Command		Description/ uses
➢ **uname**	=>	Give name of **current os** name
➢ **uname -a**	=>	to display the **all system information**
➢ **uname -v**	=>	to display the **version of system**.
➢ **top**	=>	Get information of all **running program** (like taskManeger)
➢ **q**	=>	after giving top command and quit from top command then simple press **q** for out form top command.
➢ **whoami**	=>	print **current username**.
➢ **pwd / echo $PWD**	=>	this command print an **absolute path** of present **working directory.**
➢ **ls**	=>	list **directory contents**.
➢ **clear**	=>	to **clear** the **screen and take cursor** to the **left top**, and blinking cursor resided **prompt symbol**. Short cut key for clear is **[ctrl + l].**
➢ **cd**	=>	change directory this command is used for navigation throughout filesystem, i.e. by using cd command we can go from one location to another location.
➢ **mkdir**	=>	Command used to **create new directory/ directories**.
➢ **mkdir**	=>	If we want to **create more than one directory** inside same directory then use below command. **mkdir one two three** … where one two three are dir name
➢ **mkdir**	=>	If we want to **create directory inside another directory** then use below command **mkdir -p one/ two /three**

- **rmdir** => Command used for **delete only empty directory**.
- **cd /** => Command used to **goto to root directory.**
- **cd ~** => Command used to **goto to the user's home directory.**
- **cd .** => Command used to **goto to the current director /remains in current directory.**
- **cd ..** => Command used to **goto to parent directory of current directory.**
- **cd -** => Command used to **goto to prev accessed directory.**
- **cd ../..** => Command used to find **parent of parent of current directory.**
- **ls -R** => Command used to **content of the current directory get display recursively**.
- **rm -R <dirName>** => **1)** It deletes first contents of all sub directories and files of the directory recursively in it and then it delete directory
  2) To **delete non-empty directory** use
  rm -R <directoryName>
- **ls** => By default it **displays contents** of current directory in a **column wise** alphamatically sorted manner.
- **ls -1(one)** => It **displays contents** of current directory **per line one** entry in alphamatically sorted manner.
- **ls -l (l for Lion) =>** Display contents of directory in a **long list format**
- **ls -s** => It display **how many data block allocated** for file to store actual data in to the disk.
  It is show the **logical data** block.

  total n
  n = sum of **total no. of data blocks** allocated for files in that directory.
- **ls -a** => Command use to display all contents of directory **including all hidden** files with **single dot(.)** and **double dot(..)**

- **ls -A** => Command use to display all contents of directory **including all hidden** files **excluding** single dot(.) and double dot(..)
- **ctrl+u** => Command use to **erase the command line.**
- **ls -i** => Command use to **display iNode number** of all file.
- **man ls** => Command used to if we want to **get information of any command** then we man command.
- **ln -s** => Command is used to **create symbolic link** (short cut) i.e. linkable file for an existing file which is located deeply into the filesystem.
  Syntax => ln -s <src_file_path> <dest_file_path>
- **man man** => This command gives documentation.
- **man 1 <name>** => This is show user command details.
- **Man 2 <name>** => This is show function provided by kernel.
- **man 3 <name>** => This is show library functions.
- **whatis <command_name>** => this command **prints one line description** of a command from man pages
- **cat** => **1)** This command is used to **concatenates file/ files** and **display onto console.**

  2) cat command also can be used to **write data into the file** as well **append data into the file.**

  3) cat command also can be used to **display contents of regular file.**

  Syntax => **1) cat > filename.txt** => To **create a file** and **write contents into that file** & press **[ctrl + d]** to stop file writing.

  2) cat filename.txt => Display contents of the file

Note :- ***cat command cannot be used for file editing.***

- **Concatenates two file by using cat command :-**

  We can concatenate **multiple file** at a time.

  Syntax =>

  cat <file_name> <file_name> …

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l
total 8
-rw-rw-r-- 1 sunbeam sunbeam 71 Jun  8 18:13 file1.txt
-rw-rw-r-- 1 sunbeam sunbeam 36 Jun  8 18:14 file2.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ cat file1.txt file2.txt
dac sunbeam pune
dac sunbeam karad
desd sunbeam pune
dbda sunbeam pune
java programming
iphone programming
```

Whatever output we are getting in concatenate format this output is **temporary**.

- **Create file and write content inside it by using cat command:-**

  Syntax =>

  cat <file_name>

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ cat > file3.txt
dac
dbda
desd
dmc
```

