30-9-2021

grep Command

* • The grep filter searches a file for a particular pattern of characters, and displays all line that contain that pattern

option Description.

(1) -c : This used to print the no. of line which contain particular pattern.

(2) -i : Its ignore, case of matching

grep -i 'LINUX' filename
give output highlight all the word line where Linux LINUX linux becoz -i not case sensitive.

(3) -n : It display matched lines and there number

grep -n 'text' filename

(4) -V : It display all the line that do not matche the pattern.

grep -v 'text' filename

(5) -w : When you want to mathched whole word

grep -w 'pattern' filename

Page No.:
Date: / /

⑥ grep -l 'pattern' file

[It display · only file name]

Ex ⌈grep -i -w 'data' file1⌉

give all the line o/p which have full whole word 'data'.

*⑦ grep '^a' file,

give line of file which have 'a' as a staring Symbol.

*⑧ grep -An : print the n line after patter matched

ex grep -A 4 "word" file1

⑨ grep Bn: print the 'n' line before pattern matched

grep B 5 'word' file2

⑩ grep Cn: print 'n' line before and after patter matched.

grep C4 'word' file2

⑪ grep 'data$' filename

'data' pattern end of the line

To search two differen word in file

* grep -w 'word1|word2' /path/to/file

Head & tail command

head use for output of first part of the file while tail commend will print the last part of the file.

(1) head -n7 file

print first 7 line of file

(3) head -c 3 file

It print all the data which is 3 digit के है

(4) head -n3 file1 file2

give first free line within both file

(5) head -v file

o/p ==> text <= =

2021/10/01 08:10

~~tail~~ $\rightarrow$ $\boxed{\text{-tail -n 3 file}}$

Tail command

(6)
```
tail -n 3 file
```
give last 3 line of file data

(7)
```
tail -n3 file1 file2
```
give output of 3 last line from both file

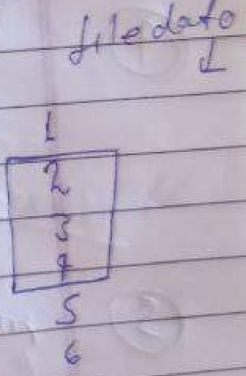
$\boxed{\text{tail -f timestamp.log}}$

Head tail together

[particular portion print करवाने]

(8)
```
tail -n+2 file | head -n3
```

यह This give the output of start with 2 line and upto 4 line



o/p → 2 ~~
3 ~~
4 ~~

More command

(1) more file
[whole text appeard at once

(2) more -d file
It display 25% of data of file
press 'space' to continue
'q' for quit.

(3) more -f filename
more -c filename

(4) more -p filename
(Clear the screen then display the text)

(5) more +30 file
Display the text after 30 lines

for read cat file | more.

2021/10/01 08:10

Chmod

It change the file/directory permission by command

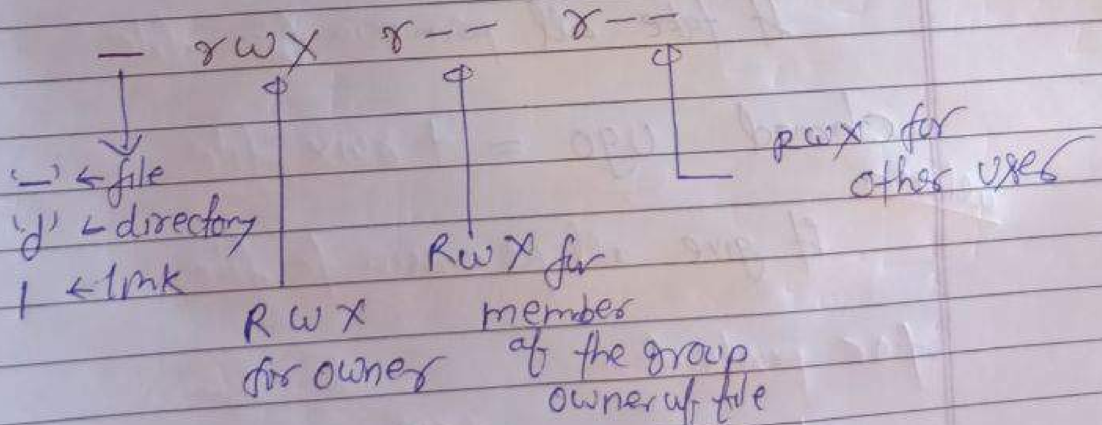
u ← user
g ← group user
o ← other ~~ouser~~ user
a ← all.

0 — No
1 — X (execute)
2 — W (write)
4 — R (read)

$\boxed{777}$ — all permission.

+ ← add permission
- ← revoke
= ← specif permission

a ←



* give all permission to all files / directory

① ~~*~~ · chmod 777 *

② given permission to particular file

chmod 777 filename

or

* chmod a = rwx file
[all permission]

③ revoke or add

chmod ugo = -rwx file

It take all permission from file

chmod $\underbrace{\text{ugo}}_{\text{all user}}$ = + rwx file.

It give all permission to file.

for particular

chmod $\boxed{u}$ = $\overset{+rw}{-rw}$ filename
$\boxed{g}$
$\boxed{o}$

It help to revoke add permission for particular user.

