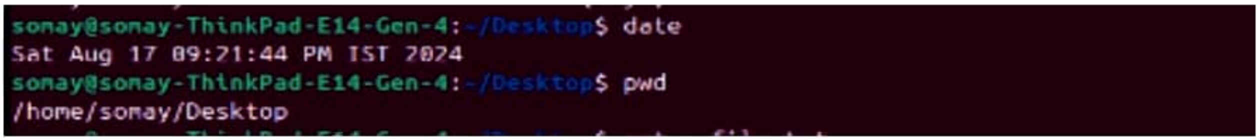
   For example, who will display the username, terminal, login time, and host name of each user.



```
somay@somay-ThinkPad-E14-Gen-4:~/Desktop$ who
somay    seat0        2024-08-17 21:18 (login screen)
somay    tty2         2024-08-17 21:18 (tty2)
somay@somay-ThinkPad-E14-Gen-4:~/Desktop$ date
```

5. **date -** This command displays the current date and time of the system.

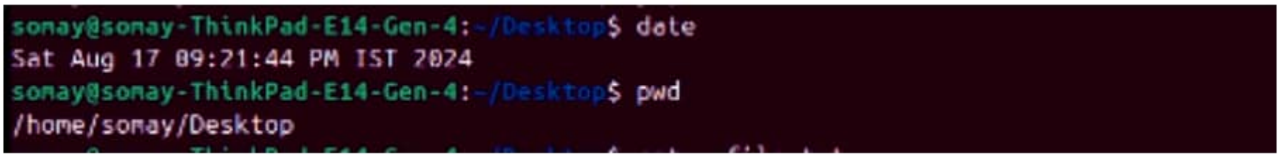
   For example, date will show something like Wed Oct 25 22:22:37 IST 2023



```
somay@somay-ThinkPad-E14-Gen-4:~/Desktop$ date
Sat Aug 17 09:21:44 PM IST 2024
somay@somay-ThinkPad-E14-Gen-4:~/Desktop$ pwd
/home/somay/Desktop
```

6. **pwd -** This command prints the name of the current working directory.

   For example, pwd will show something like /home/sudeep



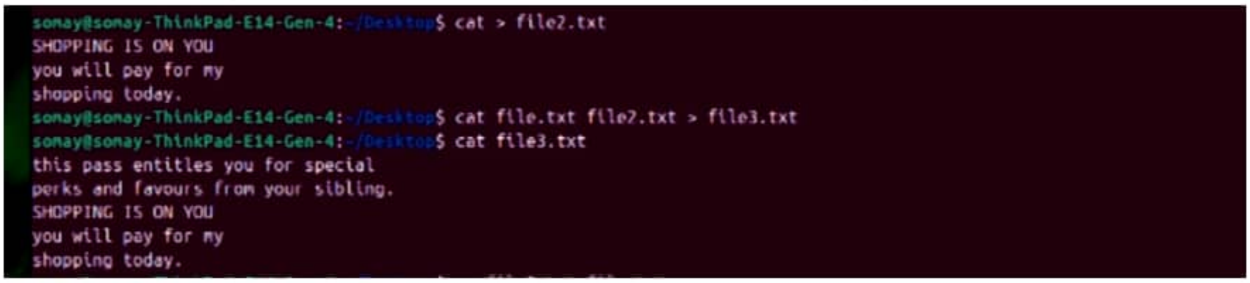
```
somay@somay-ThinkPad-E14-Gen-4:~/Desktop$ date
Sat Aug 17 09:21:44 PM IST 2024
somay@somay-ThinkPad-E14-Gen-4:~/Desktop$ pwd
/home/somay/Desktop
```

File Management:

1. **cat** - This command can be used to create, display, or concatenate files.

   For example,

   a. **cat > file.txt** will create a new file named file.txt and allow you to enter its content. Press Ctrl+D to save and exit.
   b. **cat file.txt** will display the content of file.txt
   c. **cat file.txt file2.txt > 1.txt** will concatenate file.txt and 1.txt and store the result in newfile.txt.