- **tag command :-**

  This command is user to **reverse the file content .**

  syntax =>

  tac <file_name>

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ cat file1.txt
dac sunbeam pune
dac sunbeam karad
desd sunbeam pune
dbda sunbeam pune
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ tac file1.txt
dbda sunbeam pune
desd sunbeam pune
dac sunbeam karad
dac sunbeam pune
```

- **tac concatenate files as well as reverse =**

tac <file_name> <file_name>

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ tac file1.txt file2.txt file3.txt
dbda sunbeam pune
desd sunbeam pune
dac sunbeam karad
dac sunbeam pune
iphone programming
java programming
dmc
desd
dbda
dac
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ cat file1.txt file2.txt file3.txt
dac sunbeam pune
dac sunbeam karad
desd sunbeam pune
dbda sunbeam pune
java programming
iphone programming
dac
dbda
desd
dmc
```

- **rev command :-**

This rev command is user to **reverse line characterwise**

Syntax=>

rev <file_name>

File1.text contents are =

dac sunbeam pune
dac sunbeam karad
desd sunbeam pune
dbda sunbeam pune

after **rev file1.txt** =

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ rev file1.txt
enup maebnus cad
darak maebnus cad
enup maebnus dsed
enup maebnus adbd
```

- **rm *.txt** => This command is used to **delete all txt file** from **current directory.**
- **touch <file_name> =>** **1)** touch command is used to **update time stamps of a file.**
  2) By using touch command an **empty file** also can be created
- **ls -1 *.txt** => This command used to print only txt (text) file of directory.
- **lscpu(or /prob/cpuinfo) =>** This command used to display the **information** about the **cpu architecture**.
- **echo** => This command is used to **print message** or for **printing purpose.**
- **echo $HOME** => It contains an **absolute path of user's home** directory of **currently logged in user**.
- **echo $USER** => It contains name of **currently logged in user.**
- **echo $SHELL** => It contains an **absolute path** of **current active shell program.**
- **stat <file_name> =>** This command is used to **displays information about file.**
- **Stat -f <file_name> =>** This command is used to **displays information about file system.**
- **printenv/ env** => This command is used to **print all environment variable of shell program.**

Note:- Path is not a environment variable.

- **Read <veriable_name> =>** read command is used to **scan/read** value of variable of any type from user through terminal.

- **cp** => **copy file/files** from one location to another location

cp <src_dir_path>/file(s) <dest_dir_path>

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ touch file1.txt file2.txt
 file3.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l
total 4
-rw-rw-r-- 1 sunbeam sunbeam    0 Jun  8 19:01 file1.txt
-rw-rw-r-- 1 sunbeam sunbeam    0 Jun  8 19:01 file2.txt
-rw-rw-r-- 1 sunbeam sunbeam    0 Jun  8 19:01 file3.txt
drwxrwxr-x 3 sunbeam sunbeam 4096 Jun  8 19:01 two
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ cp *.txt ./two/three/
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -R
.:
file1.txt  file2.txt  file3.txt  two

./two:
three

./two/three:
file1.txt  file2.txt  file3.txt  four

./two/three/four:
five

./two/three/four/five:
```

Note :- whereas src_dir_path & dest_dir_path may be an absolute path or relative path.

- **mv** => **move** file/files from one location to another location

mv <src_dir_path>/file(s) <dest_dir_path>

1) whereas src_dir_path & dest_dir_path may be an absolute path or relative path.

2) If src_dir_path & dest_dir_path is same, then mv command is **used for rename purpose.**

- **Rename the file/directory name by using mv command :-**

  Syntax =>

  mv <file_name> <new_file_name>

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l
total 4
-rw-rw-r-- 1 sunbeam sunbeam    0 Jun  8 19:01 file1.txt
-rw-rw-r-- 1 sunbeam sunbeam    0 Jun  8 19:01 file2.txt
-rw-rw-r-- 1 sunbeam sunbeam    0 Jun  8 19:01 renamed_file3.txt
drwxrwxr-x 3 sunbeam sunbeam 4096 Jun  8 19:01 two
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ mv two Two
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls
file1.txt  file2.txt  renamed_file3.txt  Two
```

- **alias** => This command is used to **give nickname to command.**

  alias <nick_name> = '<command(s)>'

  alias ls = 'ls -1 -i -h -s'

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l -i -h -s
total 4.0K
14550690    0 -rw-rw-r-- 1 sunbeam sunbeam    0 Jun  8 19:15 file2.txt
14550776 4.0K drwxrwxr-x 3 sunbeam sunbeam 4.0K Jun  8 19:01 Two
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ alias ls='ls -l -i -h -s'
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls
total 4.0K
14550690    0 -rw-rw-r-- 1 sunbeam sunbeam    0 Jun  8 19:15 file2.txt
14550776 4.0K drwxrwxr-x 3 sunbeam sunbeam 4.0K Jun  8 19:01 Two
```

Note :- 1) given alias to command is **temporary**. It's valid till terminate terminal.