2021/10/01 08:10

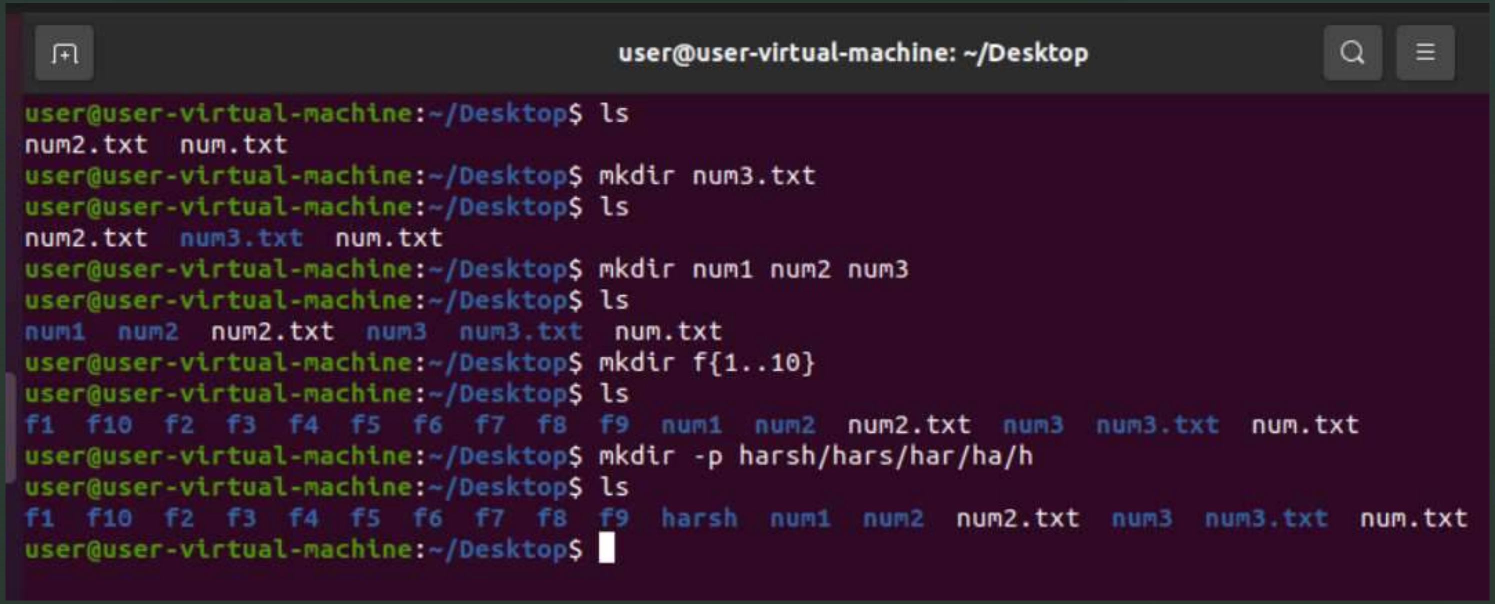
COMMANDS

NITIN KUMAR GAUR
EMP ID 6168

1. **mkdir:**

   Command used for creating directory.

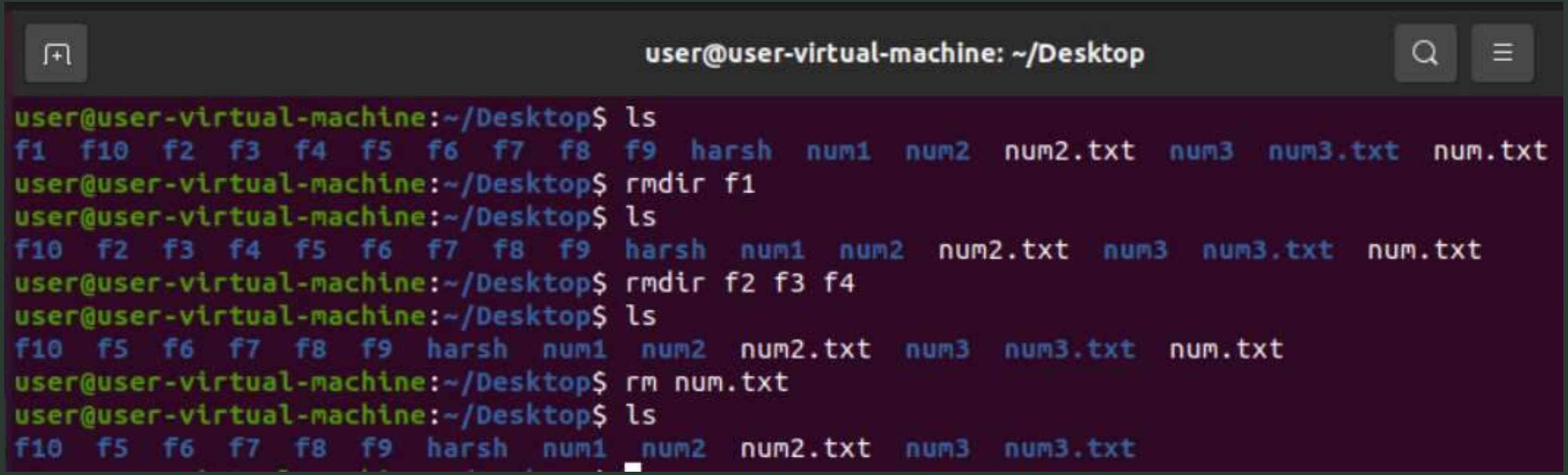
   - ❑ *For creating multiple folders→ mkdir foldername1 foldername2 foldername3*
   - ❑ *For creating 10 or more folders→ mkdir foldername{1..10}*
   - ❑ *For creating folder and subfolders→ mkdir –p ‘f1/f2/f3’*



```
user@user-virtual-machine:~/Desktop$ ls
num2.txt  num.txt
user@user-virtual-machine:~/Desktop$ mkdir num3.txt
user@user-virtual-machine:~/Desktop$ ls
num2.txt  num3.txt  num.txt
user@user-virtual-machine:~/Desktop$ mkdir num1 num2 num3
user@user-virtual-machine:~/Desktop$ ls
num1  num2  num2.txt  num3  num3.txt  num.txt
user@user-virtual-machine:~/Desktop$ mkdir f{1..10}
user@user-virtual-machine:~/Desktop$ ls
f1  f10  f2  f3  f4  f5  f6  f7  f8  f9  num1  num2  num2.txt  num3  num3.txt  num.txt
user@user-virtual-machine:~/Desktop$ mkdir -p harsh/hars/har/ha/h
user@user-virtual-machine:~/Desktop$ ls
f1  f10  f2  f3  f4  f5  f6  f7  f8  f9  harsh  num1  num2  num2.txt  num3  num3.txt  num.txt
user@user-virtual-machine:~/Desktop$ 
```

2. **<u>rmdir and rm</u>**

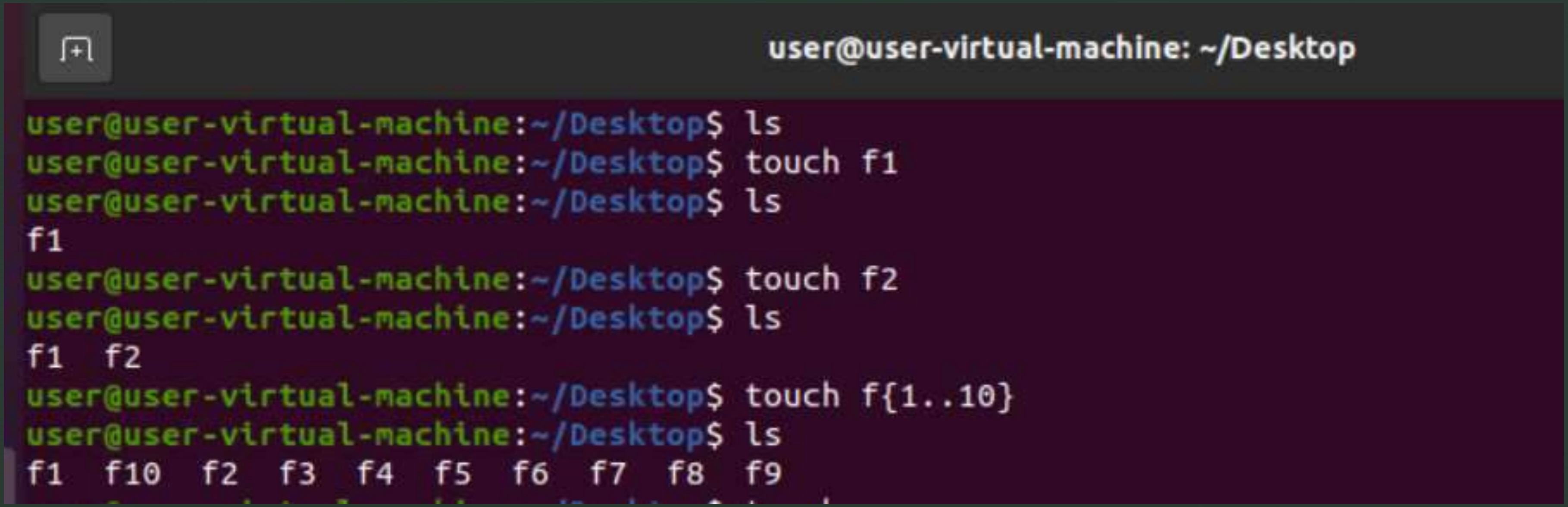
- ❑ *rmdir→Used for removing directory only,can delete only empty files.*
- ❑ *rmdir –rf→Used for deleting non-empty files forcefully.*
- ❑ *rm→Used for removing files.*
- ❑ *rm –i→Will delete file but,ask if Y(Yes) or N(No) for file deletion.*
- ❑ *rm*→deletes all file*



```
user@user-virtual-machine:~/Desktop$ ls
f1  f10  f2  f3  f4  f5  f6  f7  f8  f9  harsh  num1  num2  num2.txt  num3  num3.txt  num.txt
user@user-virtual-machine:~/Desktop$ rmdir f1
user@user-virtual-machine:~/Desktop$ ls
f10  f2  f3  f4  f5  f6  f7  f8  f9  harsh  num1  num2  num2.txt  num3  num3.txt  num.txt
user@user-virtual-machine:~/Desktop$ rmdir f2 f3 f4
user@user-virtual-machine:~/Desktop$ ls
f10  f5  f6  f7  f8  f9  harsh  num1  num2  num2.txt  num3  num3.txt  num.txt
user@user-virtual-machine:~/Desktop$ rm num.txt
user@user-virtual-machine:~/Desktop$ ls
f10  f5  f6  f7  f8  f9  harsh  num1  num2  num2.txt  num3  num3.txt
```

