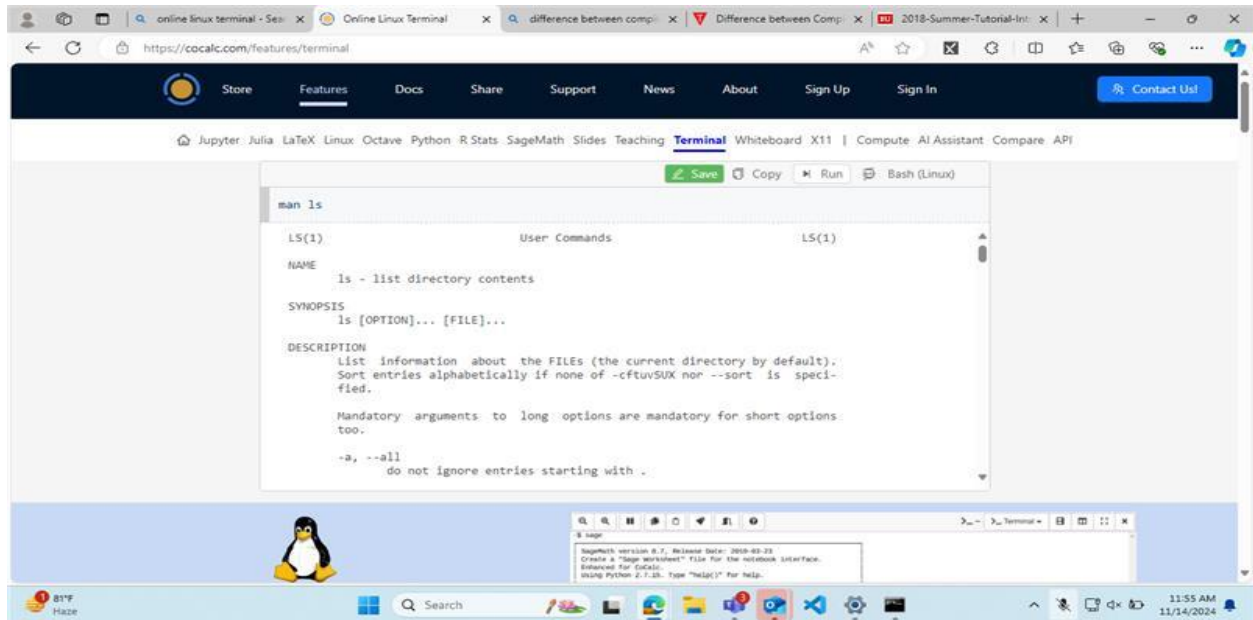


LINUX TRAINING- DAY1

MAN: It provides the manual for all commands

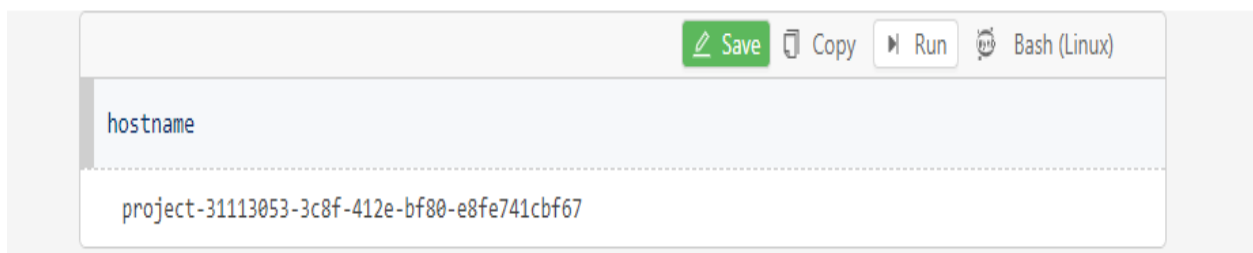


○ **whoami** # my login

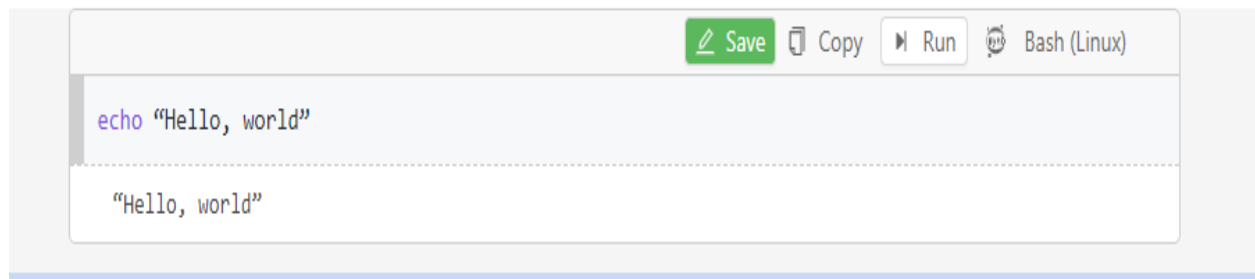
Jupyter Julia LaTeX Linux Octave Python RStats SageMath Slides Teaching **Terminal** Whiteboard X11 | Compute AI Assistant Compare API



○ **hostname** # name of this computer

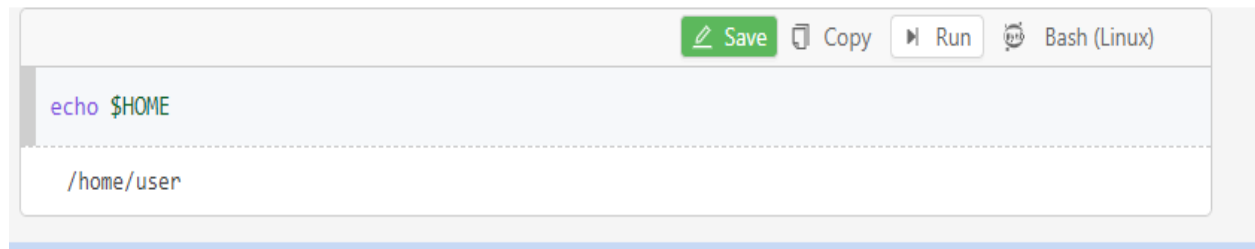


- **echo** “Hello, world” # print characters to screen



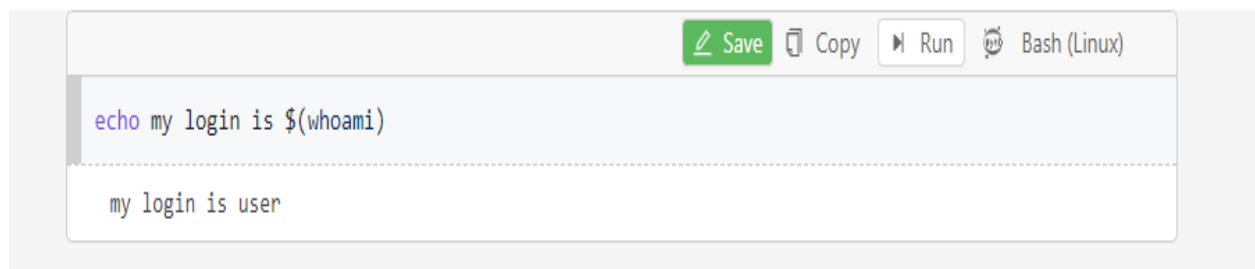
A terminal window with a light gray background. At the top right, there are four buttons: a green 'Save' button, a 'Copy' button with