2) **disable** the alias effect use **unalias <file_name>**

- **Output redirection operator (>) right cheveron :-**

  - By using output redirection operator (>), program can **writes output into any specific file** instead of stdout.

Syntax =>

ls -1 > (file_name)

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022$ ls -l
total 36
drwxrwxr-x 2 sunbeam sunbeam 4096 May 30 13:03 commands
drwxrwxr-x 6 sunbeam sunbeam 4096 Jun  8 13:58 DAC
drwxrwxr-x 7 sunbeam sunbeam 4096 May 27 13:52 DBDA
-rw-rw-r-- 1 sunbeam sunbeam   16 May 23 11:45 dbda.txt
-rw-rw-r-- 1 sunbeam sunbeam   16 May 23 11:47 Dbda.txt
drwxrwxr-x 3 sunbeam sunbeam 4096 May 13 22:14 DESD
drwxrwxr-x 3 sunbeam sunbeam 4096 Apr 27 14:11 DMC
drwxrwxr-x 3 sunbeam sunbeam 4096 Apr 27 14:13 KDAC
drwxrwxr-x 8 sunbeam sunbeam 4096 May 16 13:43 PRECAT
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022$ ls -l > out.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022$ cat out.txt
total 36
drwxrwxr-x 2 sunbeam sunbeam 4096 May 30 13:03 commands
drwxrwxr-x 6 sunbeam sunbeam 4096 Jun  8 13:58 DAC
drwxrwxr-x 7 sunbeam sunbeam 4096 May 27 13:52 DBDA
-rw-rw-r-- 1 sunbeam sunbeam   16 May 23 11:45 dbda.txt
-rw-rw-r-- 1 sunbeam sunbeam   16 May 23 11:47 Dbda.txt
drwxrwxr-x 3 sunbeam sunbeam 4096 May 13 22:14 DESD
drwxrwxr-x 3 sunbeam sunbeam 4096 Apr 27 14:11 DMC
drwxrwxr-x 3 sunbeam sunbeam 4096 Apr 27 14:13 KDAC
-rw-rw-r-- 1 sunbeam sunbeam    0 Jun  9 09:55 out.txt
drwxrwxr-x 8 sunbeam sunbeam 4096 May 16 13:43 PRECAT
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022$
```

- **error redirection operator (2>):-**

  by using error redirection operator (2>), program can writes list of errors if any into any specific file instead of stderr.

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022$ ls -M
ls: invalid option -- 'M'
Try 'ls --help' for more information.
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022$ ls -M 2> err.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022$ ls -l
total 44
drwxrwxr-x 2 sunbeam sunbeam 4096 May 30 13:03 commands
drwxrwxr-x 6 sunbeam sunbeam 4096 Jun  8 13:58 DAC
drwxrwxr-x 7 sunbeam sunbeam 4096 May 27 13:52 DBDA
-rw-rw-r-- 1 sunbeam sunbeam   16 May 23 11:45 dbda.txt
-rw-rw-r-- 1 sunbeam sunbeam   16 May 23 11:47 Dbda.txt
drwxrwxr-x 3 sunbeam sunbeam 4096 May 13 22:14 DESD
drwxrwxr-x 3 sunbeam sunbeam 4096 Apr 27 14:11 DMC
-rw-rw-r-- 1 sunbeam sunbeam   64 Jun  9 09:57 err.txt
drwxrwxr-x 3 sunbeam sunbeam 4096 Apr 27 14:13 KDAC
-rw-rw-r-- 1 sunbeam sunbeam  544 Jun  9 09:55 out.txt
drwxrwxr-x 8 sunbeam sunbeam 4096 May 16 13:43 PRECAT
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022$ cat err.txt
ls: invalid option -- 'M'
Try 'ls --help' for more information.
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022$
```

- **wc** => This command is used to count no. of lines, no. of words and no. of characters (alphabates, digits, space char, newline char, special chars)

- **Input redirection operator (<) left cheveron:-**

  by using input redirection operator (<), program can takes input from any specific file instead of stdin.