3. **touch**

- *Creates empty file can create one or more than one file at a time.*

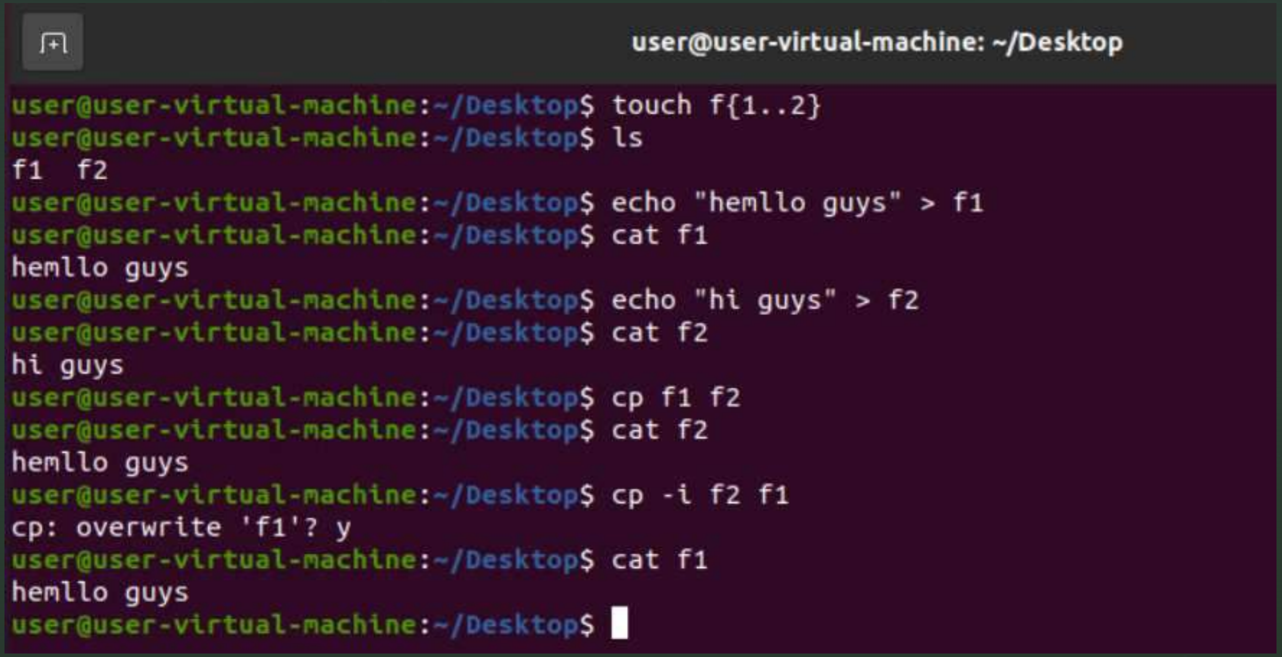


user@user-virtual-machine: ~/Desktop

```
user@user-virtual-machine:~/Desktop$ ls
user@user-virtual-machine:~/Desktop$ touch f1
user@user-virtual-machine:~/Desktop$ ls
f1
user@user-virtual-machine:~/Desktop$ touch f2
user@user-virtual-machine:~/Desktop$ ls
f1  f2
user@user-virtual-machine:~/Desktop$ touch f{1..10}
user@user-virtual-machine:~/Desktop$ ls
f1  f10  f2  f3  f4  f5  f6  f7  f8  f9
```

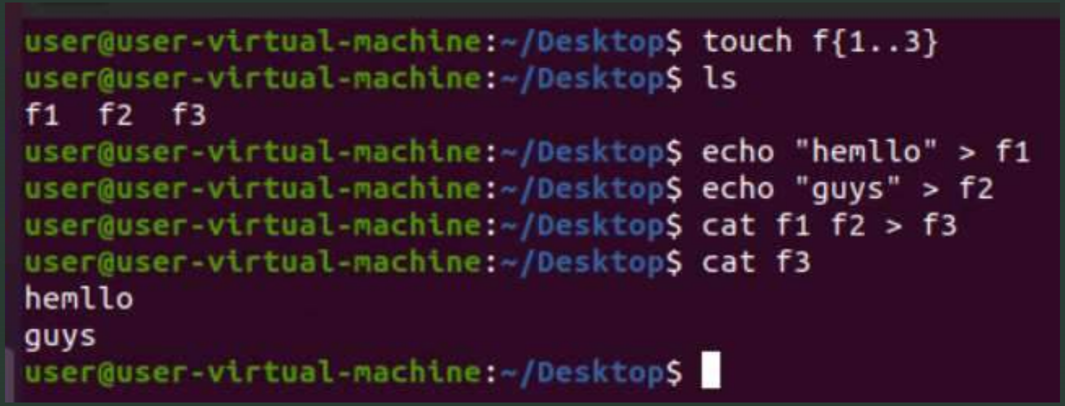
4. **cp**

- *Command used for copying data from one location(source) to other(destination).*
- *Overwrites the data in destination folder if data present before.*
- *Syntax: cp file1 file2*
- *cp –i file1 file2→asks before overwriting Y or N*



