Symbiosis Skills and Professional University

Skill Journal

Name: Yograj Anant Gadekar **Date: 04/10/2021**
PRN:
School: School of Data Science
Course: Data Associate (Data Science) DA13
Module Name: Python for Data Analysis **/** Managing with Data **/** Analyzing Data from Disparate Sources **(tick any one)**

1. **Skill Activity Number :** 07

2. **Title :** Linux Admin Commands

3. **Skills / Competencies to be acquired : VMware**

4. **Duration: 1 day.**

5. **What is the purpose of the activity?**
   Purpose is to understand the basic commands of linux to perform operations

6. **Steps Performed in this activity?**
   Follow step by step commands to performed this activity

7. **What resources / materials / equipment / tools did you use for this activity?**
   VMware, MS word

8. **What skills did you acquire?**
   Creation of many files, directories, copy, move, cat etc. so many commands we can learn in this activity.

9. **Time taken to complete this activity?**
   1Hrs

Linux Admin Commands

1) pwd- (Present working directory) command writes the full pathname of the current working directory

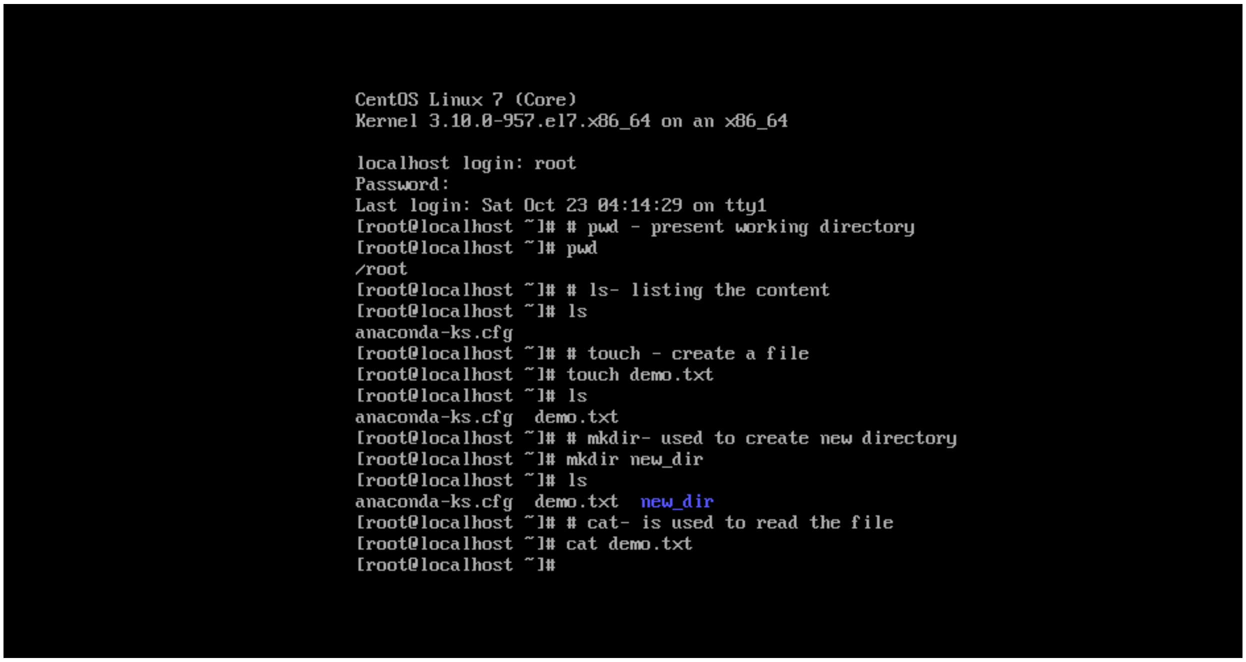
Syntax: pwd

2) ls – ls command is used to list files or listing the content
Syntax: ls

3) touch- used to create a file
Syntax: touch filename

4) mkdir- This command is used to create new directory
Syntax: mkdir foldername

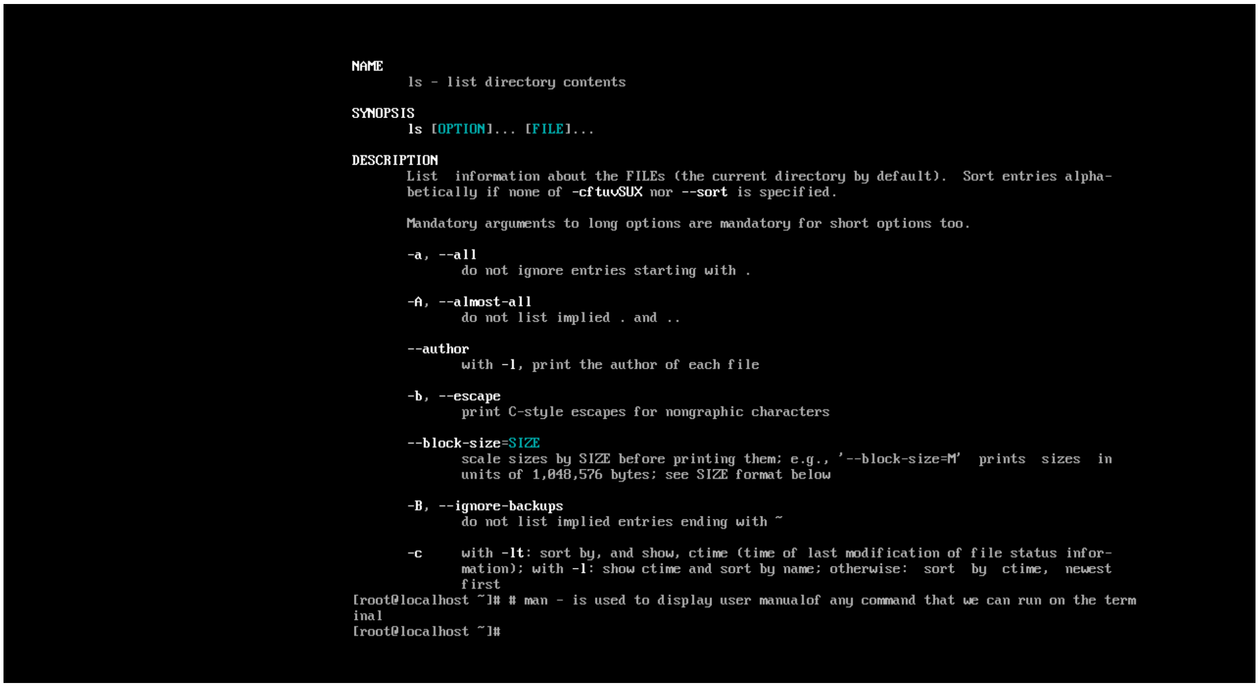
5) cat- cat command is used to read the file
Syntax: cat filename



```
CentOS Linux 7 (Core)
Kernel 3.10.0-957.el7.x86_64 on an x86_64

localhost login: root
Password:
Last login: Sat Oct 23 04:14:29 on tty1
[root@localhost ~]# # pwd - present working directory
[root@localhost ~]# pwd
/root
[root@localhost ~]# # ls- listing the content
[root@localhost ~]# ls
anaconda-ks.cfg
[root@localhost ~]# # touch - create a file
[root@localhost ~]# touch demo.txt
[root@localhost ~]# ls
anaconda-ks.cfg  demo.txt
[root@localhost ~]# # mkdir- used to create new directory
[root@localhost ~]# mkdir new_dir
[root@localhost ~]# ls
anaconda-ks.cfg  demo.txt  new_dir
[root@localhost ~]# # cat- is used to read the file
[root@localhost ~]# cat demo.txt
[root@localhost ~]#
```

6) man- command is used to display user manual of any command that we can run on the terminal

Syntax: man command name



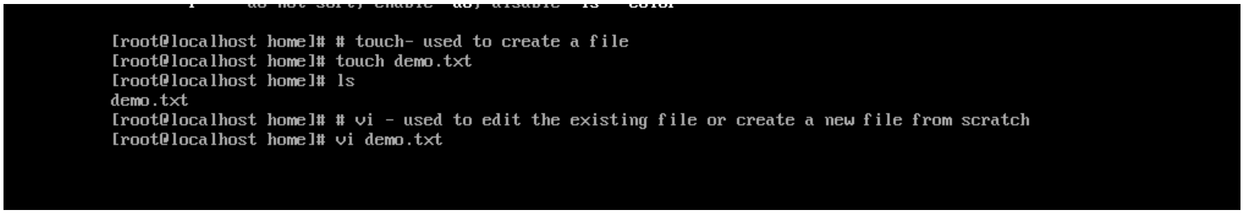
```
NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information about the FILEs (the current directory by default).  Sort entries alpha-
       betically if none of -cftuvSUX nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              scale sizes by SIZE before printing them; e.g., '--block-size=M'  prints  sizes  in
              units of 1,048,576 bytes; see SIZE format below

       -B, --ignore-backups
              do not list implied entries ending with ~

       -c     with -lt: sort by, and show, ctime (time of last modification of file status infor-
              mation); with -l: show ctime and sort by name; otherwise:  sort  by  ctime,  newest
              first
[root@localhost ~]# # man - is used to display user manualof any command that we can run on the term
inal
[root@localhost ~]#
```

7) touch- used to create a new file

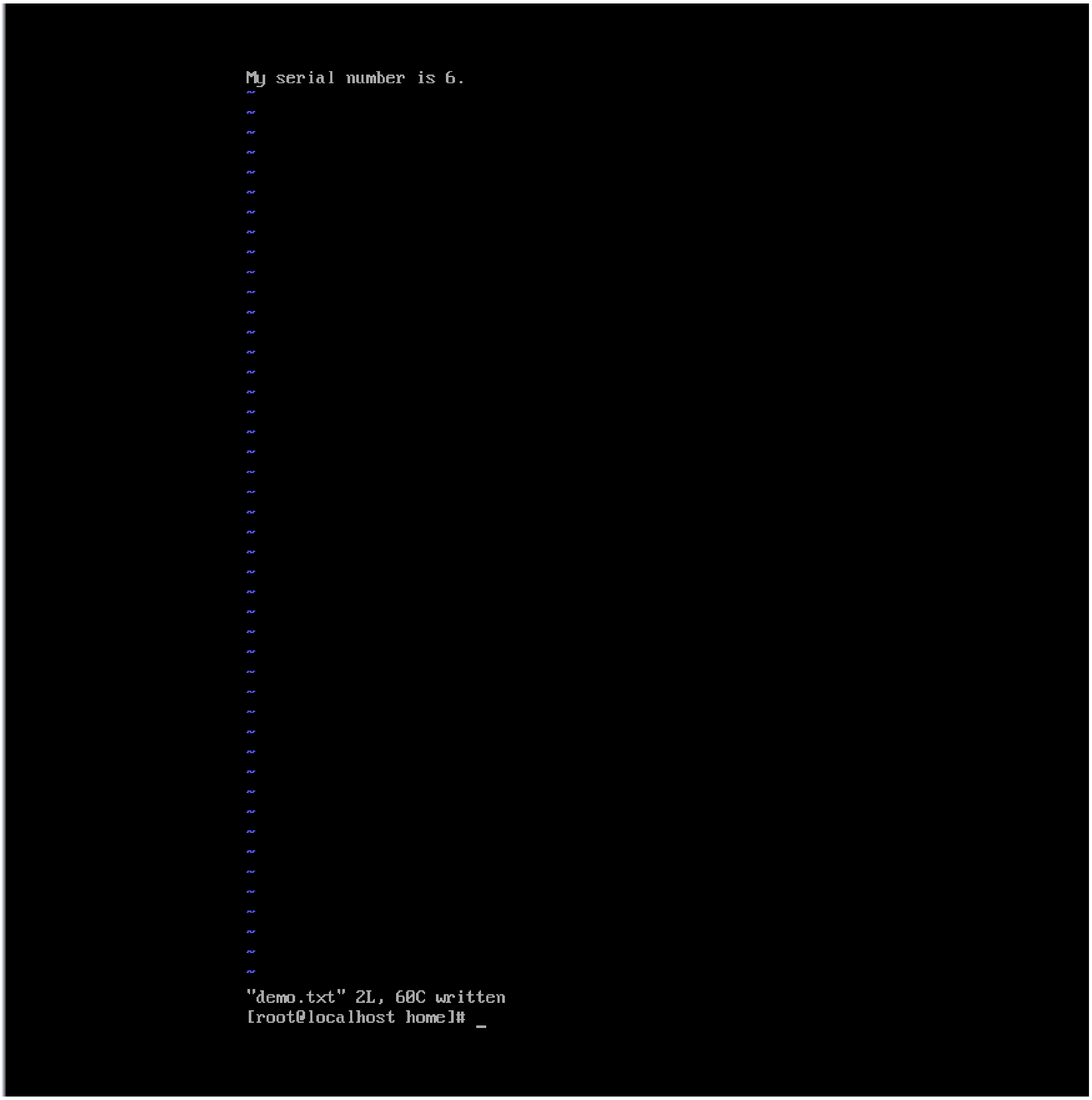
Syntax: touch filename

8) vi – Used to edit the existing file or create a new file from scratch.