1 st create one file -

cat > (file_name)

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022$ cat > india.txt
india is my country
i love india
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022$
```

2 nd display the file -

cat <file_name>

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022$ cat india.txt
india is my country
i love india
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022$
```

3 **rd count line, word, characters –**

wc > (file_name)

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022$ wc < india.txt
 2  7 33
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022$
```

- **file <file_name>** => This command is used **to check file is empty or not.**

- **Pipe command (|) :-**

  Pipe command is used to pass an **output of one program/command** as an **input to another program/command.**

  Example:- We want to count no. of lines, no. of words and no. of characters in an output of ls -l command.

  Solution-1:

  $ls -l > out.txt

  $wc < out.txt

  Solution-1:

  $ls -l | wc

- Suppose after fire command result is not fix in one page then we scroll the page up and down by using **[ctrl+shift+upArrowkey]**

  Solution :- this problem can be resolve/fix by using **pipe command** and **less** Command.

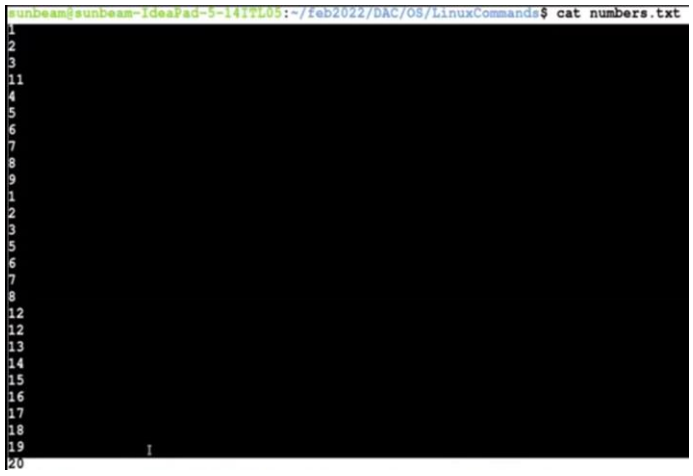
  Example :-

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~$ ls -l -a | less
```

- **Sort Numbers:-**

  1 st **create one file** (numbers.txt) and fill with numbers

  2 nd print numbers by using **cat command** (cat numbers.txt)



- **Sort numbers** by using below command :-

Sort -n <file_name>

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ sort -n numbers.txt
1
1
1
1
2
2
2
2
3
3
3
3
4
4
5
5
5
6
6
7
7
8
8
9
11
12
12
13
14
```

- **Unique number :-**

Sort -n <file_name> | uniq

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ sort -n numbers.txt  | uniq
1
2
3
4
5
6
7
8
9
11
12
13
14
15
16
17
18
19
20
```

Note :- *For unique no **data** must be in **sorted** manner*

- For store the unique data permanently:-

Sort -n <file_name> | uniq > uniq_numbers.txt

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ sort -n numbers.txt  | uniq > uniq_numbers.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ ls -l
```

- **Head Command :-**

  1 st create file.
  2 nd use head tail command on that file.

  Example :-

  Create txt file(numbers.txt) of 40 lines .

  Give command

  cat <file_name> | head

  by default it **prints first 10 lines** :-

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ cat numbers.txt | head
1
2
3
4
5
6
7
8
9
10
```

  If we wants only **first 5 lines** then :-

  cat <file_name> | head -5

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ cat numbers.txt | head -5
1
2
3
4
5
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ cat numbers.txt | head -7
1
2
3
4
5
6
7
```

If we want to **print last 10 line** then :-

cat <file_name> | tail

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ cat numbers.txt | tail
31
32
33
34
35
36
37
38
39
40
```

If we want to **print last 5 line** then :-

cat <file_name> | tail -5

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ cat numbers.txt | tail -5
36
37
38
39
40
```

Reproduction or transfer of any part of this video

If we want to **print between 31 to 35** then :-

cat <file_name> | tail | head -5

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ cat numbers.txt | tail | head -5
31
32
33
34
35
```

- **cut command:-**

  By using head & tail we sort data in horizontal but if we want sort data in vertical manner then we use **cut command.**

- If we want to **print first field** then :-

cat <file_name> | cut -d “,” -f1

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ cat records.csv
1, sachin, mumbai, 111
2, sourav, kolkatta, 999
3, rohit, maharashtra, 555
4, virat, delhi, 666
5, yuvraj, punjab, 777
6, msd, ranchi, 888
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ cat records.csv | cut -d "," -f1
1
2
3
4
5
6
```

- If we want to **print second field** then :-

cat <file_name> | cut -d “,” -f2

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ cat records.csv
1, sachin, mumbai, 111
2, sourav, kolkatta, 999
3, rohit, maharashtra, 555
4, virat, delhi, 666
5, yuvraj, punjab, 777
6, msd, ranchi, 888
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ cat records.csv | cut -d "," -f2
 sachin
 sourav
 rohit
 virat
 yuvraj
 msd
```