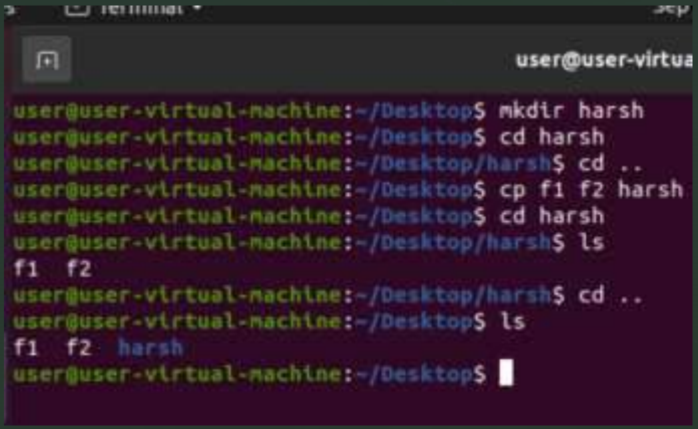
```
user@user-virtual-machine: ~/Desktop
user@user-virtual-machine:~/Desktop$ touch f{1..2}
user@user-virtual-machine:~/Desktop$ ls
f1  f2
user@user-virtual-machine:~/Desktop$ echo "hemllo guys" > f1
user@user-virtual-machine:~/Desktop$ cat f1
hemllo guys
user@user-virtual-machine:~/Desktop$ echo "hi guys" > f2
user@user-virtual-machine:~/Desktop$ cat f2
hi guys
user@user-virtual-machine:~/Desktop$ cp f1 f2
user@user-virtual-machine:~/Desktop$ cat f2
hemllo guys
user@user-virtual-machine:~/Desktop$ cp -i f2 f1
cp: overwrite 'f1'? y
user@user-virtual-machine:~/Desktop$ cat f1
hemllo guys
user@user-virtual-machine:~/Desktop$ 
```

- *cp file1 file2 > file3 → Used for copying content from different files to desired file location.*



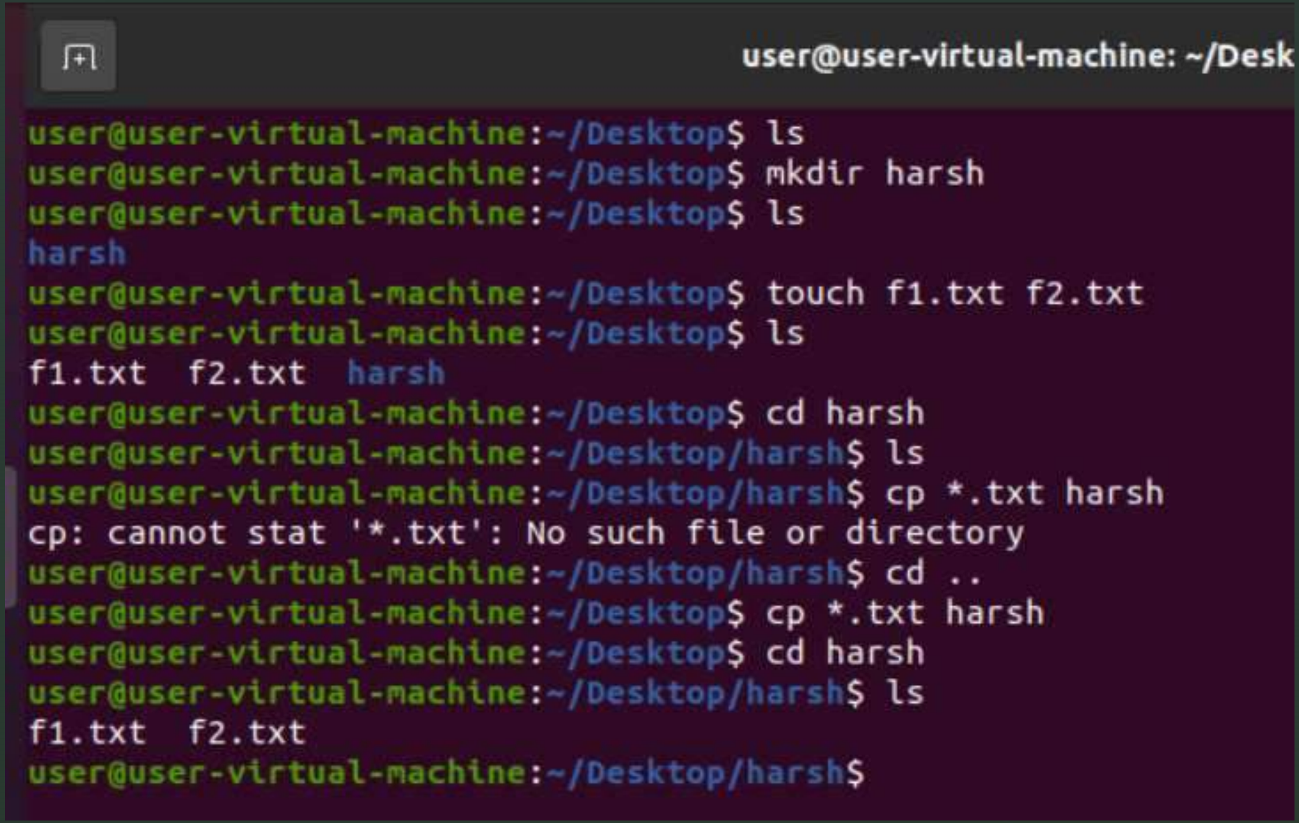
```
user@user-virtual-machine:~/Desktop$ touch f{1..3}
user@user-virtual-machine:~/Desktop$ ls
f1  f2  f3
user@user-virtual-machine:~/Desktop$ echo "hemllo" > f1
user@user-virtual-machine:~/Desktop$ echo "guys" > f2
user@user-virtual-machine:~/Desktop$ cat f1 f2 > f3
user@user-virtual-machine:~/Desktop$ cat f3
hemllo
guys
user@user-virtual-machine:~/Desktop$ 
```