Syntax: vi filename



```
        f      do not sort, enable -aU, disable -ls --color

[root@localhost home]# # touch- used to create a file
[root@localhost home]# touch demo.txt
[root@localhost home]# ls
demo.txt
[root@localhost home]# # vi - used to edit the existing file or create a new file from scratch
[root@localhost home]# vi demo.txt
```

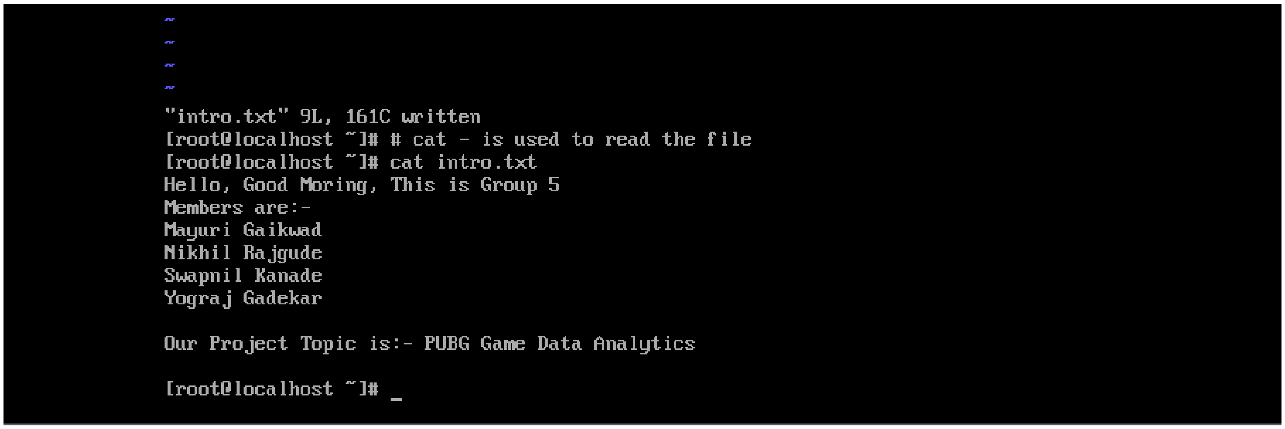


```
My serial number is 6.
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"demo.txt" 2L, 60C written
[root@localhost home]# _
```

Insert mode – This mode enables you to insert text into file
To come in insert mode you simply type **i .**

Escape mode-(Last line)- Line mode invoked by typing a colon**(:)**
After that type **wq (w-** is used for save the file and **q-** is used for quit**)**

9) cat- used to read the file
Syntax: cat demo.txt



```
~
~
~
~
"intro.txt" 9L, 161C written
[root@localhost ~]# # cat - is used to read the file
[root@localhost ~]# cat intro.txt
Hello, Good Moring, This is Group 5
Members are:-
Mayuri Gaikwad
Nikhil Rajgude
Swapnil Kanade
Yograj Gadekar

Our Project Topic is:- PUBG Game Data Analytics

[root@localhost ~]# _
```

10) mkdir- create new folder

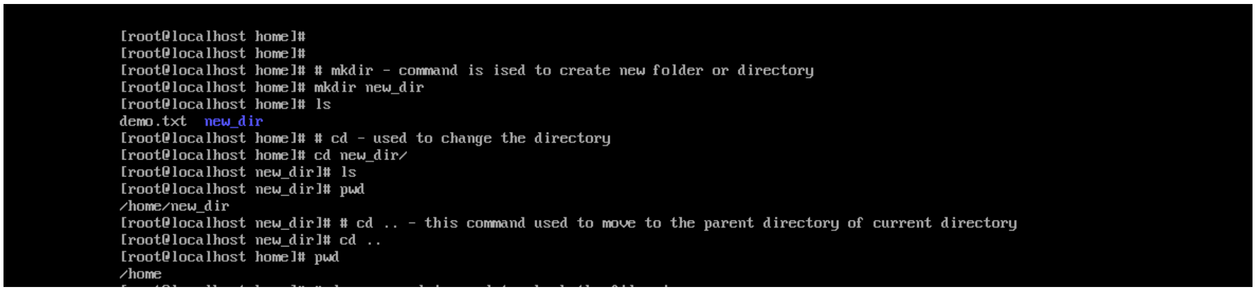
Syntax: mkdir new_dir

11) cd – change the directory

Syntax: cd

12) cd .. – used to move to the parent directory of current directory

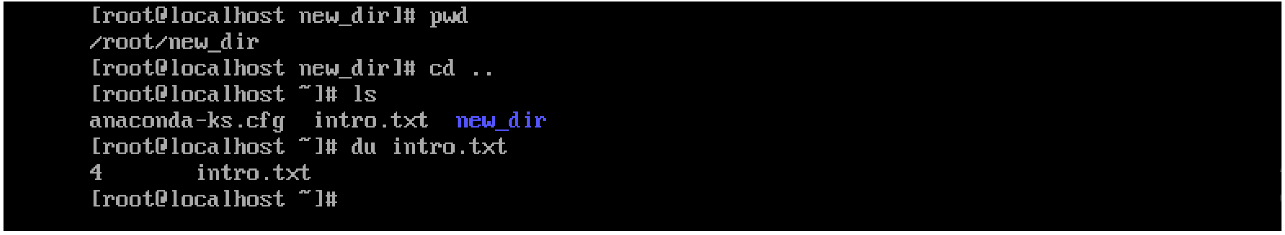
Syntax: cd ..



```
[root@localhost home]#
[root@localhost home]#
[root@localhost home]# # mkdir - command is ised to create new folder or directory
[root@localhost home]# mkdir new_dir
[root@localhost home]# ls
demo.txt  new_dir
[root@localhost home]# # cd - used to change the directory
[root@localhost home]# cd new_dir/
[root@localhost new_dir]# ls
[root@localhost new_dir]# pwd
/home/new_dir
[root@localhost new_dir]# # cd .. - this command used to move to the parent directory of current directory
[root@localhost new_dir]# cd ..
[root@localhost home]# pwd
/home
```

13) du - This command is used to move to the parent directory of current directory

Syntax: du filename



```
[root@localhost new_dir]# pwd
/root/new_dir
[root@localhost new_dir]# cd ..
[root@localhost ~]# ls
anaconda-ks.cfg  intro.txt  new_dir
[root@localhost ~]# du intro.txt
4       intro.txt
[root@localhost ~]#
```

14) rmdir- This command is used to remove the directory.(it remove only empty direcorty)

Syntax: rmdir foldername

15) rm – command is used to remove the file

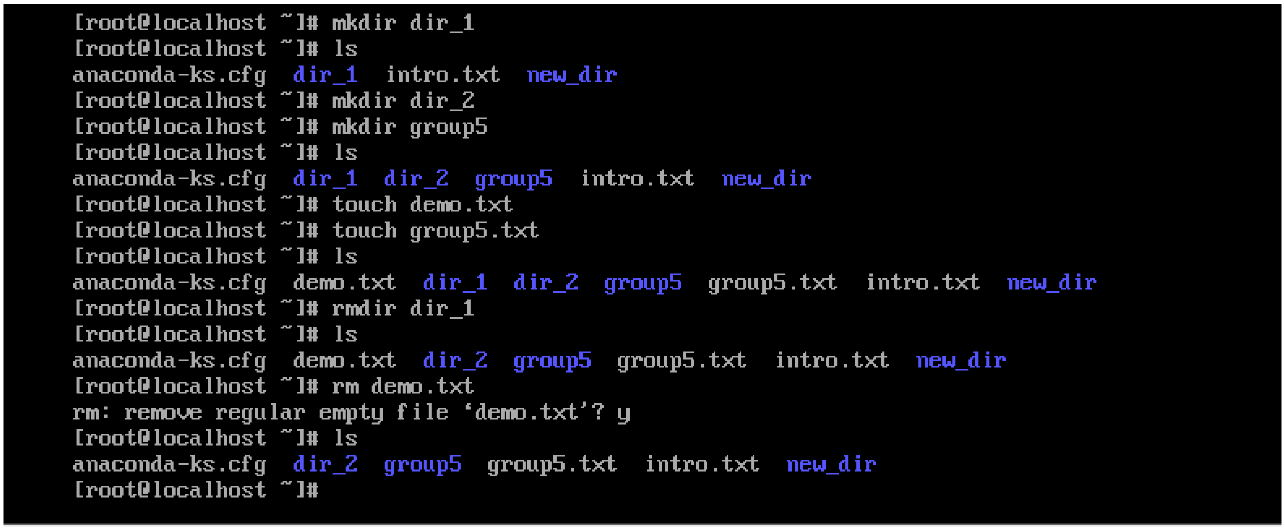
Syntax: rm filename

16) rm -r -To remove non-empty directories and all the files within them recursively, use the -r(recursive) option.

Syntax: rm -r dirname

17) rm -rf -Without asking for your consent, removes files for which you do not have write access permission.

Syntax: rm – rf dirname



```
[root@localhost ~]# mkdir dir_1
[root@localhost ~]# ls
anaconda-ks.cfg  dir_1  intro.txt  new_dir
[root@localhost ~]# mkdir dir_2
[root@localhost ~]# mkdir group5
[root@localhost ~]# ls
anaconda-ks.cfg  dir_1  dir_2  group5  intro.txt  new_dir
[root@localhost ~]# touch demo.txt
[root@localhost ~]# touch group5.txt
[root@localhost ~]# ls
anaconda-ks.cfg  demo.txt  dir_1  dir_2  group5  group5.txt  intro.txt  new_dir
[root@localhost ~]# rmdir dir_1
[root@localhost ~]# ls
anaconda-ks.cfg  demo.txt  dir_2  group5  group5.txt  intro.txt  new_dir
[root@localhost ~]# rm demo.txt
rm: remove regular empty file ‘demo.txt’? y
[root@localhost ~]# ls
anaconda-ks.cfg  dir_2  group5  group5.txt  intro.txt  new_dir
[root@localhost ~]#
```

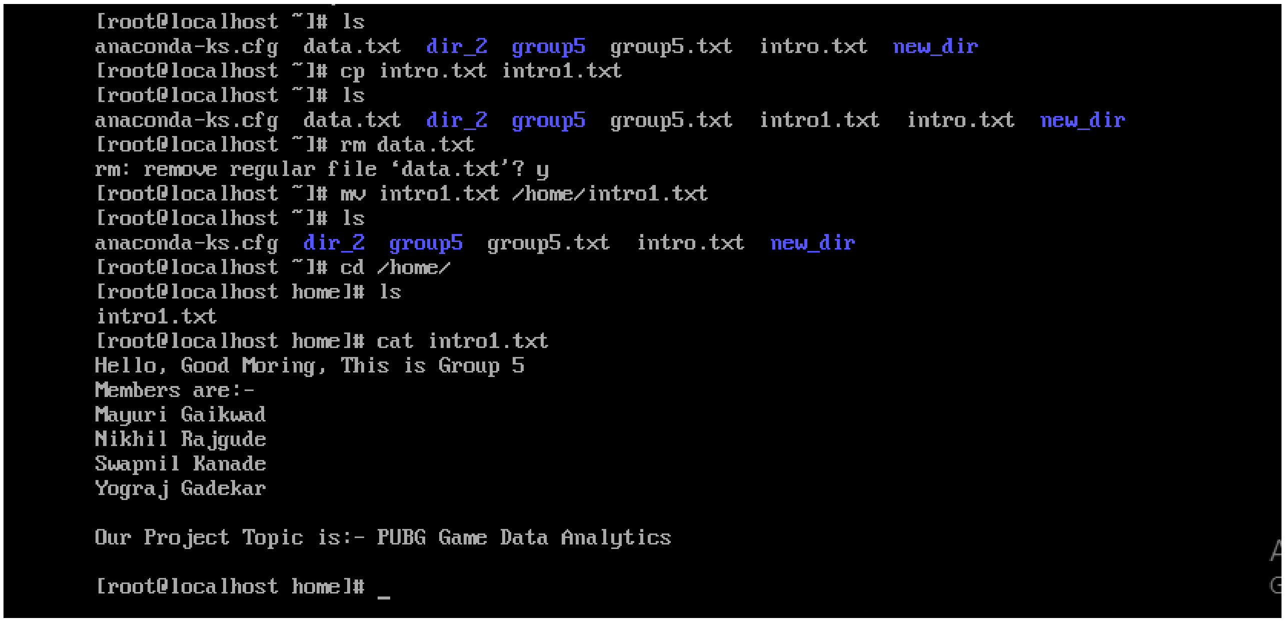
18) cp command – cp stands for **copy** This command is used to copy a file or directory into existing or to some other directory as given in the path, and it can also rename a copied file(that is the new one)

Syntax: cp <existing file name> <new file name>

19) mv – mv command is used to move existing file or directory from one location to another. It is also used to rename a file or directory. If you want to rename a single directory or file then **'mv'** option will be better to use.