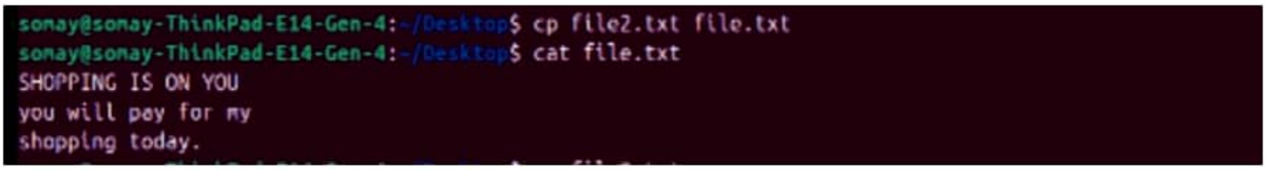


```
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ cat > file2.txt
SHOPPING IS ON YOU
you will pay for my
shopping today.
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ cat file.txt file2.txt > file3.txt
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ cat file3.txt
this pass entitles you for special
perks and favours from your sibling.
SHOPPING IS ON YOU
you will pay for my
shopping today.
```

2. **cp** - This command copies a file or directory from one location to another.

   For example, cp file.txt file2.txt will copy file.txt to file2.txt in the same directory

   cp file.txt /home/sudeep/folder/ will copy file.txt to the folder directory.

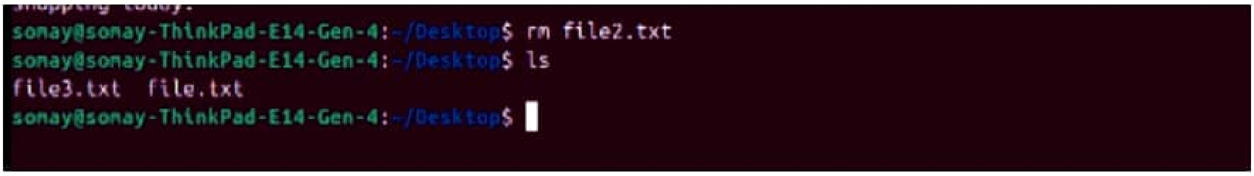


```
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ cp file2.txt file.txt
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ cat file.txt
SHOPPING IS ON YOU
you will pay for my
shopping today.
```

3. rm - This command removes a file or directory.

For example,

rm file.txt will delete file.txt.

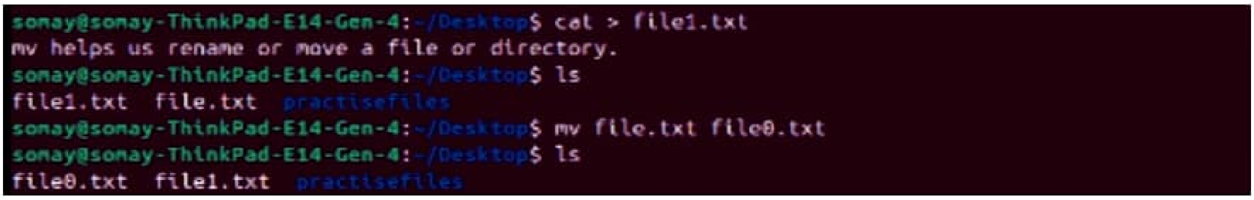


```
shopping today.
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ rm file2.txt
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ ls
file3.txt  file.txt
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ █
```

rm -r folder/ will delete folder and all its contents recursively.