a clipboard icon, a 'Run' button with a play icon, and a 'Bash (Linux)' button with a terminal icon. The command `echo "Hello, world"` is entered in the input area. Below a dashed line, the output `"Hello, world"` is displayed.

- **echo** \$HOME # print environment variable



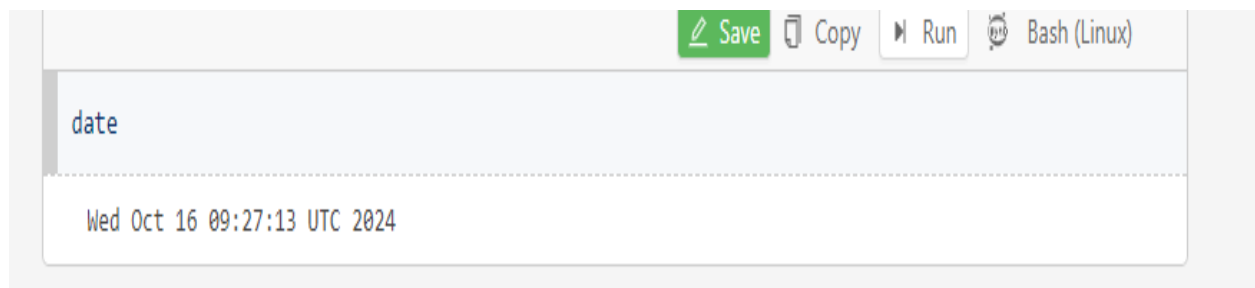
A terminal window with a light gray background. At the top right, there are four buttons: a green 'Save' button, a 'Copy' button with a clipboard icon, a 'Run' button with a play icon, and a 'Bash (Linux)' button with a terminal icon. The command `echo $HOME` is entered in the input area. Below a dashed line, the output `/home/user` is displayed.