Syntax : mv [Option] source destination

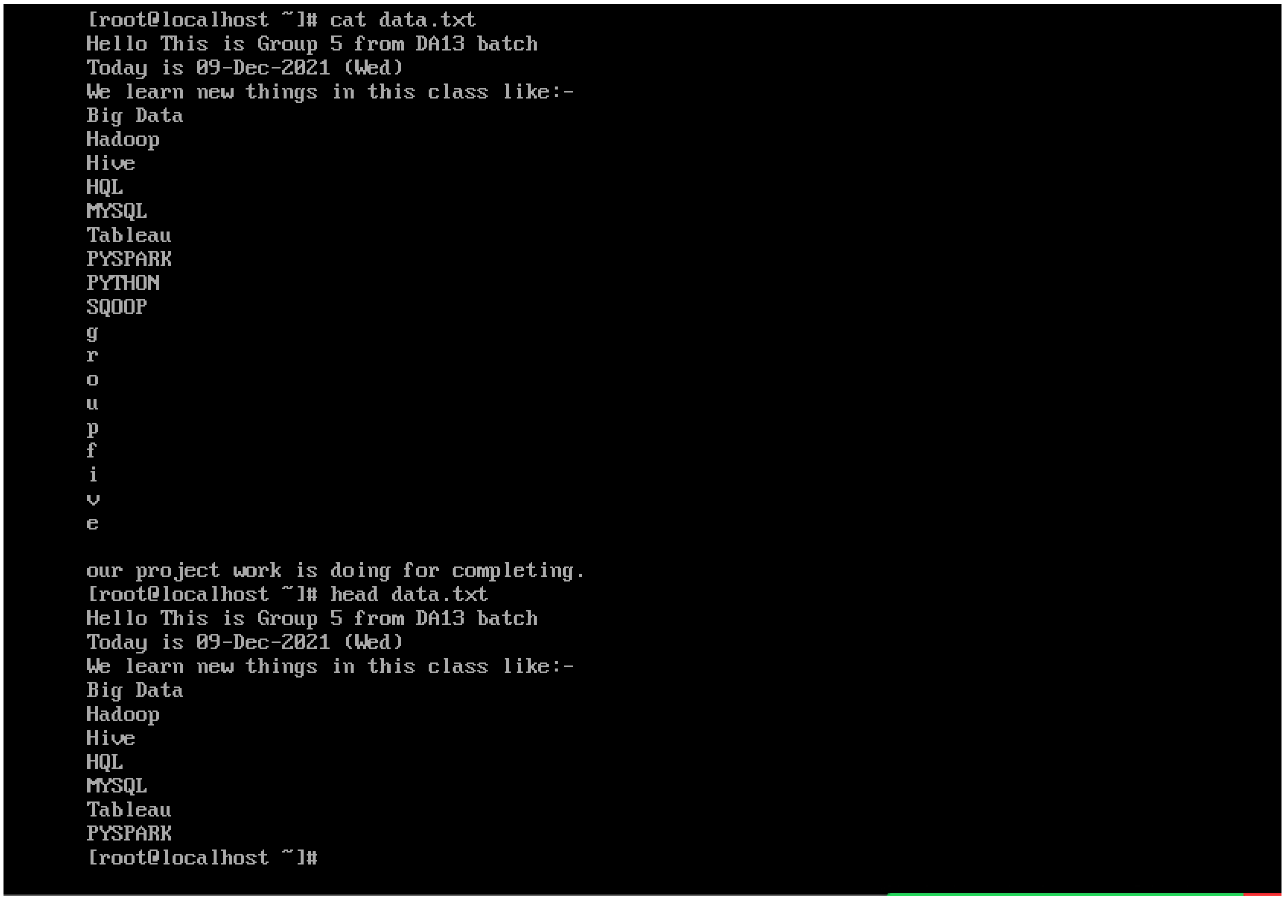
Example: mv week.txt /home/week1.txt



```
[root@localhost ~]# ls
anaconda-ks.cfg  data.txt  dir_2  group5  group5.txt  intro.txt  new_dir
[root@localhost ~]# cp intro.txt intro1.txt
[root@localhost ~]# ls
anaconda-ks.cfg  data.txt  dir_2  group5  group5.txt  intro1.txt  intro.txt  new_dir
[root@localhost ~]# rm data.txt
rm: remove regular file ‘data.txt’? y
[root@localhost ~]# mv intro1.txt /home/intro1.txt
[root@localhost ~]# ls
anaconda-ks.cfg  dir_2  group5  group5.txt  intro.txt  new_dir
[root@localhost ~]# cd /home/
[root@localhost home]# ls
intro1.txt
[root@localhost home]# cat intro1.txt
Hello, Good Moring, This is Group 5
Members are:-
Mayuri Gaikwad
Nikhil Rajgude
Swapnil Kanade
Yograj Gadekar

Our Project Topic is:- PUBG Game Data Analytics

[root@localhost home]# _
```

20) **head command-** The 'head' command displays the starting content of a file. By default, it displays starting 10 lines of any file.

Syntax: head <file name>



```
[root@localhost ~]# cat data.txt
Hello This is Group 5 from DA13 batch
Today is 09-Dec-2021 (Wed)
We learn new things in this class like:-
Big Data
Hadoop
Hive
HQL
MYSQL
Tableau
PYSPARK
PYTHON
SQOOP
g
r
o
u
p
f
i
v
e

our project work is doing for completing.
[root@localhost ~]# head data.txt
Hello This is Group 5 from DA13 batch
Today is 09-Dec-2021 (Wed)
We learn new things in this class like:-
Big Data
Hadoop
Hive
HQL
MYSQL
Tableau
PYSPARK
[root@localhost ~]#
```

Head command for multiple files

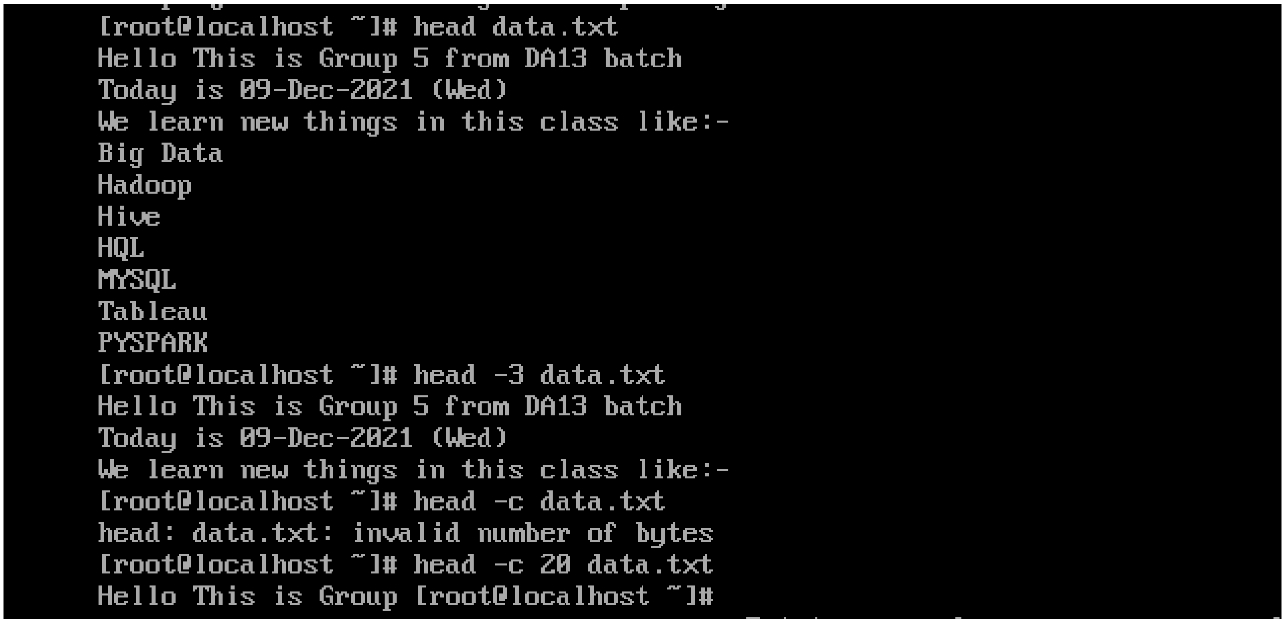
If we'll write two file names then it will display first ten lines (in this case file has five lines only) of each file separated by a heading.

Syntax: head <file name> <file name>

21) head – n option displays specified number of lines.

Syntax: head -n <file name>

22) head – c command counts the number of bytes of a file.
Syntax: head -c <number> <file name>

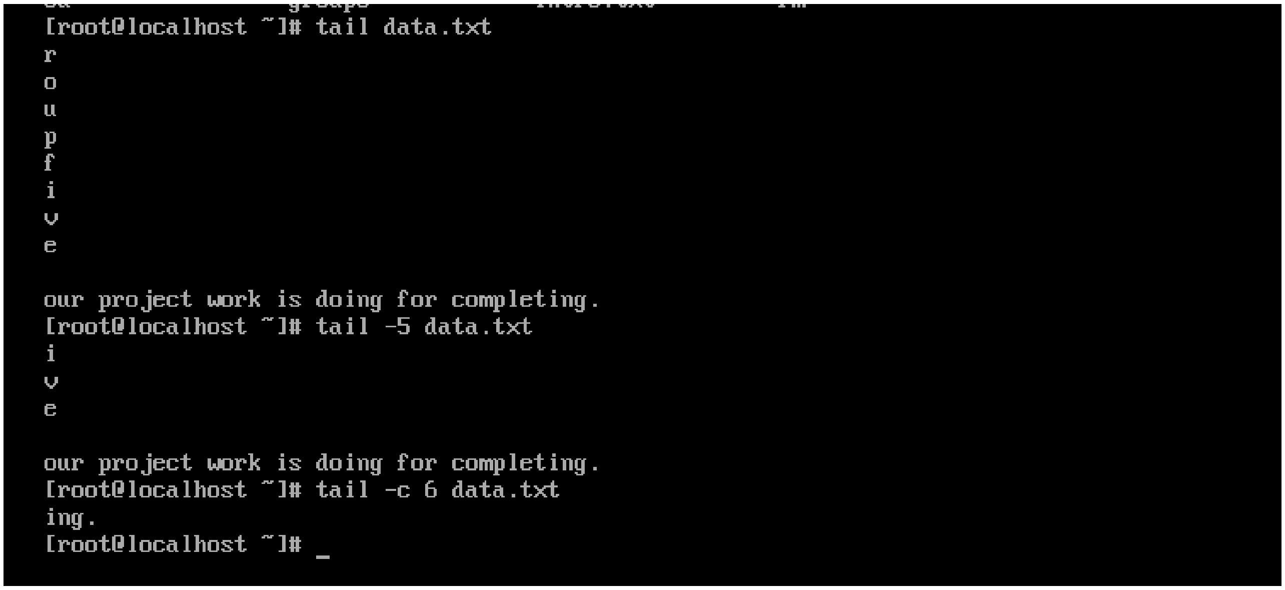


```
[root@localhost ~]# head data.txt
Hello This is Group 5 from DA13 batch
Today is 09-Dec-2021 (Wed)
We learn new things in this class like:-
Big Data
Hadoop
Hive
HQL
MYSQL
Tableau
PYSPARK
[root@localhost ~]# head -3 data.txt
Hello This is Group 5 from DA13 batch
Today is 09-Dec-2021 (Wed)
We learn new things in this class like:-
[root@localhost ~]# head -c data.txt
head: data.txt: invalid number of bytes
[root@localhost ~]# head -c 20 data.txt
Hello This is Group [root@localhost ~]#
```

23) tail- tail command is used to display the last ten lines of one or more files. Its main purpose is to read the error message. By default, it displays the last ten lines of a file.
Syntax: tail <file name>

23) tail -n option displays the specified number of lines.
Syntax: tail -n **<number> <file** name**>**
Example: tail -n 5 data.txt

24) tail -c option displays the specified number of bytes from the last.
Syntax: tail -c <number> <file name>
Example: tail -c 6 data.txt



```
[root@localhost ~]# tail data.txt
r
o
u
p
f
i
v
e

our project work is doing for completing.
[root@localhost ~]# tail -5 data.txt
i
v
e

our project work is doing for completing.
[root@localhost ~]# tail -c 6 data.txt
ing.
[root@localhost ~]# _
```

25) cat- command is used to read files & concatenate or combine multiple files together.

Syntax: cat filename

Example: cat data.txt

cat is also used to join multiple files into one single file using **">"**

Syntax: cat file1.txt file2.txt file3.txt > file-all.txt

cat command is used to create new file with below syntax.

Syntax: cat > new_file.txt

cat -n used to number all output lines of a file including empty lines.