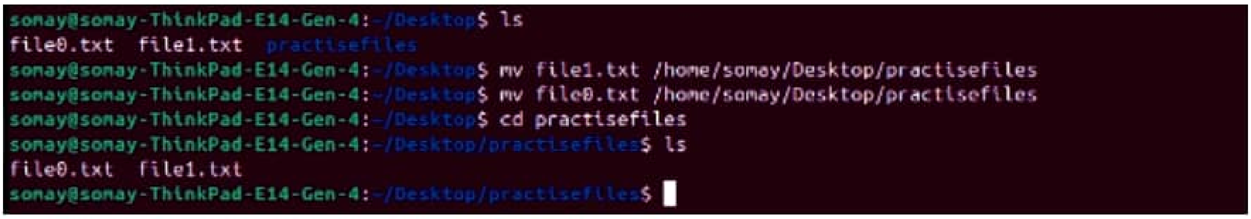
4. mv - This command moves or renames a file or directory.

For example, mv file.txt file2.txt will rename file.txt to file2.txt.



```
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ cat > file1.txt
mv helps us rename or move a file or directory.
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ ls
file1.txt  file.txt  practisefiles
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ mv file.txt file0.txt
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ ls
file0.txt  file1.txt  practisefiles
```

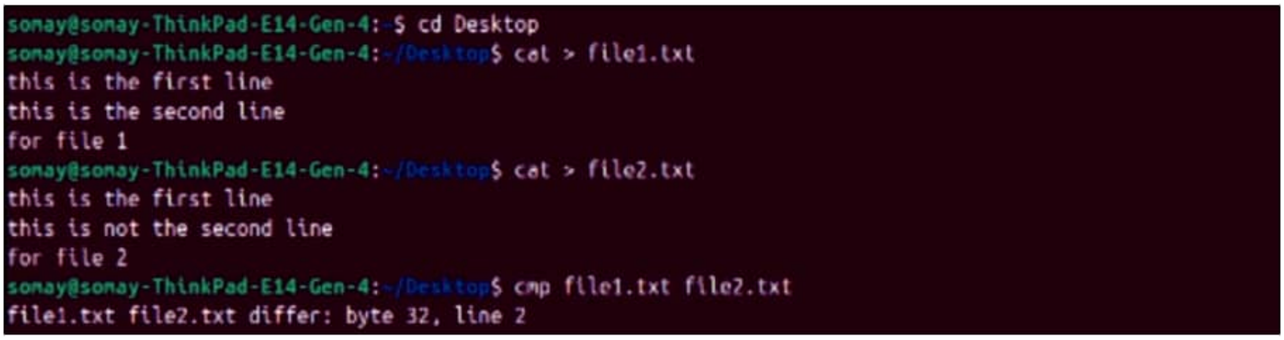
mv file2.txt /home/sudeep/folder/ will move file2.txt to the folder directory.



```
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ ls
file0.txt  file1.txt  practisefiles
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ mv file1.txt /home/sonay/Desktop/practisefiles
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ mv file0.txt /home/sonay/Desktop/practisefiles
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ cd practisefiles
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop/practisefiles$ ls
file0.txt  file1.txt
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop/practisefiles$ █
```

5. cmp - This command compares two files byte by byte and reports the first mismatch if any.

For example, cmp file1.txt file2.txt will compare file1.txt and file2.txt and show something like "file1.txt file2.txt differ: byte 10, line 2" if they are different.



```
sonay@sonay-ThinkPad-E14-Gen-4:~$ cd Desktop
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ cat > file1.txt
this is the first line
this is the second line
for file 1
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ cat > file2.txt
this is the first line
this is not the second line
for file 2
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ cmp file1.txt file2.txt
file1.txt file2.txt differ: byte 32, line 2
```