- If we want to **print third field** then :-

cat <file_name> | cut -d “,” -f3

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ cat records.csv
1, sachin, mumbai, 111
2, sourav, kolkatta, 999
3, rohit, maharashtra, 555
4, virat, delhi, 666
5, yuvraj, punjab, 777
6, msd, ranchi, 888
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ cat records.csv | cut -d "," -f3
 mumbai
 kolkatta
 maharashtra
 delhi
 punjab
 ranchi
```

- If we want to **print second & third field** then :-

cat <file_name> | cut -d “,” -f1

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ cat records.csv
1, sachin, mumbai, 111
2, sourav, kolkatta, 999
3, rohit, maharashtra, 555
4, virat, delhi, 666
5, yuvraj, punjab, 777
6, msd, ranchi, 888
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ cat records.csv | cut -d "," -f2,3
 sachin, mumbai
 sourav, kolkatta
 rohit, maharashtra
 virat, delhi
 yuvraj, punjab
 msd, ranchi
```

- **tr command :-**

This command used to **translate one form to another form.**

Example :-

echo “Sachin” | tr “A-Z” “a-z”

echo “SACHIN” | tr “A-Z” “a-z”

echo “Sachin Pawar” | tr “A-Z” “a-z”

echo “Sachin Pawar” | tr “A-Za-z” “a-zA-Z”

echo $PATH | tr “:” “\n”

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ echo "Sachin" | tr "A-Z" "a-z"
sachin
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ echo "SACHIN" | tr "A-Z" "a-z"
sachin
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ echo "Sachin Pawar" | tr "A-Z" "a-z"
sachin pawar
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ echo "Sachin Pawar" | tr "A-Za-z" "a-zA-Z"
sACHIN pAWAR
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$ echo $PATH | tr ":" "\n"
/usr/local/sbin
/usr/local/bin
/usr/sbin
/usr/bin
/sbin
/bin
/usr/games
/usr/local/games
/snap/bin
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands$
```

- **chmod (change mode bits) :-**

By means of changing mode bits chmod command is used to **change access permission** i.e. **either assign or remove access perms** of a file to / from **user or owner, group members and others.**

- **chmod +x <fileName>** => assign **execute permission** to all.

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l file1.txt
-rw-rw-r-- 1 sunbeam sunbeam 17 Jun  9 11:51 file1.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ chmod +x file1.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l file1.txt
-rwxrwxr-x 1 sunbeam sunbeam 17 Jun  9 11:51 file1.txt
```

- **chmod -x <fileName>** => **remove execute** permission from all

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l file1.txt
-rwxrwxr-x 1 sunbeam sunbeam 17 Jun  9 11:51 file1.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ chmod -x file1.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l file1.txt
-rw-rw-r-- 1 sunbeam sunbeam 17 Jun  9 11:51 file1.txt
```

- **chmod u+x <fileName>** => assign **execute permission to user/owner** only.

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l file1.txt
-rw-rw-r-- 1 sunbeam sunbeam 17 Jun  9 11:51 file1.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ chmod u+x file1.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l file1.txt
-rwxrw-r-- 1 sunbeam sunbeam 17 Jun  9 11:51 file1.txt
```

- **chmod u-x <fileName>** => **remove execute** permission from **user/owner** only.

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ chmod u-x file1.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l file1.txt
-rw-rw-r-- 1 sunbeam sunbeam 17 Jun  9 11:51 file1.txt
```

- **chmod g+x <filename>** => **Assign execute** permission to **group members** only.

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ chmod g+x file1.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l file1.txt
-rw-rwxr-- 1 sunbeam sunbeam 17 Jun  9 11:51 file1.txt
```

- **chmod o+x <filename>** => **Assign execute** permission to **others members** only.

- **chmod o-x <filename>** => **Remove execute** permission from **others** only.

directory file => It is a special type of file can be created by the user by using mkdir command, whereas to maintain contents inside it is the responsibility of the system (filesystem).

read permission is there for directories file => enables to apply **ls** command on it.

write perm is there for dir file => enables to create new files and sub dir's inside it.

execute perm is there for dir file => enables to apply **cd** command on it.

- **octal format:-**

General Syntax:- chmod 0XYZ <filename>

In chmod command in a arg-1 => leading 0 indicates its an octal constant.