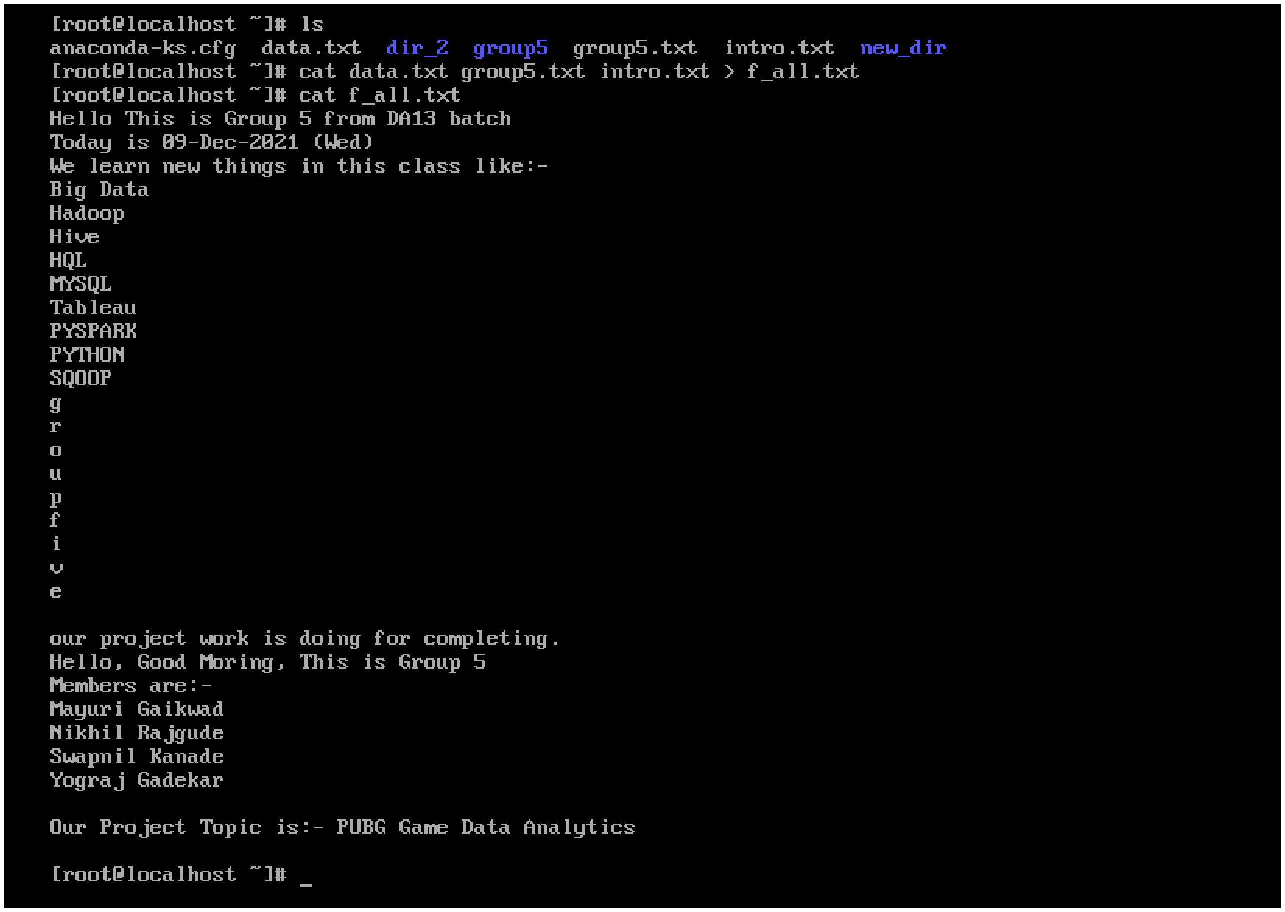
cat -n filename.txt

cat -b used to display the number of each non-empty line

cat -b filename

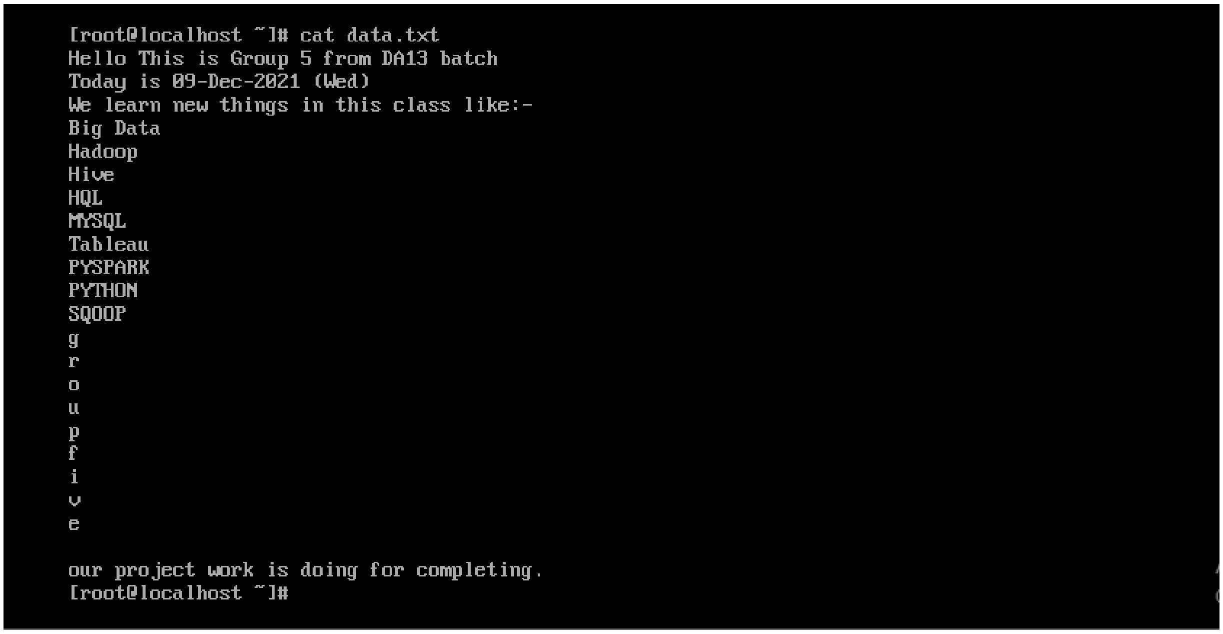
26) tac- The 'tac' command is the reverse of the 'cat' command. It is also known as 'cat' backward. It will display the file content in reverse order. It prints the last line first, then second last and so on. Such way, it prints the first line at last.

Syntax- tac **<file** name**>**



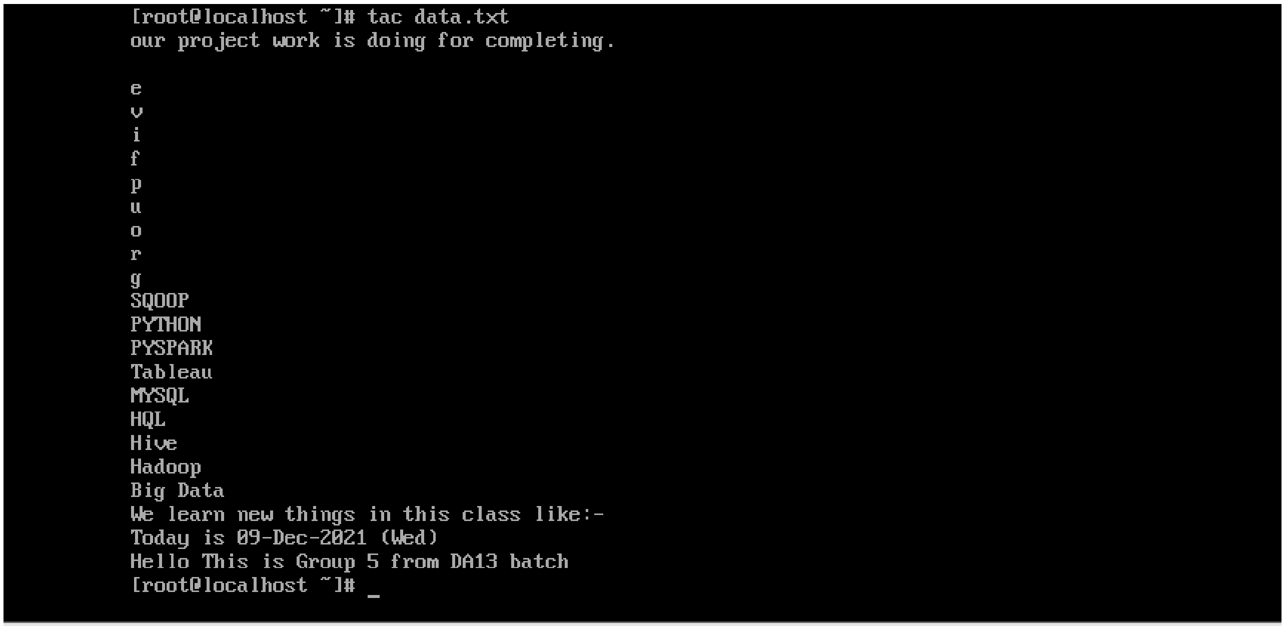
```
[root@localhost ~]# ls
anaconda-ks.cfg  data.txt  dir_2  group5  group5.txt  intro.txt  new_dir
[root@localhost ~]# cat data.txt group5.txt intro.txt > f_all.txt
[root@localhost ~]# cat f_all.txt
Hello This is Group 5 from DA13 batch
Today is 09-Dec-2021 (Wed)
We learn new things in this class like:-
Big Data
Hadoop
Hive
HQL
MYSQL
Tableau
PYSPARK
PYTHON
SQOOP
g
r
o
u
p
f
i
v
e

our project work is doing for completing.
Hello, Good Moring, This is Group 5
Members are:-
Mayuri Gaikwad
Nikhil Rajgude
Swapnil Kanade
Yograj Gadekar

Our Project Topic is:- PUBG Game Data Analytics

[root@localhost ~]# _
```

cat data.txt



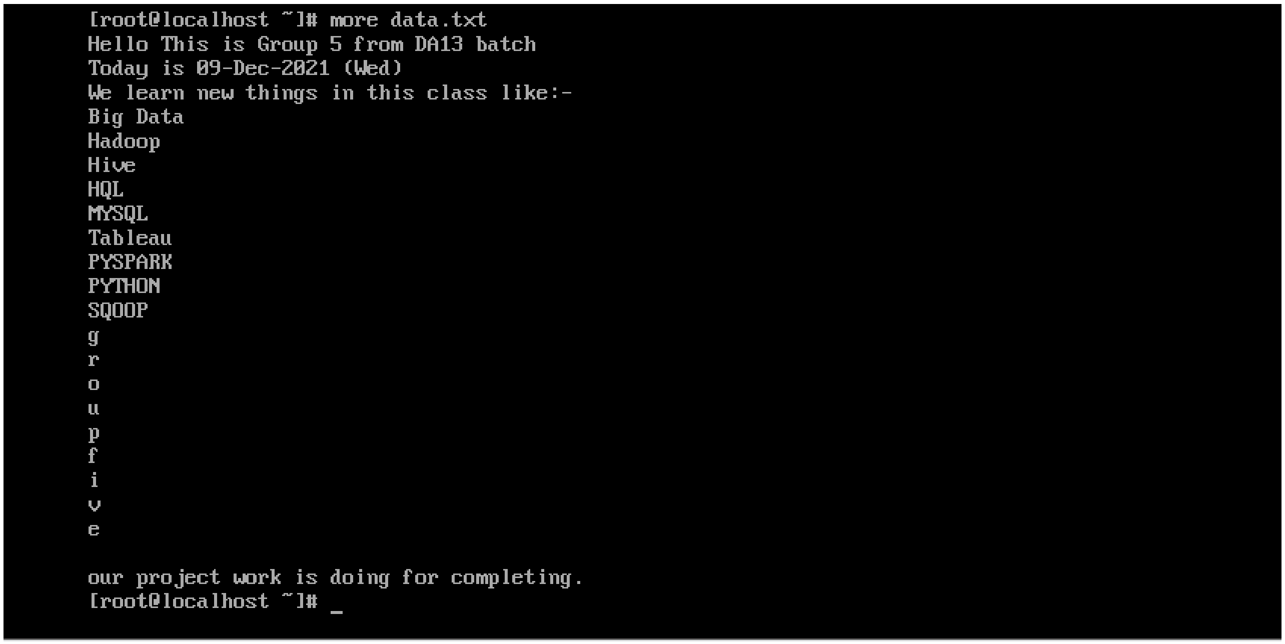
```
[root@localhost ~]# cat data.txt
Hello This is Group 5 from DA13 batch
Today is 09-Dec-2021 (Wed)
We learn new things in this class like:-
Big Data
Hadoop
Hive
HQL
MYSQL
Tableau
PYSPARK
PYTHON
SQOOP
g
r
o
u
p
f
i
v
e

our project work is doing for completing.
[root@localhost ~]#
```

tac data.txt



```
[root@localhost ~]# tac data.txt
our project work is doing for completing.

e
v
i
f
p
u
o
r
g
SQOOP
PYTHON
PYSPARK
Tableau
MYSQL
HQL
Hive
Hadoop
Big Data
We learn new things in this class like:-
Today is 09-Dec-2021 (Wed)
Hello This is Group 5 from DA13 batch
[root@localhost ~]# _
```