- **echo** my login is \$(whoami) # replace \$(xx) with program output



A terminal window with a light gray background. At the top right, there are four buttons: a green 'Save' button, a 'Copy' button with a clipboard icon, a 'Run' button with a play icon, and a 'Bash (Linux)' button with a terminal icon. The command `echo my login is $(whoami)` is entered in the input area. Below a dashed line, the output `my login is user` is displayed.

- **date** # print current time/date



A terminal window with a light gray background. At the top right, there are four buttons: a green 'Save' button, a 'Copy' button with a clipboard icon, a 'Run' button with a play icon, and a 'Bash (Linux)' button with a terminal icon. The command `date` is entered in the input area. Below a dashed line, the output `Wed Oct 16 09:27:13 UTC 2024` is displayed.

- **cal** # print this month's calendar

```
cal

September 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
```

- **shazam** # bad command

```
shazam

bash: shazam: command not found
```

- **~** : home directory

```
~

bash: /home/user: Is a directory
```

- **mkdir dir_name**: create directory

```
Jupyter Julia LaTeX Linux Octave Python R Stats SageMath Slides Teaching Terminal Whiteboard X11 | Compute AI Assistant Compare API

mkdir hi
ls
hi
```

- **-j** in cal md: julian calander

🏠 Jupyter Julia LaTeX Linux Octave Python R Stats SageMath Slides Teaching **Terminal** Whiteboard X11 | Compute AI Assistant Compare API

Save Copy Run Bash (Linux)

```
cal -j 3 1999
```

March 1999						
Su	Mo	Tu	We	Th	Fr	Sa
	60	61	62	63	64	65
66	67	68	69	70	71	72
73	74	75	76	77	78	79
80	81	82	83	84	85	86
87	88	89	90			

🏠 Jupyter Julia LaTeX Linux Octave Python R Stats SageMath Slides Teaching **Terminal** Whiteboard X11 | Compute AI Assistant Compare API

Save Copy Run Bash (Linux)

```
cal -j 12 2025
```

December 2025						
Su	Mo	Tu	We	Th	Fr	Sa
	335	336	337	338	339	340
341	342	343	344	345	346	347
348	349	350	351	352	353	354
355	356	357	358	359	360	361
362	363	364	365			

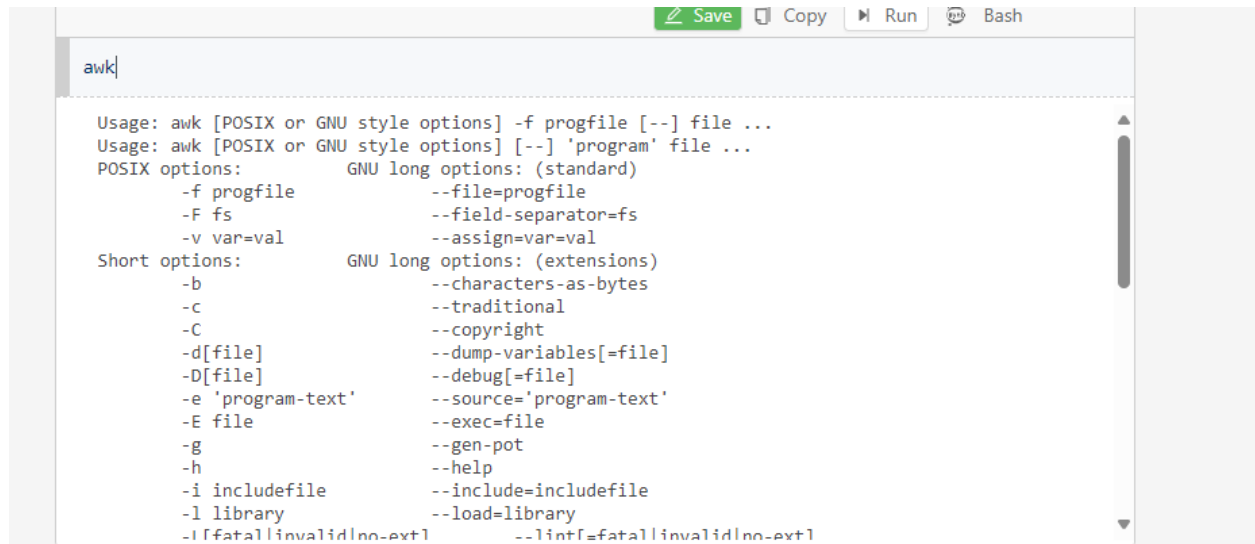
- **touch :** create empty files. Can create multiple files at a time.