- *cp file1 file2 directory →Copies file1 file2 data into desired directories.*



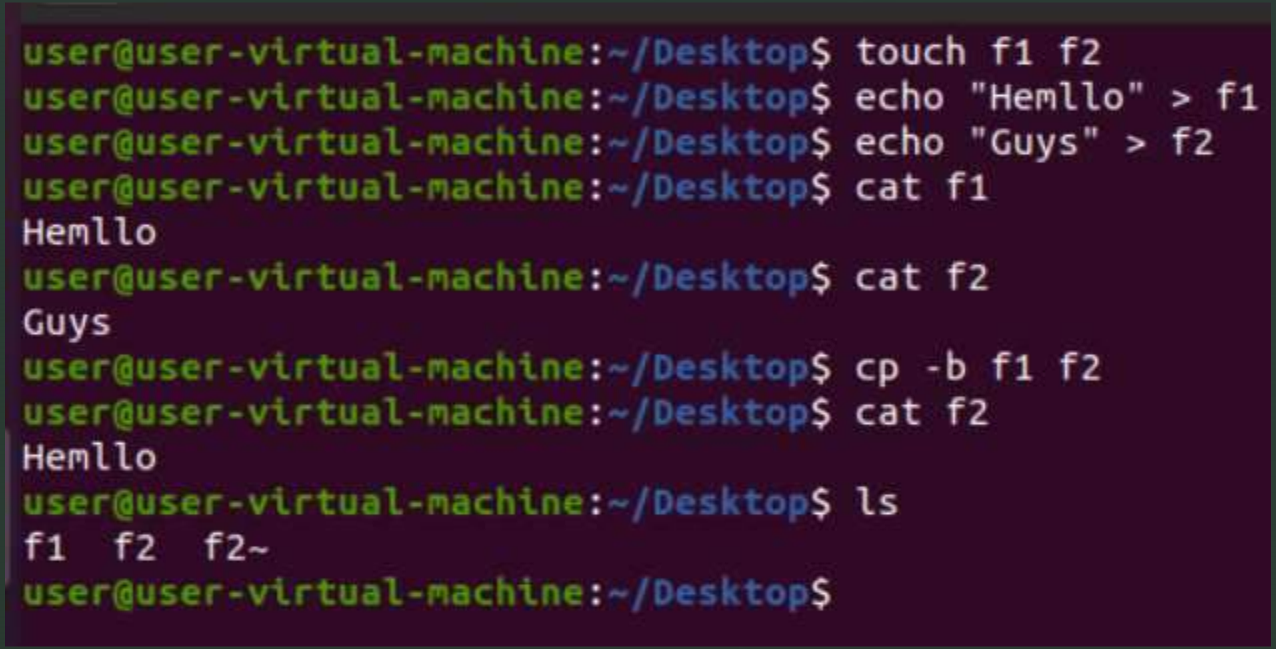
```
user@user-virtual-machine:~/Desktop$ mkdir harsh
user@user-virtual-machine:~/Desktop$ cd harsh
user@user-virtual-machine:~/Desktop/harsh$ cd ..
user@user-virtual-machine:~/Desktop$ cp f1 f2 harsh
user@user-virtual-machine:~/Desktop$ cd harsh
user@user-virtual-machine:~/Desktop/harsh$ ls
f1  f2
user@user-virtual-machine:~/Desktop/harsh$ cd ..
user@user-virtual-machine:~/Desktop$ ls
f1  f2  harsh
user@user-virtual-machine:~/Desktop$ 
```

- cp * .extension →It will remove all file having that particular extension from the location. Eg: cp * .txt It will remove all text file.



```
user@user-virtual-machine: ~/Desk
user@user-virtual-machine:~/Desktop$ ls
user@user-virtual-machine:~/Desktop$ mkdir harsh
user@user-virtual-machine:~/Desktop$ ls
harsh
user@user-virtual-machine:~/Desktop$ touch f1.txt f2.txt
user@user-virtual-machine:~/Desktop$ ls
f1.txt  f2.txt  harsh
user@user-virtual-machine:~/Desktop$ cd harsh
user@user-virtual-machine:~/Desktop/harsh$ ls
user@user-virtual-machine:~/Desktop/harsh$ cp *.txt harsh
cp: cannot stat '*.txt': No such file or directory
user@user-virtual-machine:~/Desktop/harsh$ cd ..
user@user-virtual-machine:~/Desktop$ cp *.txt harsh
user@user-virtual-machine:~/Desktop$ cd harsh
user@user-virtual-machine:~/Desktop/harsh$ ls
f1.txt  f2.txt
user@user-virtual-machine:~/Desktop/harsh$
```

- *cp –b file1 file2→Copies data from file1 to file 2 but just a small difference between normal copy and this,that it makes a backup of the file which is about to be overwritten or the destination file,It is denoted by ~filename.*

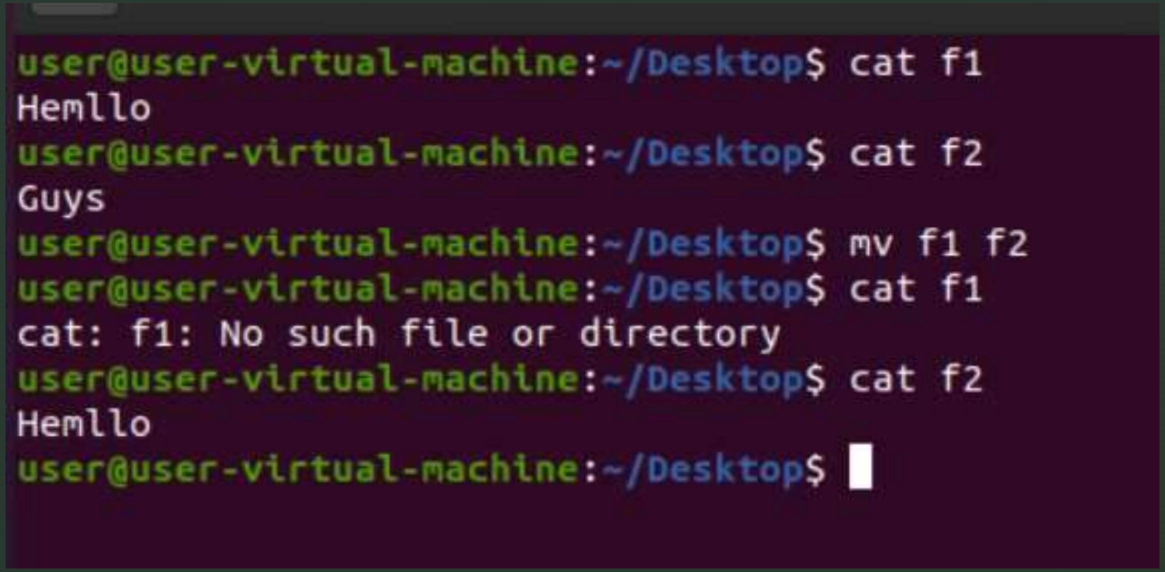


```
user@user-virtual-machine:~/Desktop$ touch f1 f2
user@user-virtual-machine:~/Desktop$ echo "Hemllo" > f1
user@user-virtual-machine:~/Desktop$ echo "Guys" > f2
user@user-virtual-machine:~/Desktop$ cat f1
Hemllo
user@user-virtual-machine:~/Desktop$ cat f2
Guys
user@user-virtual-machine:~/Desktop$ cp -b f1 f2
user@user-virtual-machine:~/Desktop$ cat f2
Hemllo
user@user-virtual-machine:~/Desktop$ ls
f1  f2  f2~
user@user-virtual-machine:~/Desktop$
```