X – it indicates **access permission for user or owner** of a file
Y – it indicates **access permission for group** members
Z – it indicates **access permission for others**

read – 4
write – 2
execute – 1

0 => nothing
1 => execute only
2 => write only
3 => write + execute
4 => read only
5 => read + execute
6 => read + write
7 => read + write + execute

- 1 st **create file** (cat > (file_name))

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ cat > file.txt
one
two
three
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l
total 4
-rw-rw-r-- 1 sunbeam sunbeam 14 Jun  9 12:42 file.txt
```

- Change access permissions of **user, group, others** :-

- Denied all permission by using **0000**

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l
total 4
-rw-rw-r-- 1 sunbeam sunbeam 14 Jun  9 12:42 file.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ chmod 0000 file.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l
total 4
---------- 1 sunbeam sunbeam 14 Jun  9 12:42 file.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$
```

- Give all **permission to user, group, others** by using **0777 :-**

```
---------- 1 sunbeam sunbeam 14 Jun  9 12:42 file.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ chmod 0777 file.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l
total 4
-rwxrwxrwx 1 sunbeam sunbeam 14 Jun  9 12:42 file.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$
```

- **User/Owner** give **read,write and execute permission**, **group** having only **execute permission** and **other** having only **execute permission** by using **0711 :-**

```
---------- 1 sunbeam sunbeam 14 Jun  9 12:42 file.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ chmod 0711 file.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l
total 4
-rwx--x--x 1 sunbeam sunbeam 14 Jun  9 12:42 file.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$
```

- **User/Owner** give **read,write and execute permission**, **group** having **read** & **write permission** and **other** having only **execute permission** by using **0761 :-**

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ chmod 0761 file.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l
total 4
-rwxrw---x 1 sunbeam sunbeam 14 Jun  9 12:42 file.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$
```

- **User/Owner** give **read,write and execute permission**, **group** having **read** & **execute permission** and **other** having only **execute permission** by using **0751 :-**

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ chmod 0751 file.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$ ls -l
total 4
-rwxr-x--x 1 sunbeam sunbeam 14 Jun  9 12:42 file.txt
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/one$
```

- Likewise we give **any permission to user, group and others**.
- **w** => This command is used for **Show who is logged** on and **what they are doing**.
- **users** => This command is used to **find how many users currently logged in.**

- **For new user log in :-**

  Step1:- adduser

  Command => **sudo adduser <user_name>**

  Step2:- log in user

  Command => **ctrl+alt+f**

  Step3:- for checking user log in successfully or not

  Command => **w**

- **For logout from multiple user :-**

  Command => **ctrl+alt+f1.**

- **For delete user :**

  Command => **sudo deluser <user_name>**

  Note :- For **delete** user it is **mandatory** to that **user is logged out** and then this user is delete otherwise it show error

- **who** => This command is use to **display who is logged on** (that means current who user is logged)

- **grep (***globally search regular expression and print***):-**
  - In file we want to search string inside file then use **grep command**

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ cat sunbeam.txt
advanced java
1. c programming language
2. c++ programming language
3. operating system concepts with linux programming
4. algorithms and data structures
5. embedded operating system
6. device deriver programming
7. android programming
8. iphone programming
9. advanced web programming
10. database technologies
11. java programming-1
12. java programming-2

program
program is a set of instructions
process is program in execution
```

grep "java" <file_name>

grep command search for a string in a given file and whenever match is found it prints that line.

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep "java" sunbeam.txt
advanced java
11. java programming-1
12. java programming-2
```

grep command is case sencitive suppose you search **"Java"** then it not find such string in output doe's not print anything

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep "Java" sunbeam.txt
```

And in above case if you want to ignore it then use **-i** (i for ignore)

grep -i "Java" <file_name>

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep -i "Java" sunbeam.txt
advanced java
11. java programming-1
12. java programming-2
```

- **-v command:-**

This command is used for give **inverted output.**

Example :-

grep -v “java” <file_name>

by using this command print that lines which not match with “java” string.

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep -v "java" sunbeam.txt
1. c programming language
2. c++ programming language
3. operating system concepts with linux programming
4. algorithms and data structures
5. embedded operating system
6. device deriver programming
7. android programming
8. iphone programming
9. advanced web programming
10. database technologies

program
program is a set of instructions
```

New Concept :-

grep “program” <file_name>

In that case print such line which content even “program” is a substring of any string then it is consider match is found.

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep "program" sunbeam.txt
1. c programming language
2. c++ programming language
3. operating system concepts with linux programming
6. device deriver programming
7. android programming
8. iphone programming
9. advanced web programming
11. java programming-1
12. java programming-2
program
program is a set of instructions
process is program in execution
```