Save Copy Run Bash (Linux)

```
mkdir demo
touch file1.txt file2.txt
ls
rm file1.txt
ls
```

```
demo file1.txt file2.txt
demo file2.txt
```

- **awk** Pattern scanning and processing language



The screenshot shows a terminal window with the title bar containing 'Save', 'Copy', 'Run', and 'Bash'. The prompt is 'awk|'. The output displays the usage and options for the 'awk' command.

```
Usage: awk [POSIX or GNU style options] -f progfile [--] file ...
Usage: awk [POSIX or GNU style options] [--] 'program' file ...

POSIX options:          GNU long options: (standard)
  -f progfile           --file=progfile
  -F fs                 --field-separator=fs
  -v var=val            --assign=var=val

Short options:          GNU long options: (extensions)
  -b                    --characters-as-bytes
  -c                    --traditional
  -C                    --copyright
  -d[file]              --dump-variables[=file]
  -D[file]              --debug[=file]
  -e 'program-text'     --source='program-text'
  -E file               --exec=file
  -g                    --gen-pot
  -h                    --help
  -i includefile        --include=includefile
  -l library            --load=library
  -I[fatal|invalid|no-ext]  --lint[=fatal|invalid|no-ext]
```

- **cat** Display file(s) → Cat <filename>



The screenshot shows a terminal window with the title bar containing 'Save', 'Copy', 'Run', and 'Bash (Linux)'. The prompt is 'mkdir demo'. The output shows the creation of a directory 'demo' and a file 'file1.txt'.

```
mkdir demo
touch file1.txt
echo "1 deepthi 7981428529" >> file1.txt
echo "2 keerthi 6303645400" >> file1.txt
echo "3 koushik 7382507369" >> file1.txt
echo "4 jaya 7981424567" >> file1.txt
echo "5 Harika 7986468429" >> file1.txt
cat file1.txt
```

```
1 deepthi 7981428529
2 keerthi 6303645400
3 koushik 7382507369
4 jaya 7981424567
5 Harika 7986468429
```

- **cut** Extract selected fields of each line of a file

```
Save Copy Run Bash (Linux)

touch file1.txt file2.txt
echo "Apples is red" >>file2.txt
echo "Mango is yellow" >> file2.txt
echo "Note:-- >cutting the first and sixth character from each line, and the second command is cutting the first to the third character from each line."
cut -c 1,6 file2.txt
cut -c 1-3 file2.txt
```

Note:-- >cutting the first and sixth character from each line, and the second command is cutting the first to the third character from each line.

As
M
App
Man

- **diff** Compare two files

```
Save Copy Run Bash (Linux)

mkdir demo
touch file1.txt file2.txt
echo "1 deepthi 7981428529" >> file1.txt
echo "2 keerthi 6303645400" >> file1.txt
echo "3 koushik 7382507369" >> file1.txt
echo "4 jaya 7981424567" >> file1.txt
echo "5 Harika 7986468429" >> file1.txt
echo "3 koushik 7382507369" >> file2.txt
echo "4 jaya 7981424567" >> file2.txt
echo "5 Harika 7986468429" >> file2.txt
diff file1.txt file2.txt
```