5. **mv**

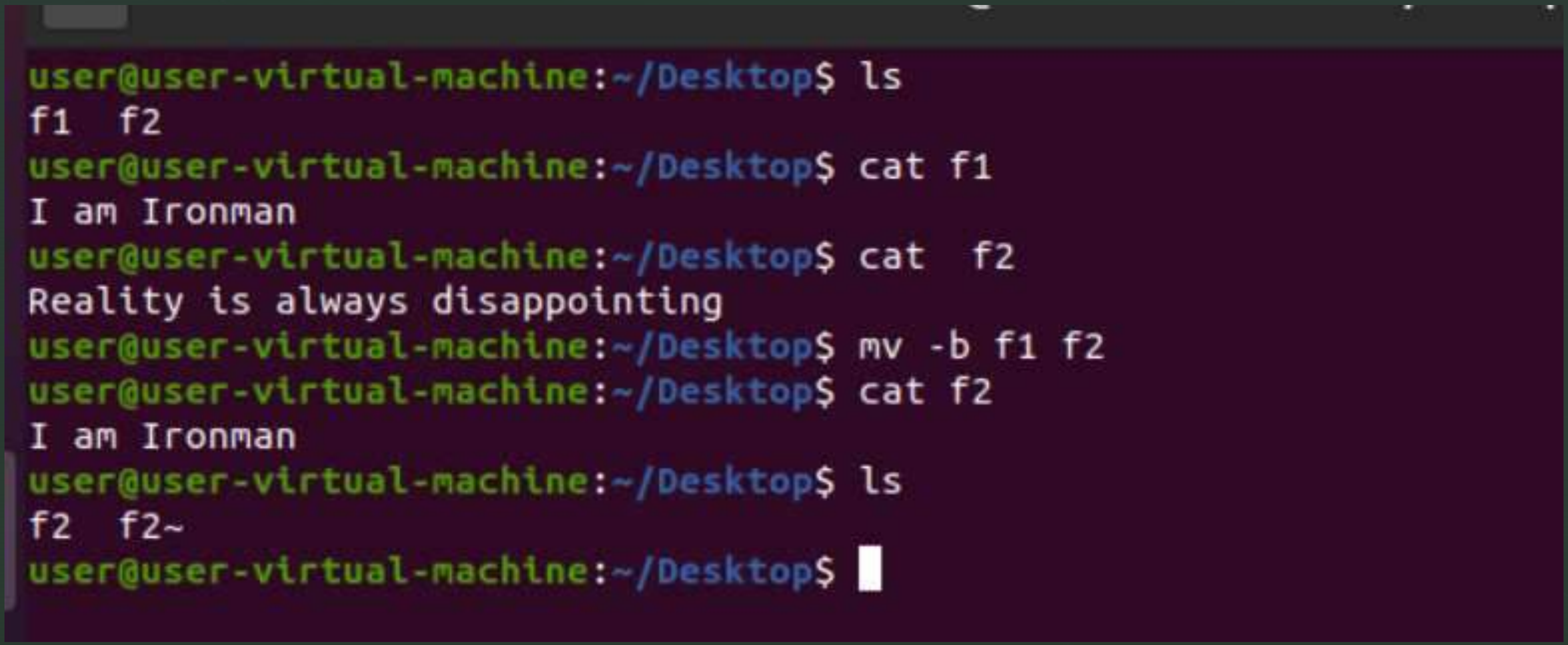
To relocate an existing file or directory from one location to another, use the mv command in Linux. It can also be used to change the name of a file or directory. The'mv' option is ideal to use if you only want to rename a single directory or file.

It moves data from one file to other thus deleting the source file and it can be illustrated by the below example.



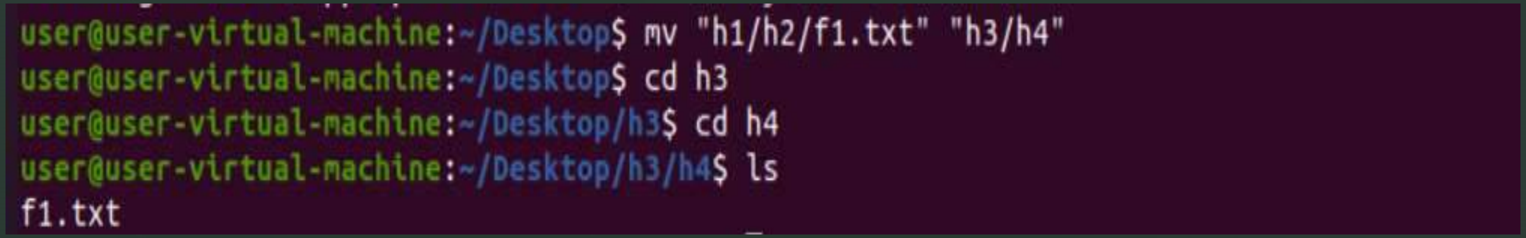
```
user@user-virtual-machine:~/Desktop$ cat f1
Hemllo
user@user-virtual-machine:~/Desktop$ cat f2
Guys
user@user-virtual-machine:~/Desktop$ mv f1 f2
user@user-virtual-machine:~/Desktop$ cat f1
cat: f1: No such file or directory
user@user-virtual-machine:~/Desktop$ cat f2
Hemllo
user@user-virtual-machine:~/Desktop$ 
```

- *mv –b file1 file2 →Move data from one file to another and having the backup of which file is about to be overwritten.*
- *It is denoted by ~filename.*



```
user@user-virtual-machine:~/Desktop$ ls
f1  f2
user@user-virtual-machine:~/Desktop$ cat f1
I am Ironman
user@user-virtual-machine:~/Desktop$ cat  f2
Reality is always disappointing
user@user-virtual-machine:~/Desktop$ mv -b f1 f2
user@user-virtual-machine:~/Desktop$ cat f2
I am Ironman
user@user-virtual-machine:~/Desktop$ ls
f2  f2~
user@user-virtual-machine:~/Desktop$ 
```