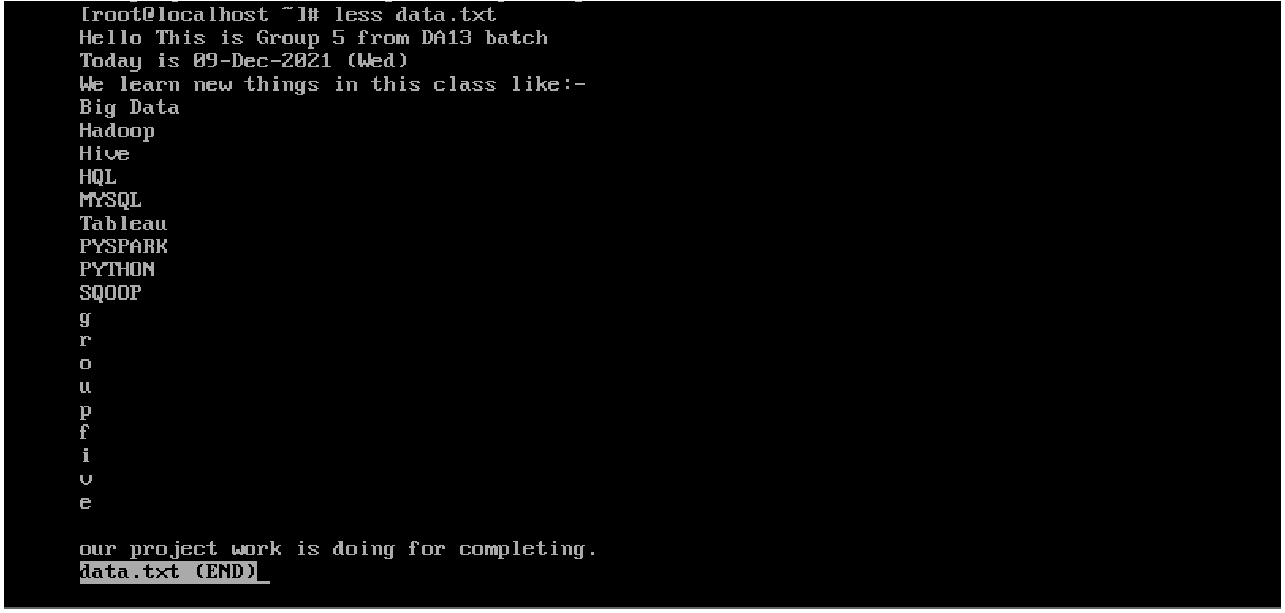
27) more- Similar to cat command also displays the content of a file.
Only difference is that, in case of larger files, 'cat' command output will scroll off your screen while 'more' command displays output one screenful at a time.
Syntax: more <file name>



```
[root@localhost ~]# more data.txt
Hello This is Group 5 from DA13 batch
Today is 09-Dec-2021 (Wed)
We learn new things in this class like:-
Big Data
Hadoop
Hive
HQL
MYSQL
Tableau
PYSPARK
PYTHON
SQOOP
g
r
o
u
p
f
i
v
e

our project work is doing for completing.
[root@localhost ~]# _
```

28) less- The 'less' command is same as 'more' command but include some more features.
It automatically adjust with the width and height of the teminal window, while 'more' command cuts the content as the width of the terminal window get shorter.

Syntax: less <file name>

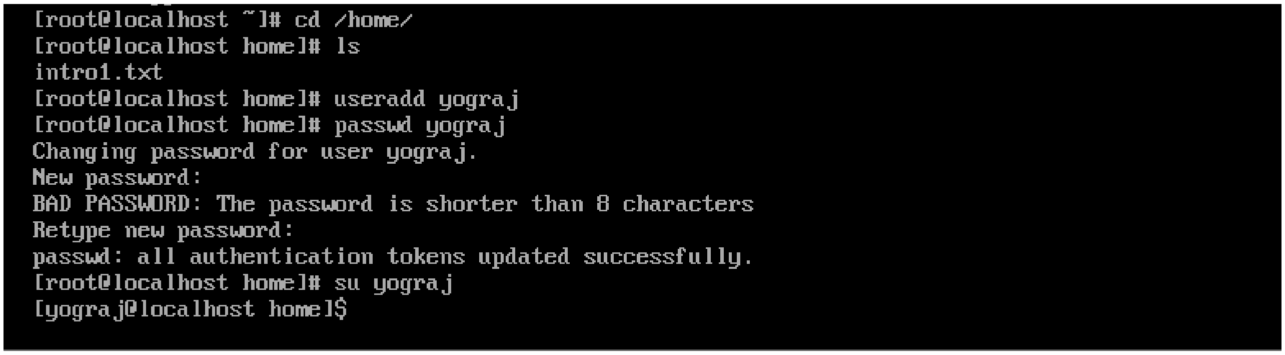


```
[root@localhost ~]# less data.txt
Hello This is Group 5 from DA13 batch
Today is 09-Dec-2021 (Wed)
We learn new things in this class like:-
Big Data
Hadoop
Hive
HQL
MYSQL
Tableau
PYSPARK
PYTHON
SQOOP
g
r
o
u
p
f
i
v
e

our project work is doing for completing.
data.txt (END)
```

29) su- The su command allows you to run a shell as another user.
Syntax: su <username>

30) useradd- Used to add or remove a user on a linux server
Syntax: useradd username

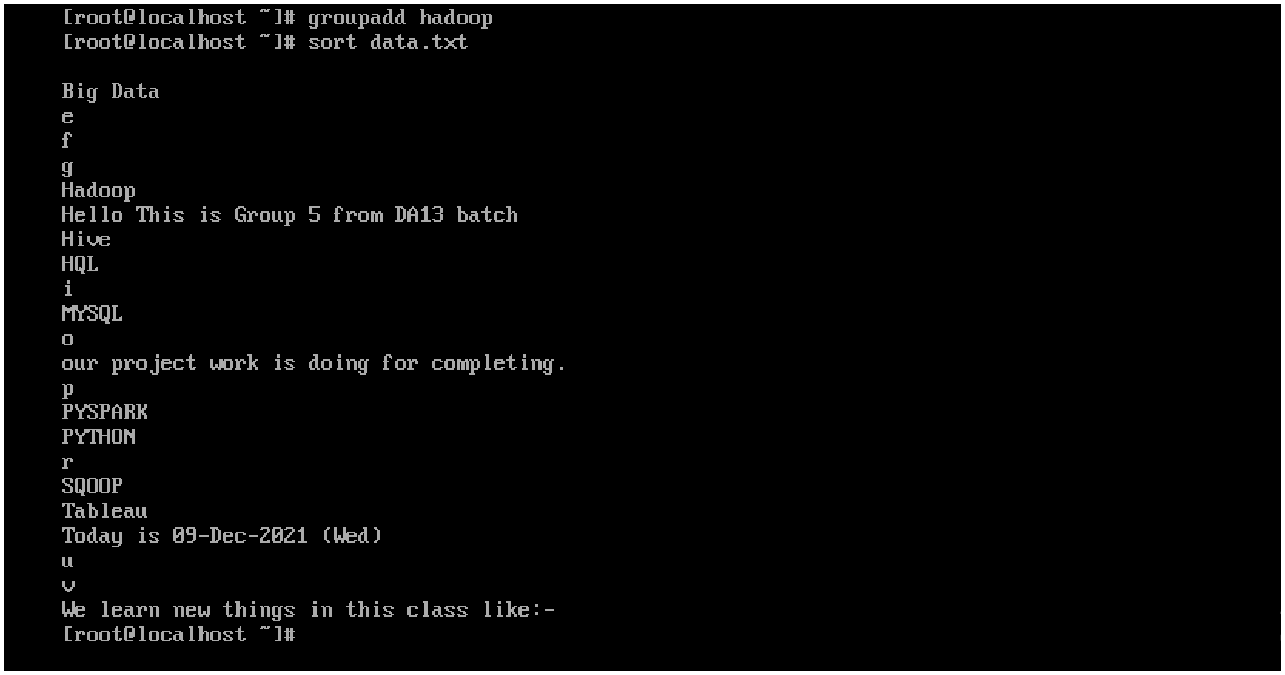
31) passwd- A user can set the password with the command **passwd**
Syntax: passwd



```
[root@localhost ~]# cd /home/
[root@localhost home]# ls
intro1.txt
[root@localhost home]# useradd yograj
[root@localhost home]# passwd yograj
Changing password for user yograj.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost home]# su yograj
[yograj@localhost home]$
```

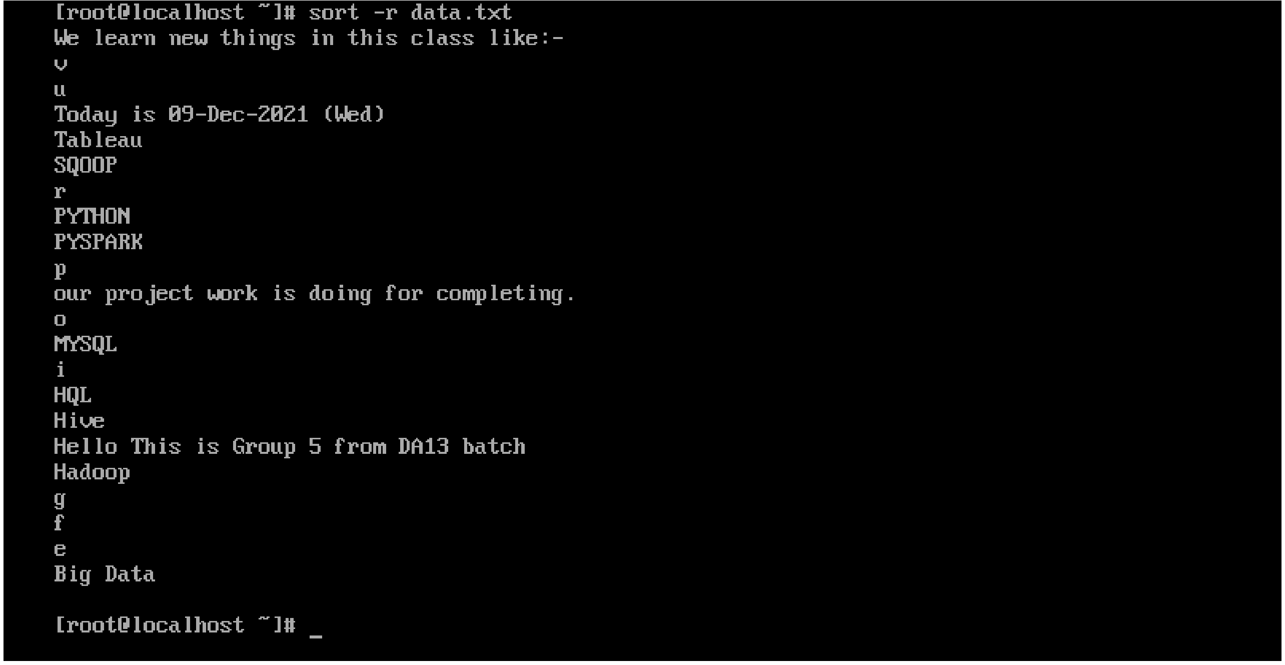
32) groupadd- The groupadd command creates or add a group in our system.
Syntax: groupadd <groupName>

33) sort - The 'sort' command sorts the file content in an alphabetical order.
Syntax: sort <fileName>
Example: sort data.txt