1,2d0
< 1 deepthi 7981428529
< 2 keerthi 6303645400

```
mkdir demo
touch file1.txt file2.txt
echo "1 deepthi 7981428529" >> file1.txt
echo "2 keerthi 6303645400" >> file1.txt
echo "3 koushik 7382507369" >> file1.txt
echo "4 jaya 7981424567" >> file1.txt
echo "5 Harika 7986468429" >> file1.txt
echo "3 koushik 7382507369" >> file2.txt
echo "4 jaya 7981424567" >> file2.txt
echo "5 Harika 7986468429" >> file2.txt
echo "6 lulumall 123456789" >> file2.txt
diff file1.txt file2.txt
```

```
1,2d0
< 1 deepthi 7981428529
< 2 keerthi 6303645400
5a4
> 6 lulumall 123456789
```

- **grep** Search text for a pattern

```
touch file1.txt file2.txt
echo "1 deepthi 7981428529" >> file1.txt
echo "2 keerthi 6303645400" >> file1.txt
echo "3 koushik 7382507369" >> file1.txt
echo "4 jaya 7981424567" >> file1.txt
echo "5 Harika 7986468429" >> file1.txt
echo "6 deepthi 7981428529" >> file1.txt
echo "7 keerthi 6303645400" >> file1.txt
echo "8 koushik 7382507369" >> file1.txt
echo "9 jaya 7981424567" >> file1.txt
echo "10 Harika 7986468429" >> file1.txt
cat file1.txt | grep 52
echo "-----"
grep jay file1.txt
```

```
1 deepthi 7981428529
6 deepthi 7981428529
-----
4 jaya 7981424567
9 jaya 7981424567
```

- **head** Display the first part of files → first 10 lines

```
touch file1.txt
echo "1 deepthi 7981428529" >> file1.txt
echo "2 keerthi 6303645400" >> file1.txt
echo "3 koushik 7382507369" >> file1.txt
echo "4 jaya 7981424567" >> file1.txt
echo "5 Harika 7986468429" >> file1.txt
echo "6 deepthi 7981428529" >> file1.txt
echo "7 keerthi 6303645400" >> file1.txt
echo "8 koushik 7382507369" >> file1.txt
echo "9 jaya 7981424567" >> file1.txt
echo "10 Harika 7986468429" >> file1.txt
echo "11 deepthi 7981428529" >> file1.txt
echo "12 keerthi 6303645400" >> file1.txt
echo "13 koushik 7382507369" >> file1.txt
head file1.txt
```

```
1 deepthi 7981428529
2 keerthi 6303645400
3 koushik 7382507369
4 jaya 7981424567
5 Harika 7986468429
6 deepthi 7981428529
7 keerthi 6303645400
8 koushik 7382507369
9 jaya 7981424567
10 Harika 7986468429
```

- **less** Display files on a page-by-page basis (not executed)
- **sed** Stream editor (esp. search and replace)

Note: Linux 'sed' command stands for stream editor. It is used to edit streams (files) **using regular expressions**. But this editing is not permanent. It remains only in display, but in actual, file content remains the same.

Primarily, it is used for text substitution; additionally, it can be used for other text manipulation operations like insert, delete, search, and more. The sed command allows us to edit files without opening them. Regular expression support makes it a more powerful text manipulation tool.

```
mkdir demo
touch file1.txt file2.txt
ls
echo "1 deepthi 7981428529" >> file1.txt
echo "2 keerthi 6303645400" >> file1.txt
echo "3 koushik 7382507369" >> file1.txt
echo "4 jaya 7981424567" >> file1.txt
echo "5 Harika 7986468429" >> file1.txt
sed 's/jaya/jay/g' file1.txt
cat file1.txt
```

```
demo file1.txt file2.txt
1 deepthi 7981428529
2 keerthi 6303645400
3 koushik 7382507369
4 jay 7981424567
5 Harika 7986468429
1 deepthi 7981428529
2 keerthi 6303645400
3 koushik 7382507369
4 jaya 7981424567
5 Harika 7986468429
```

sed -i 's/jaya/jay/g' file1.txt

cat file1.txt

note: -i → changes the original content in file also and to display it use cat
cuz sed -I don't return anything just changes the content.