- **-w Command :-**

grep -w “program” <file_name>

-w command :- prints the lines containg matches that form **whole word**

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep -w "program" sunbeam.txt
program
program is a set of instructions
```

Whichever string is menssion over here it should be match as whole word, if they have substring of any string so it consider as match is not found.

- **-c command :-**

“This command is used **how many numbers of time given string is found** “

grep -c “string_name” <file_name>

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep -c "program" sunbeam.txt
12
```

- **-n command:-**

“If we -n command then it **print line no** “

grep -n “StringName” <filename>

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep -n "java" sunbeam.txt
1:advanced java
12:11. java programming-1
13:12. java programming-2
```

- **Two grep command can be use same time :-**

grep -n -i “string_name” <file_name>

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/Lin  mands/commands$ cat sunbeam.txt
advanced java
1. c programming language
2. c++ programming language
3. operating system concepts with linux programming
4. algorithms and data structures
5. embedded operating system
6. device deriver programming
7. android programming
8. iphone programming
9. advanced web programming
10. database technologies
11. java programming-1
12. java programming-2

program
program is a set of instructions
process is program in execution
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep -n -i "System" sunbeam.txt
4:3. operating system concepts with linux programming
6:5. embedded operating system
```

- **A & B command:-**

grip -A2 “String _name” <file_name>

this command is user to print lines after found string line
in that example A2 means after found line two lines print

grip -B2 “String_name” <file_name>

this command is user to print lines before found string line
in that example B2 means before found line two lines print

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ cat sunbeam.txt
advanced java
1. c programming language
2. c++ programming language
3. operating system concepts with linux programming
4. algorithms and data structures
5. embedded operating system
6. device deriver programming
7. android programming
8. iphone programming
9. advanced web programming
10. database technologies
11. java programming-1
12. java programming-2

program
program is a set of instructions
process is program in execution
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep "device" sunbeam.txt
6. device deriver programming
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep -A2 "device" sunbeam.txt
6. device deriver programming
7. android programming
8. iphone programming
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep -B2 "device" sunbeam.txt
4. algorithms and data structures
5. embedded operating system
6. device deriver programming
```

- **Center (-c) command :-**

  Center command used for print lines above center and below center that means we pass one string and this string is found at line from that line print lines above and below

Syntax => **grep -c<how_many_lines_print> “string_name” <file_name>**

Example:-

grep -c3 “device” sunbeam.txt

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep -C3 "device" sunbeam.txt
3. operating system concepts with linux programming
4. algorithms and data structures
5. embedded operating system
6. device deriver programming
7. android programming
8. iphone programming
9. advanced web progra
```

Reproduction or transfer of any part of this video

- grep command is also used to **search a string** in **files(more than one file)** at once and wherever **match is found file and that line** will be print.

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep "program" *.txt
grep_command.txt:$grep -w "program" sunbeam.txt --> prints the lines containg matches that form whole word
sunbeam.txt:1. c programming language
sunbeam.txt:2. c++ programming language
sunbeam.txt:3. operating system concepts with linux programming
sunbeam.txt:6. device deriver programming
sunbeam.txt:7. android programming
sunbeam.txt:8. iphone programming
sunbeam.txt:9. advanced web programming
sunbeam.txt:11. java programming-1
sunbeam.txt:12. java programming-2
sunbeam.txt:program
sunbeam.txt:program is a set of instructions
sunbeam.txt:process is program in execution
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$
```

- **Anchoring characters:-**

There are **two type** of anchoring characters - 1) **carot(^)**
2) **doller($)**

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ cat taste.txt
this
biscuit
isn't
tasty,
but
india is my countryin
that
cake
is
really
good.
```

- **^ command :-**

Line beginning with given String

Example:-

grep "^is" teast.txt

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep "^is" taste.txt
isn't
is
```

- **$ command:-**

Line end with given String

Example:-

grep "is$" teast.txt

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep "is$" taste.txt
this
is
```

- We can use **both carot and doller at same time**

Example:-

grep "^is$" teast.txt

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep "^is$" taste.txt
is
```

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ cat select.txt
bg
bag
beg
big
bog
bug
b*g
```

- **dot (.) character :-**

    - It is also call **period** character.
    - If between character either one or more any character is their so it is treated as a matches found.

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep "b.g" select.txt
bag
beg
big
bog
bug
b*g
```

- **Bracket expression : -**

    1) Bracket expression a ([a]) means in between b & g single a is there so it is treated as match is found.

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep "b[a]g" select.txt
bag
```

    2) Bracket expression [a | u | i] / [aui] means in between b & g single a, u, i is there so it is treated as match is found.