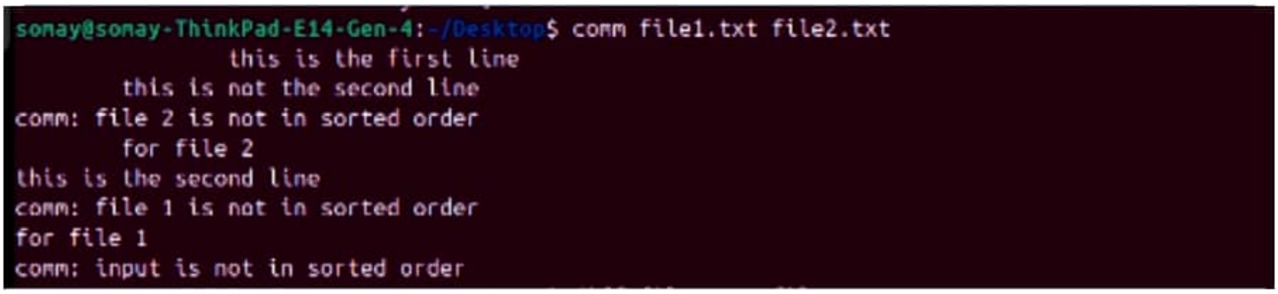
6. comm - This command compares two sorted files line by line and produces three columns of output:

lines only in the first file, lines only in the second file, and lines common to both files.

For example, comm file1.txt file2.txt will compare file1.txt and file2.txt and show something like

```
                line A
        line B
line C
```

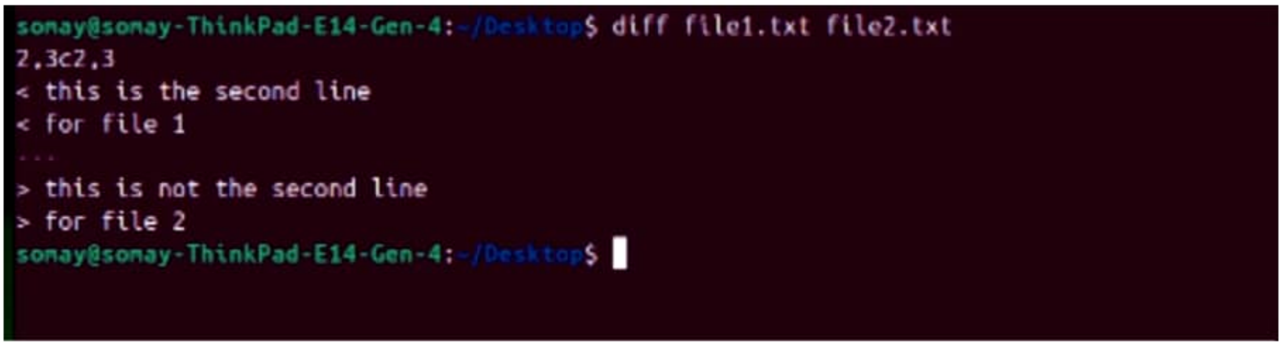
if line A is common to both files, line B is only in the first file, and line C is only in the second file.



```
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ comm file1.txt file2.txt
                this is the first line
        this is not the second line
comm: file 2 is not in sorted order
        for file 2
this is the second line
comm: file 1 is not in sorted order
for file 1
comm: input is not in sorted order
```

7. diff - This command compares two files line by line and shows the differences between them in a standard format.

For example, diff file1.txt file2.txt will compare file1.txt and file2.txt and show something like



```
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ diff file1.txt file2.txt
2,3c2,3
< this is the second line
< for file 1
---
> this is not the second line
> for file 2
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ 
```

Explanation

2,3c2,3: This indicates that lines 2 and 3 of the first file (file1.txt) are compared to lines 2 and 3 of the second file (file2.txt).

< This is second line: This line is marked with <, indicating that it's present in the first file but not in the second file. It shows the content of the line in the first file.

for file 1.: This line is also marked with < and is part of the differences in the first file.

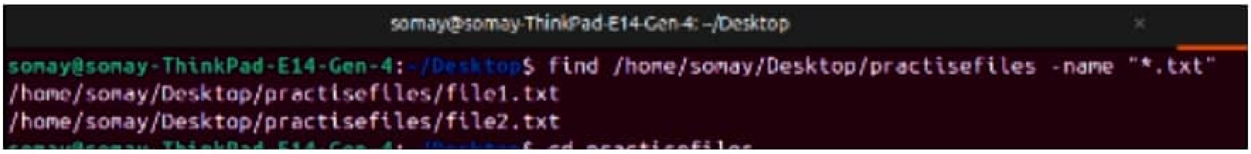
---: This is a separator line that indicates the end of the lines from the first file and the beginning of the lines from the second file.