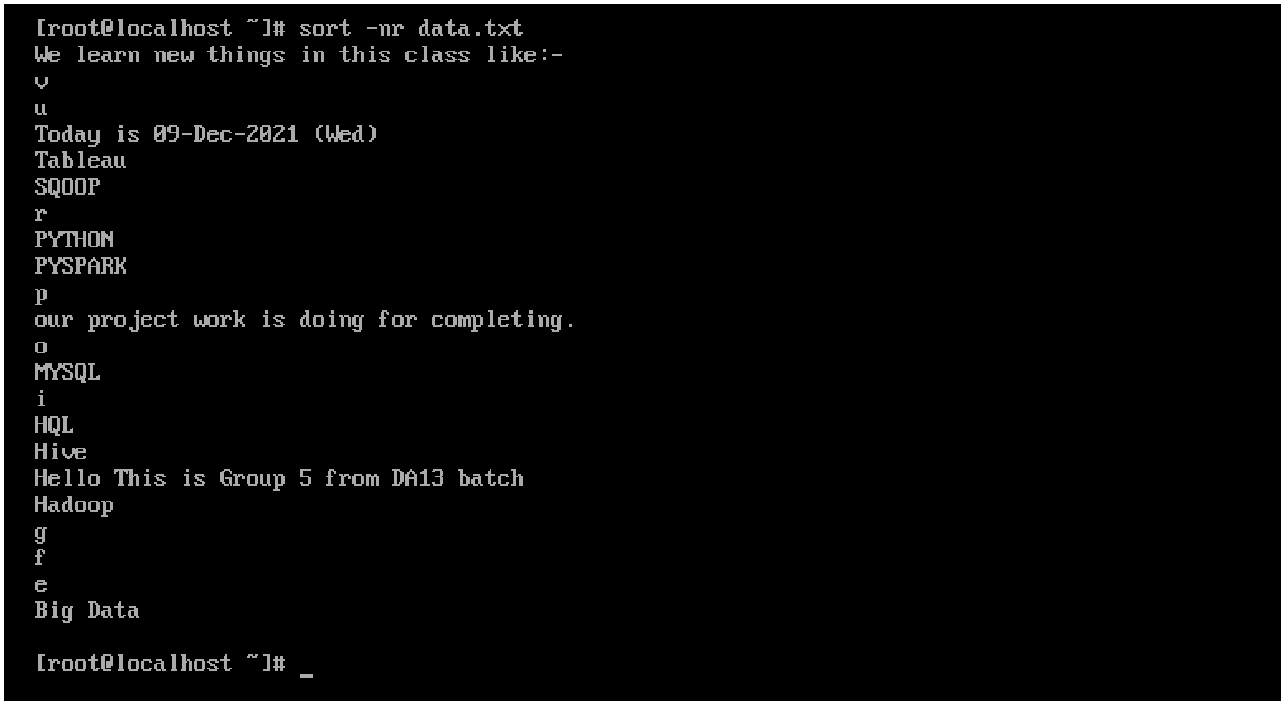
```
[root@localhost ~]# groupadd hadoop
[root@localhost ~]# sort data.txt

Big Data
e
f
g
Hadoop
Hello This is Group 5 from DA13 batch
Hive
HQL
i
MYSQL
o
our project work is doing for completing.
p
PYSPARK
PYTHON
r
SQOOP
Tableau
Today is 09-Dec-2021 (Wed)
u
v
We learn new things in this class like:-
[root@localhost ~]#
```

sort -r option for sorting in reverse order



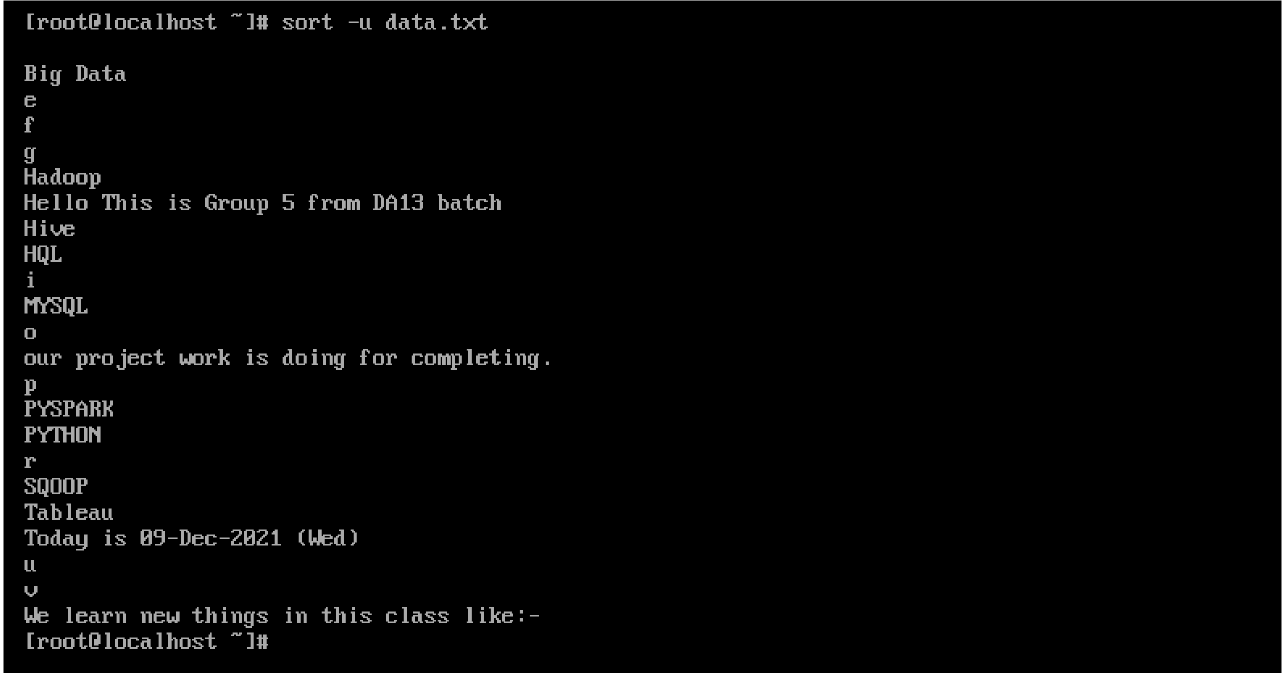
```
[root@localhost ~]# sort -r data.txt
We learn new things in this class like:-
v
u
Today is 09-Dec-2021 (Wed)
Tableau
SQOOP
r
PYTHON
PYSPARK
p
our project work is doing for completing.
o
MYSQL
i
HQL
Hive
Hello This is Group 5 from DA13 batch
Hadoop
g
f
e
Big Data

[root@localhost ~]# _
```

sort -nr option for numeric data in reverse order



```
[root@localhost ~]# sort -nr data.txt
We learn new things in this class like:-
v
u
Today is 09-Dec-2021 (Wed)
Tableau
SQOOP
r
PYTHON
PYSPARK
p
our project work is doing for completing.
o
MYSQL
i
HQL
Hive
Hello This is Group 5 from DA13 batch
Hadoop
g
f
e
Big Data

[root@localhost ~]# _
```

sort -u option to sort & remove duplicates



```
[root@localhost ~]# sort -u data.txt

Big Data
e
f
g
Hadoop
Hello This is Group 5 from DA13 batch
Hive
HQL
i
MYSQL
o
our project work is doing for completing.
p
PYSPARK
PYTHON
r
SQOOP
Tableau
Today is 09-Dec-2021 (Wed)
u
v
We learn new things in this class like:-
[root@localhost ~]#
```

sort -k option to find particular column

sort -m option for sort month name in month ascending order

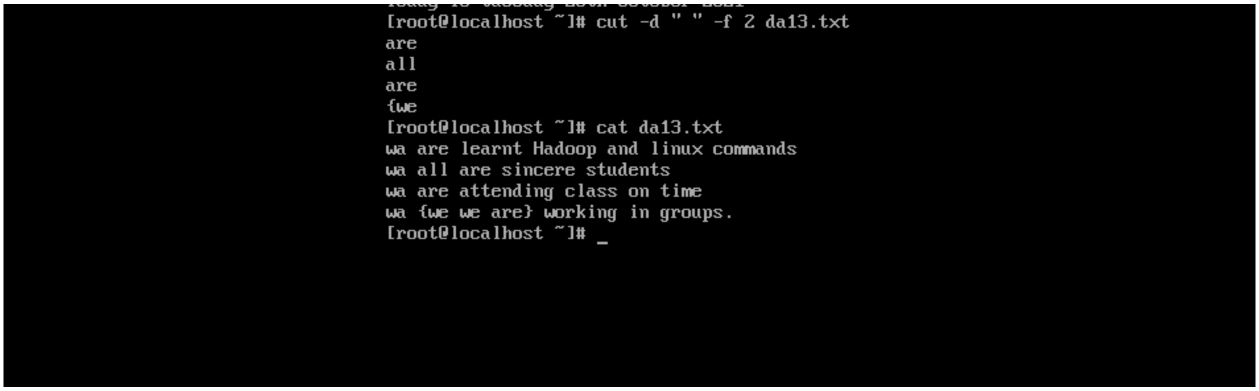
34) cut – used to cutting out the section from specified files

Syntax: cut OPTION... [FILE]..

Example: cut -d " " -f 2 da13.txt

cut -d It is used to cut a specific section by a delimiter.

cut -f It is used to select the specific fields.



```
[root@localhost ~]# cut -d " " -f 2 da13.txt
are
all
are
{we
[root@localhost ~]# cat da13.txt
wa are learnt Hadoop and linux commands
wa all are sincere students
wa are attending class on time
wa {we we are} working in groups.
[root@localhost ~]# _
```

35) grep- The 'grep' command stands for **"global regular expression print"**. grep command filters the content of a file which makes our search easy.

Syntax: grep <searchword>

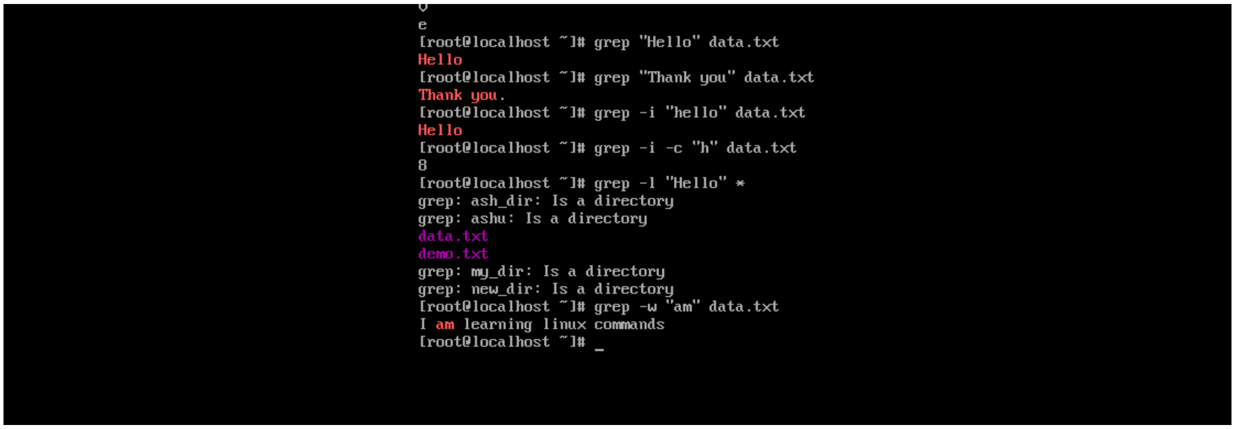
Example: grep "Hello" demo.txt

grep -i The 'grep -i' command filters output in a case-insensitive way.

grep -c -grep -C command is used to display the **line after and line before** the result.

grep -l find similar kind of names

grep -w checking for the whole word in the file



```
e
[root@localhost ~]# grep "Hello" data.txt
Hello
[root@localhost ~]# grep "Thank you" data.txt
Thank you.
[root@localhost ~]# grep -i "hello" data.txt
Hello
[root@localhost ~]# grep -i -c "h" data.txt
8
[root@localhost ~]# grep -l "Hello" *
grep: ash_dir: Is a directory
grep: ashu: Is a directory
data.txt
demo.txt
grep: my_dir: Is a directory
grep: new_dir: Is a directory
[root@localhost ~]# grep -w "am" data.txt
I am learning linux commands
[root@localhost ~]# _
```

36) comm - The 'comm' command compares two files.

By default, 'comm' will always display **three columns**.

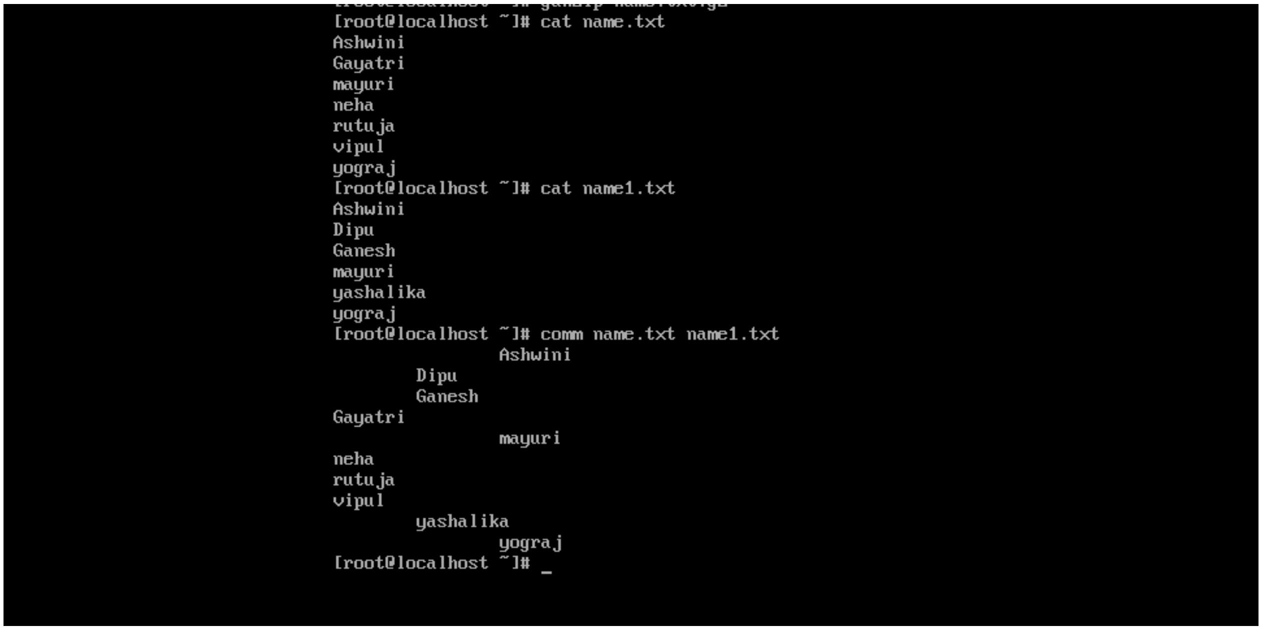
First column indicates non-matching items of first file,

second column indicates non-matching items of second file, and

third column indicates matching items of both the files.