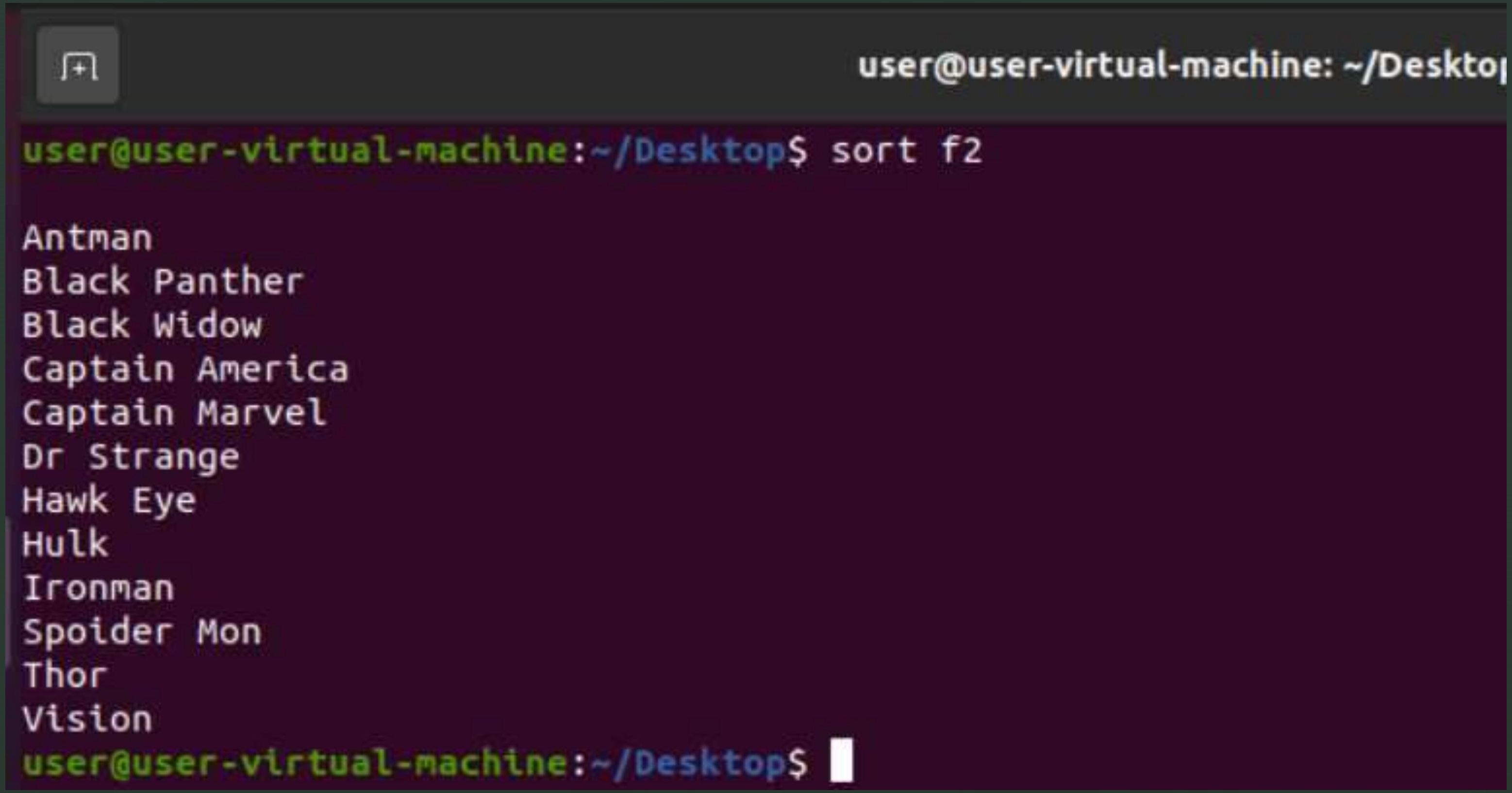
- **mv using path location**
- *mv “source path/filename” “destination path”*



```
user@user-virtual-machine:~/Desktop$ mv "h1/h2/f1.txt" "h3/h4"
user@user-virtual-machine:~/Desktop$ cd h3
user@user-virtual-machine:~/Desktop/h3$ cd h4
user@user-virtual-machine:~/Desktop/h3/h4$ ls
f1.txt
```

6. **Sort :**

- *Sort data according to dictionary order.*
- *Syntax: sort filename*

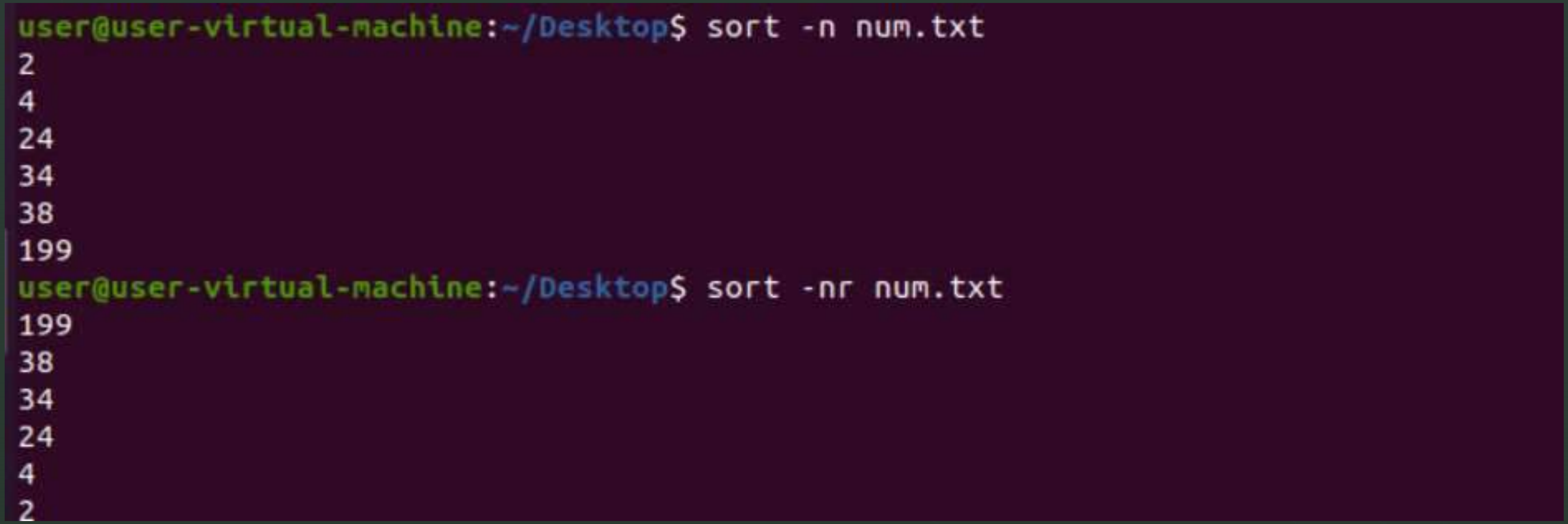


```
user@user-virtual-machine: ~/Deskto

user@user-virtual-machine:~/Desktop$ sort f2

Antman
Black Panther
Black Widow
Captain America
Captain Marvel
Dr Strange
Hawk Eye
Hulk
Ironman
Spoider Mon
Thor
Vision
user@user-virtual-machine:~/Desktop$
```

Sort according to numeric order:

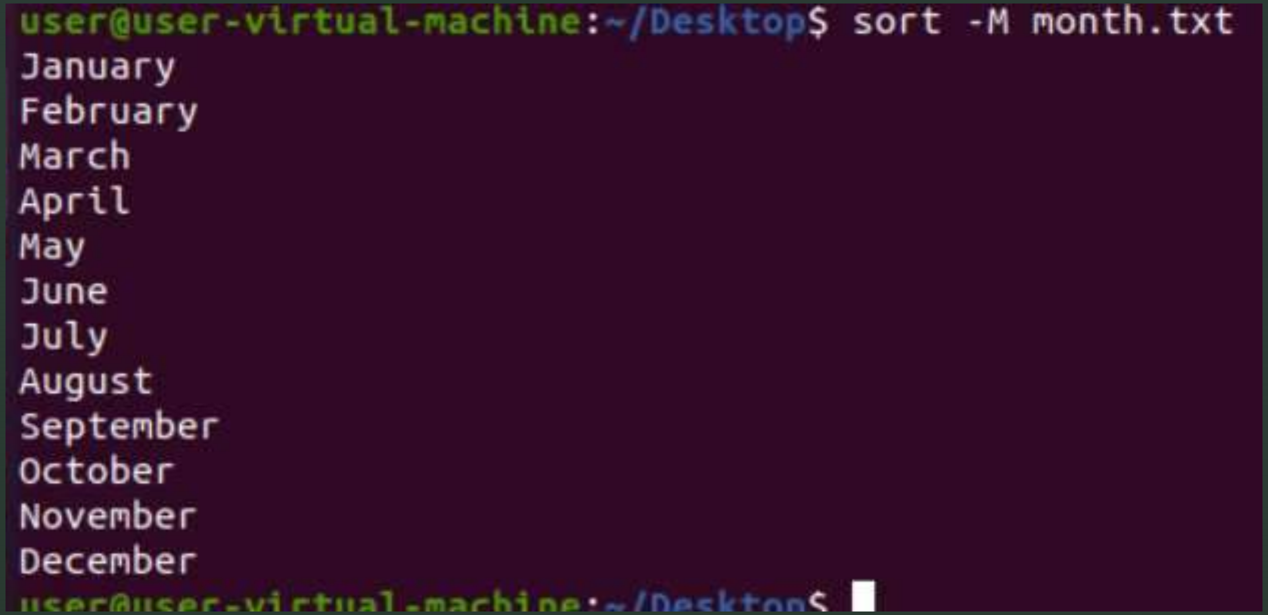
Syntax→ sort –n filename

Syntax for reverse numeric order→sort –nr filename



```
user@user-virtual-machine:~/Desktop$ sort -n num.txt
2
4
24
34
38
199
user@user-virtual-machine:~/Desktop$ sort -nr num.txt
199
38
34
24
4
2
```

- *Sorting for months given in a file.*
- *Syntax→ sort –M filename*



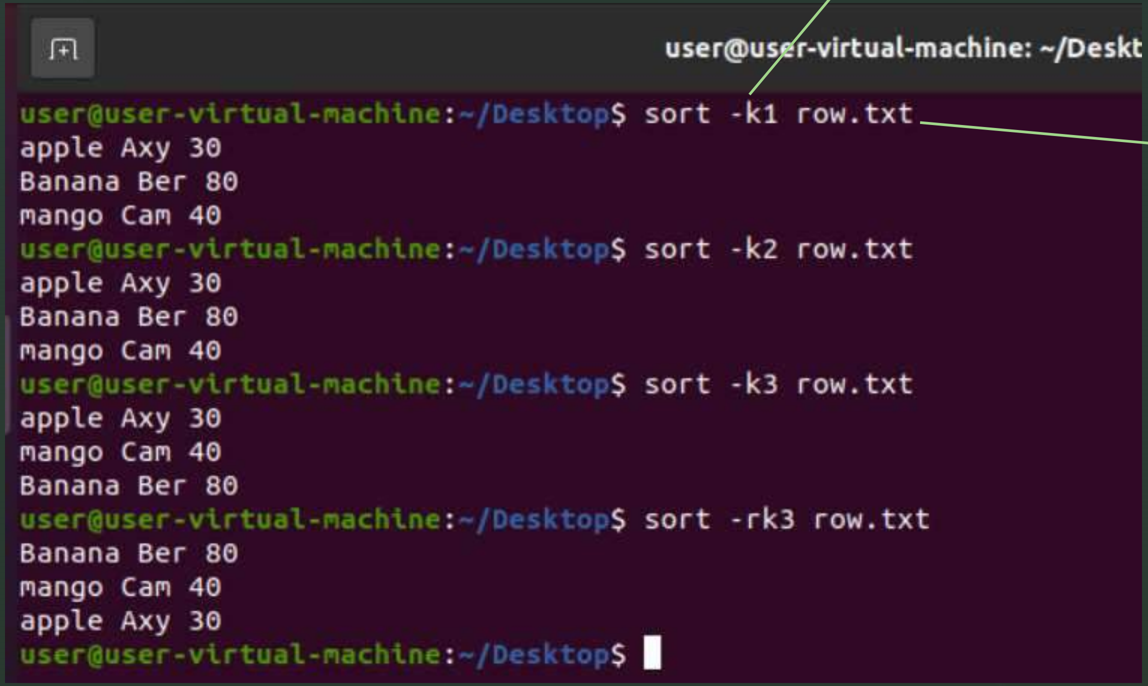
```
user@user-virtual-machine:~/Desktop$ sort -M month.txt
January
February
March
April
May
June
July
August
September
October
November
December
user@user-virtual-machine:~/Desktop$
```

- *Note:- If there is any data present in the month file which is not a month will be placed initially at the beginning of sorted month data.*

- *Sorting according to column.*
- *Syntax → sort –kColumnNo Filename*

-k with column no

Filename

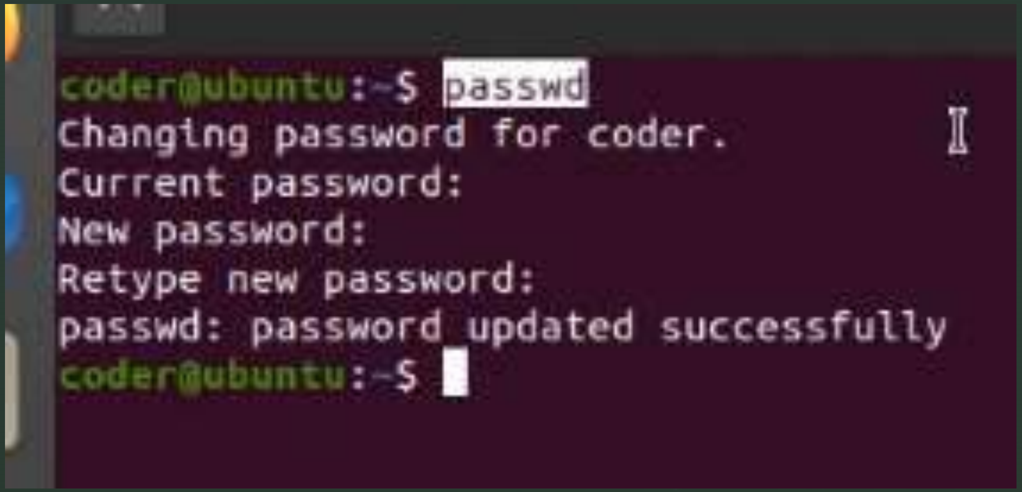


```
user@user-virtual-machine: ~/Deskt
user@user-virtual-machine:~/Desktop$ sort -k1 row.txt
apple Axy 30
Banana Ber 80
mango Cam 40
user@user-virtual-machine:~/Desktop$ sort -k2 row.txt
apple Axy 30
Banana Ber 80
mango Cam 40
user@user-virtual-machine:~/Desktop$ sort -k3 row.txt
apple Axy 30
mango Cam 40
Banana Ber 80
user@user-virtual-machine:~/Desktop$ sort -rk3 row.txt
Banana Ber 80
mango Cam 40
apple Axy 30
user@user-virtual-machine:~/Desktop$
```

7. **passwd**

Command used for changing password.

Syntax→ passwd



```
coder@ubuntu:~$ passwd
Changing password for coder.
Current password:
New password:
Retype new password:
passwd: password updated successfully
coder@ubuntu:~$
```

8. **gedit**
   - ***Command used for opening any file***
   - ***Command used for creating new file***
   - ***Syntax→gedit filename***