> This is not second line: This line is marked with >, indicating that it's present in the second file but not in the first file. It shows the content of the line in the second file.

for file 2.: This line is also marked with > and is part of the differences in the second file.

8. find - This command searches for files or directories that match certain criteria.

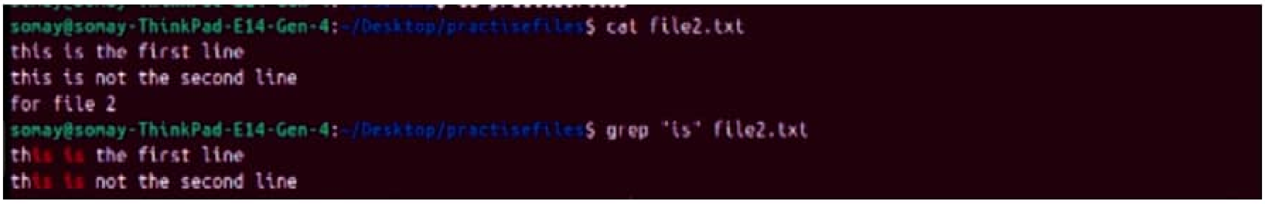
For example, find /home/sudeep/folder/ -name "*.txt" will find all the files with .txt extension under /home/sudeep/folder/ directory.



```
somay@somay-ThinkPad-E14-Gen-4: ~/Desktop
somay@somay-ThinkPad-E14-Gen-4:~/Desktop$ find /home/somay/Desktop/practisefiles -name "*.txt"
/home/somay/Desktop/practisefiles/file1.txt
/home/somay/Desktop/practisefiles/file2.txt
somay@somay-ThinkPad-E14-Gen-4:~/Desktop$ cd practisefiles
```

9. grep - This command searches for a pattern in a file or input and prints the matching lines.

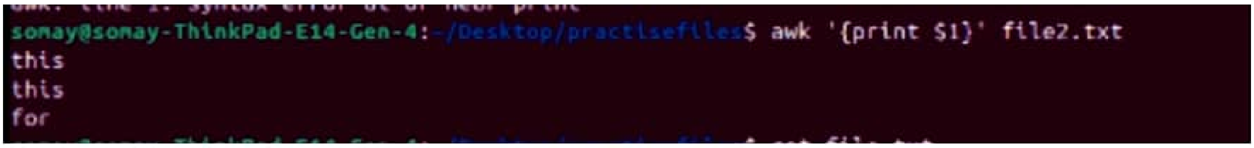
For example, grep "is" file2.txt will print all the lines in file2.txt that contain is.



```
somay@somay-ThinkPad-E14-Gen-4:~/Desktop/practisefiles$ cat file2.txt
this is the first line
this is not the second line
for file 2
somay@somay-ThinkPad-E14-Gen-4:~/Desktop/practisefiles$ grep 'is' file2.txt
this is the first line
this is not the second line
```

10. awk - This command is a powerful text processing tool that can perform various operations on files or input.

For example, awk '{print $1}' file2.txt will print the first word of each line in file2.txt.