Both the files has to be in sorted order for 'comm' command to be executed.

Syntax: comm <file1> <file2>
Example: comm name.txt name1.txt



```
[root@localhost ~]# gunzip name.txt.gz
[root@localhost ~]# cat name.txt
Ashwini
Gayatri
mayuri
neha
rutuja
vipul
yograj
[root@localhost ~]# cat name1.txt
Ashwini
Dipu
Ganesh
mayuri
yashalika
yograj
[root@localhost ~]# comm name.txt name1.txt
                Ashwini
        Dipu
        Ganesh
Gayatri
                mayuri
neha
rutuja
vipul
        yashalika
                yograj
[root@localhost ~]# _
```

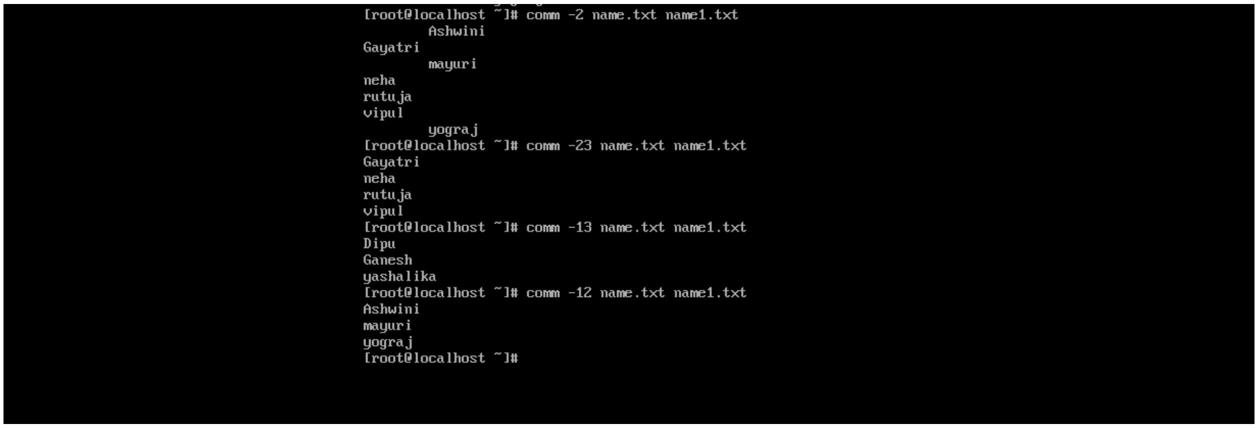
To display single column

Syntax:

comm -23 (To display first column)

comm -13 (To display second column)

comm -12 (To display third column)

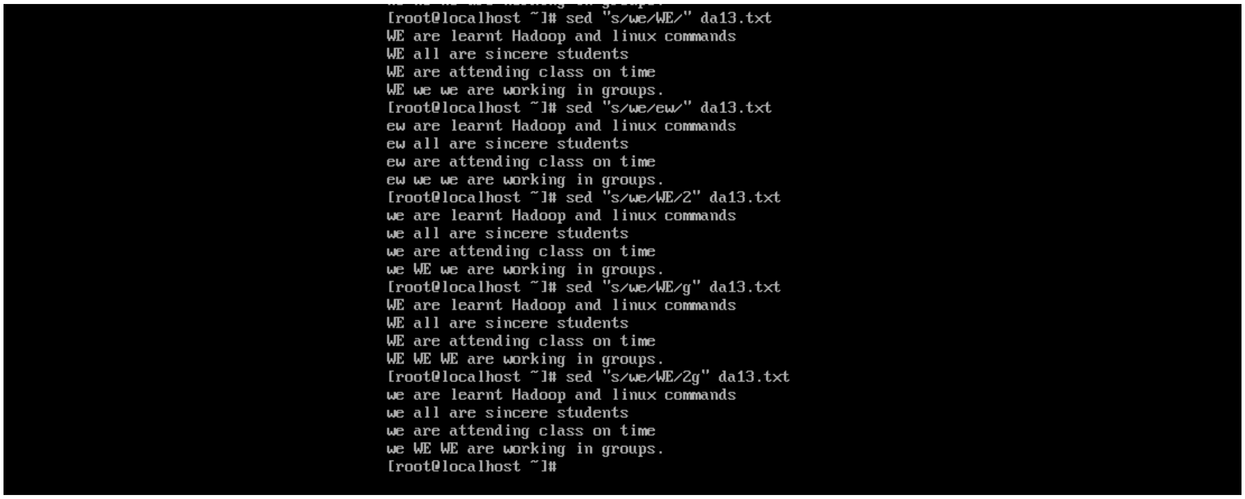


```
[root@localhost ~]# comm -2 name.txt name1.txt
        Ashwini
Gayatri
        mayuri
neha
rutuja
vipul
        yograj
[root@localhost ~]# comm -23 name.txt name1.txt
Gayatri
neha
rutuja
vipul
[root@localhost ~]# comm -13 name.txt name1.txt
Dipu
Ganesh
yashalika
[root@localhost ~]# comm -12 name.txt name1.txt
Ashwini
mayuri
yograj
[root@localhost ~]#
```

37) sed - sed command stands for stream editor. It is used to edit files using regular expressions. But this editing is not permanent. It remains only in display, but in actual, file content remains the same.

Syntax: sed [OPTION]... {script-only-if-no-other-script} [input-file]...
Example: sed “s/we/ew/” da13.txt
Sed “s/we/WE/” da13.txt
Sed “s.we/WE/2” da13.txt

Sed -i for caseinsensitive
g – used to global replacement
-n used to display only replaced line
-p for print
d used for deleting
.bak (for backup)
-i.bak



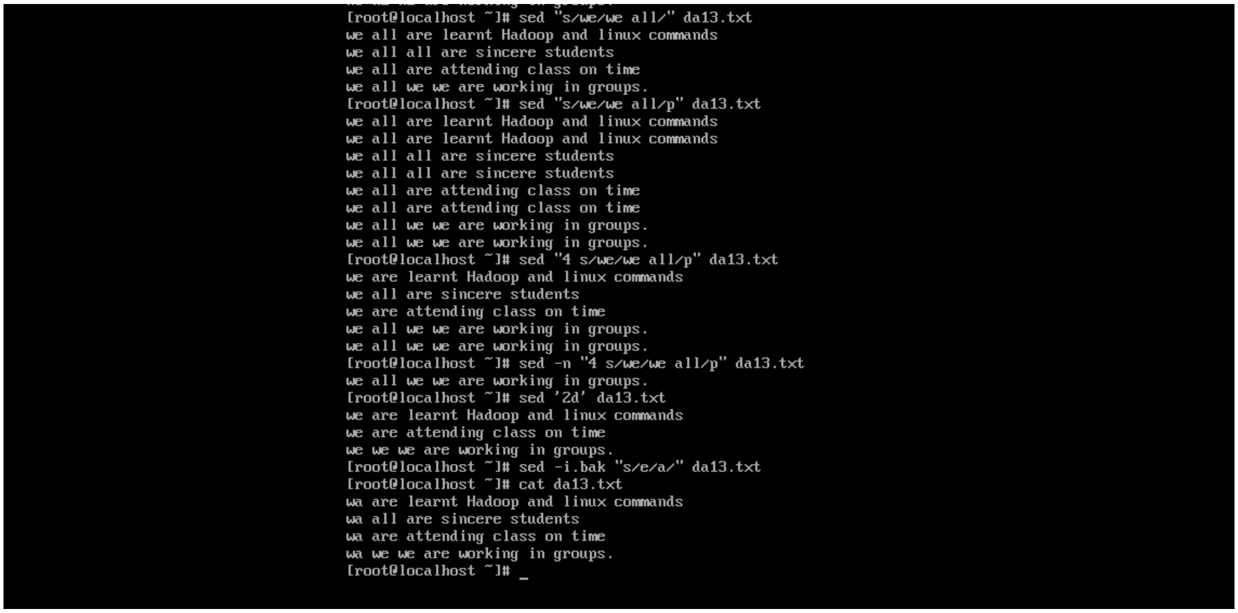
```
[root@localhost ~]# sed "s/we/WE/" da13.txt
WE are learnt Hadoop and linux commands
WE all are sincere students
WE are attending class on time
WE we we are working in groups.
[root@localhost ~]# sed "s/we/ew/" da13.txt
ew are learnt Hadoop and linux commands
ew all are sincere students
ew are attending class on time
ew we we are working in groups.
[root@localhost ~]# sed "s/we/WE/2" da13.txt
we are learnt Hadoop and linux commands
we all are sincere students
we are attending class on time
we WE we are working in groups.
[root@localhost ~]# sed "s/we/WE/g" da13.txt
WE are learnt Hadoop and linux commands
WE all are sincere students
WE are attending class on time
WE WE WE are working in groups.
[root@localhost ~]# sed "s/we/WE/2g" da13.txt
we are learnt Hadoop and linux commands
we all are sincere students
we are attending class on time
we WE WE are working in groups.
[root@localhost ~]#
```

sed “s/we/we all/” da13.txt (replacing a particular word)

sed “s/we/we all/p” da13.txt (print the line)

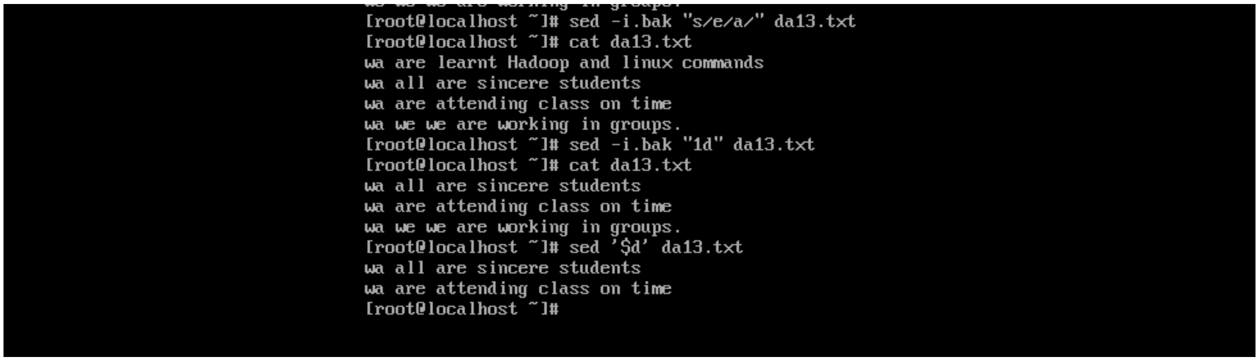
sed “4 s/we/we all/p” da13.txt (print 4th line)

sed -n “4 s/we/we all/p” da13.txt (display only replaced line)



```
[root@localhost ~]# sed "s/we/we all/" da13.txt
we all are learnt Hadoop and linux commands
we all all are sincere students
we all are attending class on time
we all we we are working in groups.
[root@localhost ~]# sed "s/we/we all/p" da13.txt
we all are learnt Hadoop and linux commands
we all are learnt Hadoop and linux commands
we all all are sincere students
we all all are sincere students
we all are attending class on time
we all are attending class on time
we all we we are working in groups.
we all we we are working in groups.
[root@localhost ~]# sed "4 s/we/we all/p" da13.txt
we are learnt Hadoop and linux commands
we all are sincere students
we are attending class on time
we all we we are working in groups.
we all we we are working in groups.
[root@localhost ~]# sed -n "4 s/we/we all/p" da13.txt
we all we we are working in groups.
[root@localhost ~]# sed '2d' da13.txt
we are learnt Hadoop and linux commands
we are attending class on time
we we we are working in groups.
[root@localhost ~]# sed -i.bak "s/e/a/" da13.txt
[root@localhost ~]# cat da13.txt
wa are learnt Hadoop and linux commands
wa all are sincere students
wa are attending class on time
wa we we are working in groups.
[root@localhost ~]# _
```

sed -i.bak “s/e/a/” da13.txt (replace e with a)
sed -i.bak “1d” da13.txt (delete 1st line)
sed -i.bak ‘$d’ da13.txt (deleting last line)