's/' → to mention re

Change the content jaya → jay

/g → all matches will change

```
mkdir demo
touch file1.txt file2.txt
ls
echo "1 deepthi 7981428529" >> file1.txt
echo "2 keerthi 6303645400" >> file1.txt
echo "3 koushik 7382507369" >> file1.txt
echo "4 jaya 7981424567" >> file1.txt
echo "5 Harika 7986468429" >> file1.txt

sed -i 's/jaya/jay/g' file1.txt
cat file1.txt
```

```
demo file1.txt file2.txt
1 deepthi 7981428529
2 keerthi 6303645400
3 koushik 7382507369
4 jay 7981424567
5 Harika 7986468429
```

● sort Sort text files

```
touch file1.txt
echo "1" >> file1.txt
echo "3" >> file1.txt
echo "2" >> file1.txt
echo "9" >> file1.txt
echo "5" >> file1.txt
echo "7" >> file1.txt
echo "6" >> file1.txt
cat file1.txt
echo "-----"
sort file1.txt
```

```
1
3
2
9
5
7
6
-----
1
2
3
5
6
7
9
```

- **split** Split files

Split: to split content into multiple files

`split -l 4 file1.txt <file2.txt opt>`

`-l` → split by lines

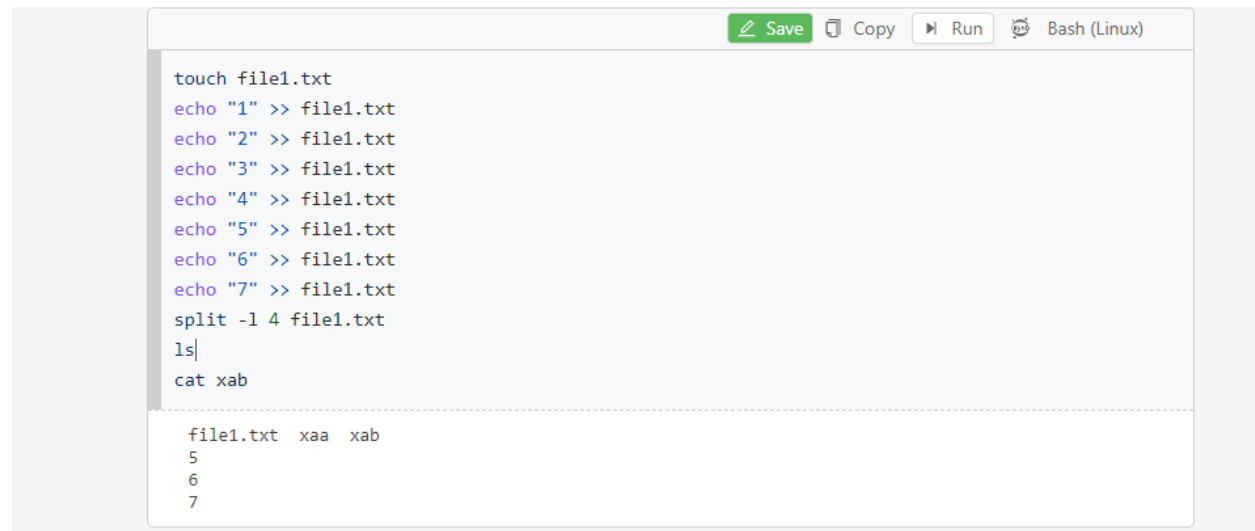
`4` → no of lines in one file

`File1.txt` → name of file

`File2.txt` is the name on which split file can be accessed if not specified
‘x’ is considered

After split file names will be `xaa,xab,xac....`

[Jupyter](#) [Julia](#) [LaTeX](#) [Linux](#) [Octave](#) [Python](#) [R Stats](#) [SageMath](#) [Slides](#) [Teaching](#) [Terminal](#) [Whiteboard](#) [X11](#) | [Compute](#) [AI Assistant](#) [Compar](#)



The screenshot shows a Jupyter Terminal window with a light blue background. At the top, there is a toolbar with buttons for 'Save' (green), 'Copy' (blue), 'Run' (blue), and 'Bash (Linux)' (blue). Below the toolbar, the terminal displays the following commands and their output:

```
touch file1.txt
echo "1" >> file1.txt
echo "2" >> file1.txt
echo "3" >> file1.txt
echo "4" >> file1.txt
echo "5" >> file1.txt
echo "6" >> file1.txt
echo "7" >> file1.txt
split -l 4 file1.txt
ls|
cat xab
```

The output of the `ls|` command is shown in a separate box below the terminal, displaying the files created by the `split` command:

```
file1.txt  xaa  xab
5
6
7
```