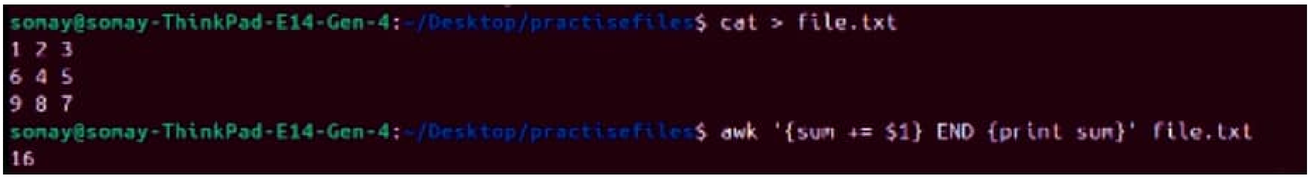
```
awk: line 1: syntax error at or near print
somay@somay-ThinkPad-E14-Gen-4:~/Desktop/practisefiles$ awk '{print $1}' file2.txt
this
this
for
```

a) **Filtering**: awk can be used to filter out lines from a file that match a specific pattern.

For instance, awk '/not/ {print $0}' file2.txt will print all the lines in file2.txt that contain the word not.

b) **Summing**: awk can be used to sum up the values in a specific column of a file.

For example, awk '{sum += $1} END {print sum}' file.txt will add up all the values in the first column of file.txt and print the total.

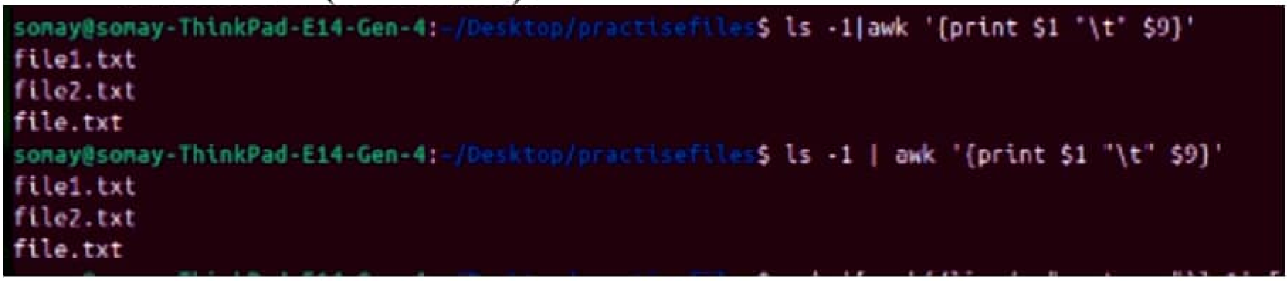


```
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop/practisefiles$ cat > file.txt
1 2 3
6 4 5
9 8 7
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop/practisefiles$ awk '{sum += $1} END {print sum}' file.txt
16
```

c) **Formatting**: awk can be used to format the output of a command.

For instance, ls -l | awk '{print $1 "\t" $9}' will print the permissions and filenames of all files in the current directory.

- awk: invokes the awk command.
- {print $1 "\t" $9}: specifies the awk program to execute. $1 refers to the first field (file permissions), "\t" inserts a tab character, and $9 refers to the ninth field (file name).

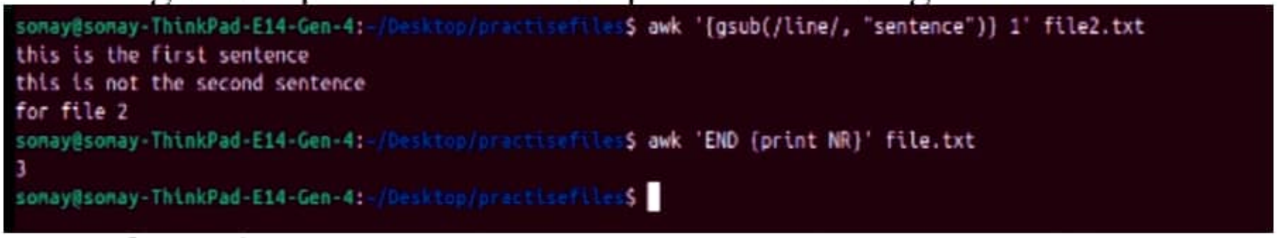


```
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop/practisefiles$ ls -1|awk '{print $1 "\t" $9}'
file1.txt
file2.txt
file.txt
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop/practisefiles$ ls -1 | awk '{print $1 "\t" $9}'
file1.txt
file2.txt
file.txt
```

d) **Replacing**: awk can be used to replace text in a file.