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep "b[a|u|i]g" select.txt
bag
big
bug

sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep "b[aui]g" select.txt
bag
big
bug

sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep "b[a-z]g" select.txt
bag
beg
big
bog
bug
```

- Here I want only b*g and I am gives command **grep "b*g" select.txt** and it show all file content because here it **takes * means** all but I want **only b*g** string

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep "b*g" select.txt
bg
bag
beg
big
bog
bug
b*g
```

For that one solution here

1) grep "b*g" select.txt

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep "b\*g" select
b*g
```

Here we sapress the effect of * by using \

2) grep "b[*]g" select.txt

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep "b[*]g" select.txt
b*g
```

3) grep -F "b*g" select.txt

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ grep -F "b*g" select.txt
b*g
```

4) fgrep "b*g" select.txt

- **Repetition operators :-**

? -- at most once i.e. 0 or 1 no. of times occurrences
***** -- zero or more occurrences
+ -- one or more occurrences
{ n } -- exact n no. of occurrences
{ n, } -- more or equal to n no. of occurrences
{ , m } -- at most m no. of occurrences
{ n, m } -- more than or equal to n and less than or equal to m.

- **Display content of file repeat.txt**

cat <file_name>

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ cat repeat.txt
ww
wow
woow
wooow
woooow
wooooow
woooooow
wooooooow
woooooooow
wooooooooow
```

- * command:-

**** stand for zero or more occurrences***

(here meaning of * is o occurred zero time or more than zero)

egrep "wo*w" repeat.txt

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ egrep "wo*w" repeat.txt
ww
wow
woow
wooow
woooow
wooooow
woooooow
wooooooow
woooooooow
wooooooooow
```

- **? command :-**

? stand for at most once i.e. 0 or 1 no. of times occurrences

(here meaning of ? is o occurred zero time or one time)

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ egrep "wo?w" repeat.tx
ww
wow
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$
```

- **+ command :-**

+ stand for one or more occurrences
(here meaning of + is o occurred one time or more than one time)

egrep “wo+w” repeat.txt

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ egrep "wo+w" repeat.txt
wow
woow
wooow
woooow
wooooow
woooooow
wooooooow
woooooooow
```

- **{ n } command:-**

exact n no. of occurrences
(here meaning of { n } is o occurred exactly n times)

egrep “wo{4}w” repeat.txt

```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ egrep "wo{4}w" repeat.txt
woooow
```

- **{ n, }command :-**

more or equal to n no. of occurrences
(here meaning of { n,} is o occurred exactly n time or more than n.)

egrep “wo{4,}w” repeat.txt

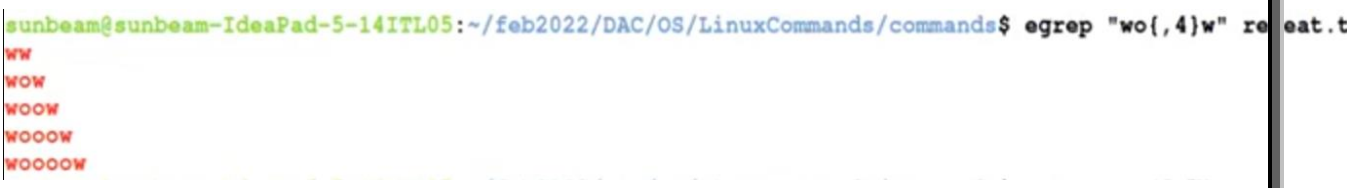
```
sunbeam@sunbeam-IdeaPad-5-14ITL05:~/feb2022/DAC/OS/LinuxCommands/commands$ egrep "wo{4,}w" re
woooow
wooooow
woooooow
wooooooow
woooooooow
```

- **{ ,m } command :-**

Less than m or equal to m no. of occurrences

(here meaning of { ,m } is o occurred exactly m time or less than m.)

egrep “wo{,4}w” repeat.txt

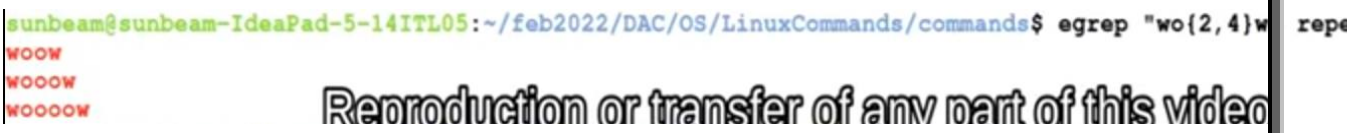


- **{n,m} command :-**

more than or equal to n and less than or equal to m

(here meaning of { n,m } is o occurred more than or equal to n and less than or equal to m)

egrep “wo{2,4}w” repeat.txt



Reproduction or transfer of any part of this video