- **tail** Display the last part of a file → max last 10 lines

```
touch file1.txt
echo "1 deepthi 7981428529" >> file1.txt
echo "2 keerthi 6303645400" >> file1.txt
echo "3 koushik 7382507369" >> file1.txt
echo "4 jaya 7981424567" >> file1.txt
echo "5 Harika 7986468429" >> file1.txt
echo "6 deepthi 7981428529" >> file1.txt
echo "7 keerthi 6303645400" >> file1.txt
echo "8 koushik 7382507369" >> file1.txt
echo "9 jaya 7981424567" >> file1.txt
echo "10 Harika 7986468429" >> file1.txt
echo "11 deepthi 7981428529" >> file1.txt
echo "12 keerthi 6303645400" >> file1.txt
echo "13 koushik 7382507369" >> file1.txt
tail file1.txt
```

```
4 jaya 7981424567
5 Harika 7986468429
6 deepthi 7981428529
7 keerthi 6303645400
8 koushik 7382507369
9 jaya 7981424567
10 Harika 7986468429
11 deepthi 7981428529
12 keerthi 6303645400
13 koushik 7382507369
```

```
touch file1.txt file2.txt
echo "1 deepthi 7981428529" >> file1.txt
echo "2 keerthi 6303645400" >> file1.txt
echo "3 koushik 7382507369" >> file1.txt
echo "4 jaya 7981424567" >> file1.txt
echo "5 Harika 7986468429" >> file1.txt
echo "6 deepthi 7981428529" >> file1.txt
echo "7 keerthi 6303645400" >> file1.txt
echo "8 koushik 7382507369" >> file1.txt
echo "9 jaya 7981424567" >> file1.txt
echo "10 Harika 7986468429" >> file1.txt
tail -n 3 file1.txt
```

```
8 koushik 7382507369
9 jaya 7981424567
10 Harika 7986468429
```

- **tr** Translate/delete characters → replace /change the chr's

```
touch file1.txt
echo "welcome to tech mahindra" >> file1.txt
cat file1.txt | tr [a-z] [A-Z]
```

```
WELCOME TO TECH MAHINDRA
```

- **uniq** Filter out repeated lines in a file

```
Save Copy Run Bash (Linux)

touch file1.txt
echo "hi">>file1.txt
echo "hi">>file1.txt
echo "hi">>file1.txt
echo "hello">>file1.txt
echo "hi dear">>file1.txt
echo "hi priya">>file1.txt
echo "hi mustafa">>file1.txt
uniq file1.txt

hi
hello
hi dear
hi priya
hi mustafa
```

- **wc** Line, word and character count

```
Save Copy Run Bash (Linux)

touch file1.txt
echo "hi">>file1.txt
echo "hi">>file1.txt
echo "hi">>file1.txt
echo "hello">>file1.txt
echo "hi dear">>file1.txt
echo "hi priya">>file1.txt
echo "hi mustafa">>file1.txt
wc file1.txt

7 10 43 file1.txt
```

Note: 7 lines, 10 words, 43 chr's

Create two files – add content to file 1, copy file1 content to file 2 using cp, moving the file2 content to file3 using mv

(while using cp both files exist after copying the content but in mv command file content will move to new file and previous file will not exist .

We can also use mv command to **rename** the file.)

cp <src file > <destination file>

mv <src file> <destination file>

Jupyter Julia LaTeX Linux Octave Python R Stats SageMath Slides Teaching Terminal Whiteboard X11 | Compute AI Assistant Comp

Save Copy Run Bash (Linux)

```
mkdir demo
touch file1.txt file2.txt file3.txt
ls
echo "1 deepthi 7981428529" >> file1.txt
echo "2 keerthi 6303645400" >> file1.txt
echo "3 koushik 7382507369" >> file1.txt
echo "4 jaya 7981424567" >> file1.txt
echo "5 Harika 7986468429" >> file1.txt

cp file1.txt file2.txt
mv file2.txt file3.txt
cat file3.txt
cat file2.txt

demo file1.txt file2.txt file3.txt
1 deepthi 7981428529
2 keerthi 6303645400
3 koushik 7382507369
4 jaya 7981424567
5 Harika 7986468429
cat: file2.txt: No such file or directory
```

rm:

Save Copy Run Bash

```
touch file1.txt file2.txt
echo "Apples is red" >>file2.txt
echo "Mango is yellow" >> file2.txt
ls
rm file2.txt
echo "----"
ls

file1.txt file2.txt
----
file1.txt
```