For example, awk '{gsub(/old/, "new")} 1' file2.txt will replace all occurrences of the word old with new in file2.txt.

gsub is an awk function that replaces all occurrences of a regular expression with a replacement string.



```
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop/practisefiles$ awk '{gsub(/line/, "sentence")} 1' file2.txt
this is the first sentence
this is not the second sentence
for file 2
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop/practisefiles$ awk 'END {print NR}' file.txt
3
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop/practisefiles$ 
```

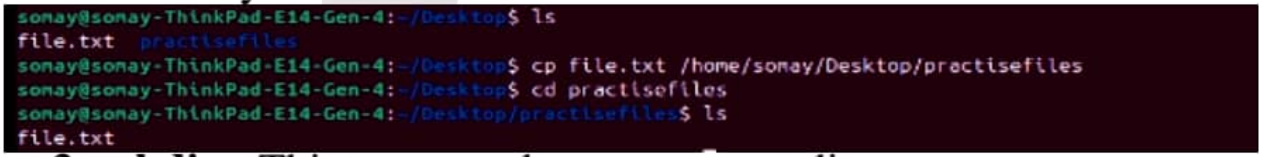
e) **Counting**: awk can be used to count the number of lines or words in a file.

For instance, awk 'END {print NR}' file.txt will print the number of lines in file.txt

Directory Management:

1. **cd -** This command changes the current working directory.

2. For example, cd /folder will change the current working directory to /folder/.



```
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ ls
file.txt  practisefiles
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ cp file.txt /home/sonay/Desktop/practisefiles
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ cd practisefiles
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop/practisefiles$ ls
file.txt
```

3. **mkdir -** This command creates a new directory.

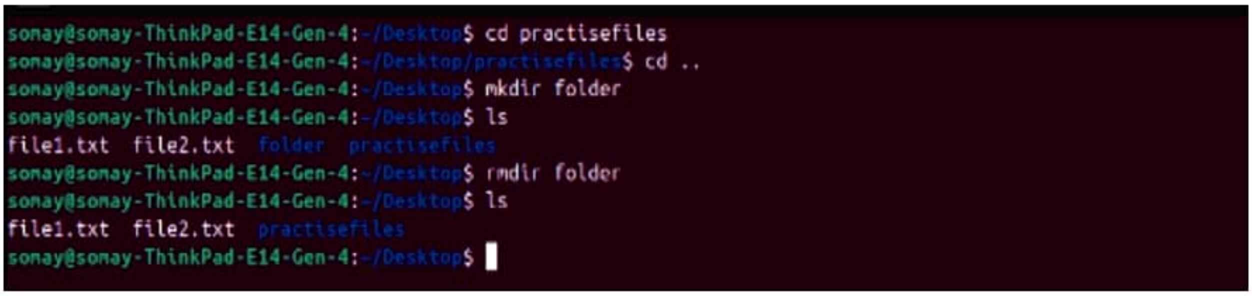
   For example, mkdir folder2 will create a new directory named folder2 in the current working directory.

4. **rmdir -** This command removes an empty directory.

   For example, rmdir folder2 will remove folder2 if it is empty.

5. **ls -** This command lists the files and directories in the current or specified directory.

   For example, ls will list the files and directories in the current working directory.
   ls -l will list them in a long format with more details.
   ls /home/user/Documents/ will list the files and directories in /home/user/Documents/.



```
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ cd practisefiles
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop/practisefiles$ cd ..
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ mkdir folder
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ ls
file1.txt  file2.txt  folder  practisefiles
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ rmdir folder
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ ls
file1.txt  file2.txt  practisefiles
sonay@sonay-ThinkPad-E14-Gen-4:~/Desktop$ 
```