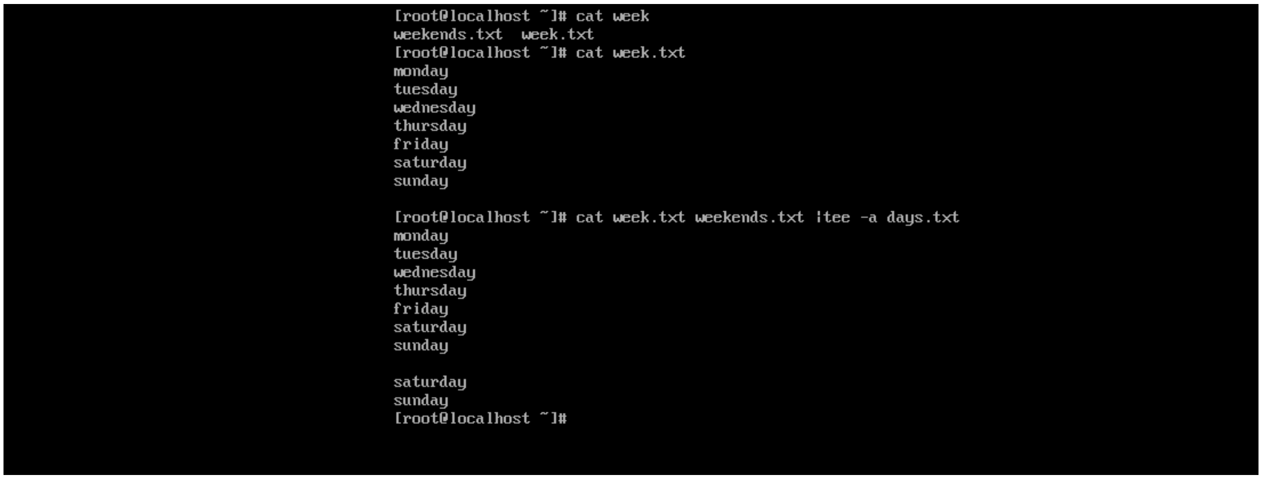
```
we we we are working in groups.
[root@localhost ~]# sed -i.bak "s/e/a/" da13.txt
[root@localhost ~]# cat da13.txt
wa are learnt Hadoop and linux commands
wa all are sincere students
wa are attending class on time
wa we we are working in groups.
[root@localhost ~]# sed -i.bak "1d" da13.txt
[root@localhost ~]# cat da13.txt
wa all are sincere students
wa are attending class on time
wa we we are working in groups.
[root@localhost ~]# sed '$d' da13.txt
wa all are sincere students
wa are attending class on time
[root@localhost ~]#
```

38) tee- tee command is quite similar to the 'cat' command, with only one difference. It puts stdin on stdout and also put them into a file.
It is one of the most used commands with other commands through piping.

Syntax- tee <options> <file name>

-a, --append: It is used to append the data to the given files, it does not overwrite data.

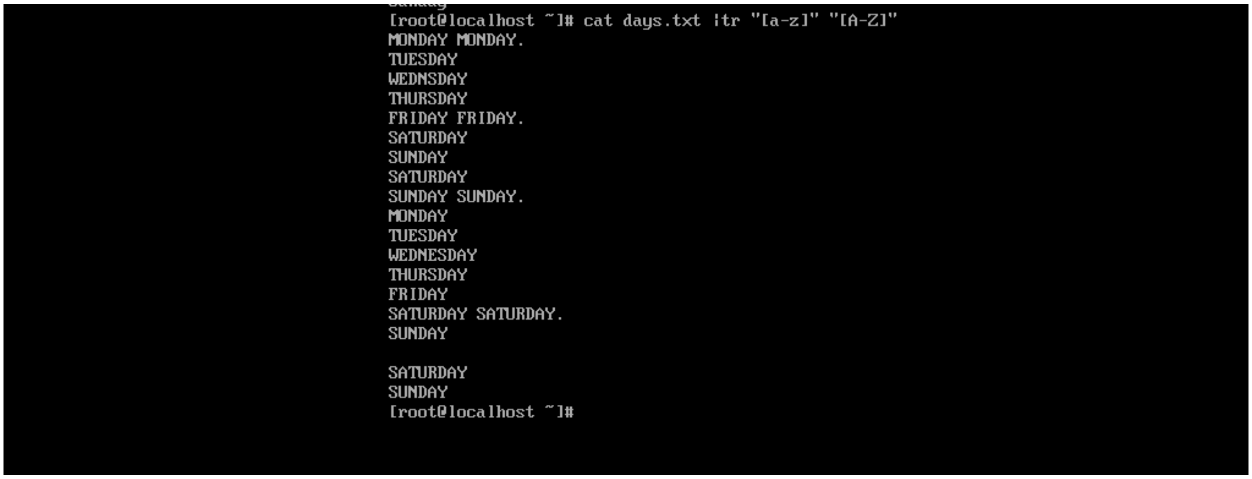
Example: cat name.txt |tee -a name1.txt
cat week.txt weekend.txt |tee -a day.txt



```
[root@localhost ~]# cat week
weekends.txt  week.txt
[root@localhost ~]# cat week.txt
monday
tuesday
wednesday
thursday
friday
saturday
sunday

[root@localhost ~]# cat week.txt weekends.txt |tee -a days.txt
monday
tuesday
wednesday
thursday
friday
saturday
sunday

saturday
sunday
[root@localhost ~]#
```

39) tr – The command 'tr' stands for **'translate'**. It is used to translate, like from lowercase to uppercase and vice versa or new lines into spaces.

Syntax: command | tr <old> <new>

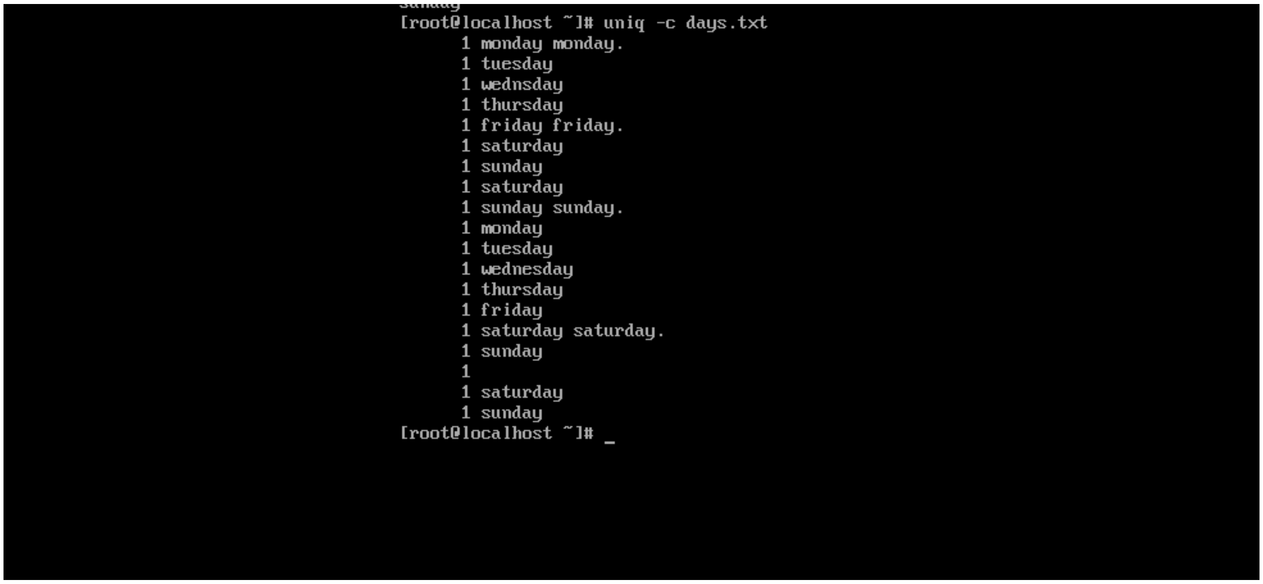


```
sunday
[root@localhost ~]# cat days.txt |tr "[a-z]" "[A-Z]"
MONDAY MONDAY.
TUESDAY
WEDNSDAY
THURSDAY
FRIDAY FRIDAY.
SATURDAY
SUNDAY
SATURDAY
SUNDAY SUNDAY.
MONDAY
TUESDAY
WEDNESDAY
THURSDAY
FRIDAY
SATURDAY SATURDAY.
SUNDAY

SATURDAY
SUNDAY
[root@localhost ~]#
```

40) uniq - uniq command is used to remove all the repeated lines from a file.

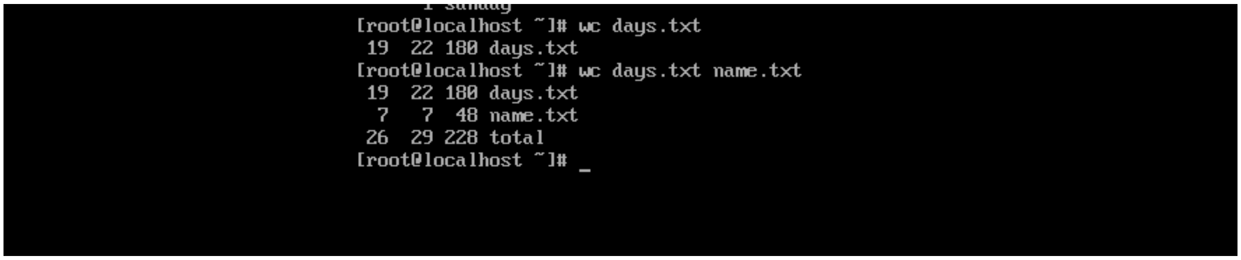
Syntax: uniq [OPTION]... [INPUT]

Example: uniq -c days.txt   (count)
uniq -d days.txt   (show only repeated option
uniq -D days.txt   (print repeated lines)
uniq -u days.txt   (only unique)



```
sunday
[root@localhost ~]# uniq -c days.txt
      1 monday monday.
      1 tuesday
      1 wednsday
      1 thursday
      1 friday friday.
      1 saturday
      1 sunday
      1 saturday
      1 sunday sunday.
      1 monday
      1 tuesday
      1 wednesday
      1 thursday
      1 friday
      1 saturday saturday.
      1 sunday
      1
      1 saturday
      1 sunday
[root@localhost ~]# _
```

41) wc – used to count the line,words,& characters in a file.
Syntax: wc filename



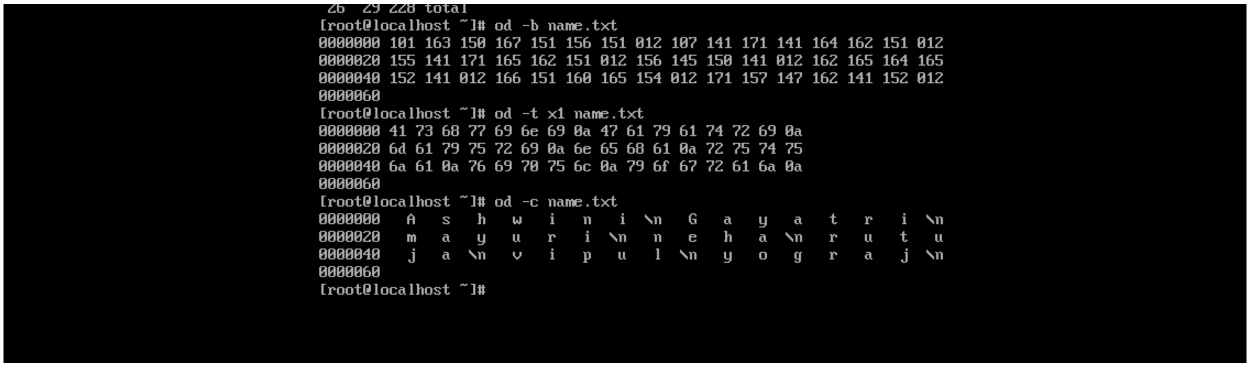
```
      1 sunday
[root@localhost ~]# wc days.txt
 19  22 180 days.txt
[root@localhost ~]# wc days.txt name.txt
 19  22 180 days.txt
  7   7  48 name.txt
 26  29 228 total
[root@localhost ~]# _
```

42) od – It displays content of a file in different human-readable formats like hexadecimal, octal and ASCII characters.

Syntax: od -b <filename>

od -t x1 <filename>

od -c <filename>



```
 26  29 228 total
[root@localhost ~]# od -b name.txt
0000000 101 163 150 167 151 156 151 012 107 141 171 141 164 162 151 012
0000020 155 141 171 165 162 151 012 156 145 150 141 012 162 165 164 165
0000040 152 141 012 166 151 160 165 154 012 171 157 147 162 141 152 012
0000060
[root@localhost ~]# od -t x1 name.txt
0000000 41 73 68 77 69 6e 69 0a 47 61 79 61 74 72 69 0a
0000020 6d 61 79 75 72 69 0a 6e 65 68 61 0a 72 75 74 75
0000040 6a 61 0a 76 69 70 75 6c 0a 79 6f 67 72 61 6a 0a
0000060
[root@localhost ~]# od -c name.txt
0000000   A   s   h   w   i   n   i  \n   G   a   y   a   t   r   i  \n
0000020   m   a   y   u   r   i  \n   n   e   h   a  \n   r   u   t   u
0000040   j   a  \n   v   i   p   u   l  \n   y   o   g   r   a   j  \n
0000060
[root@localhost ~]#
```

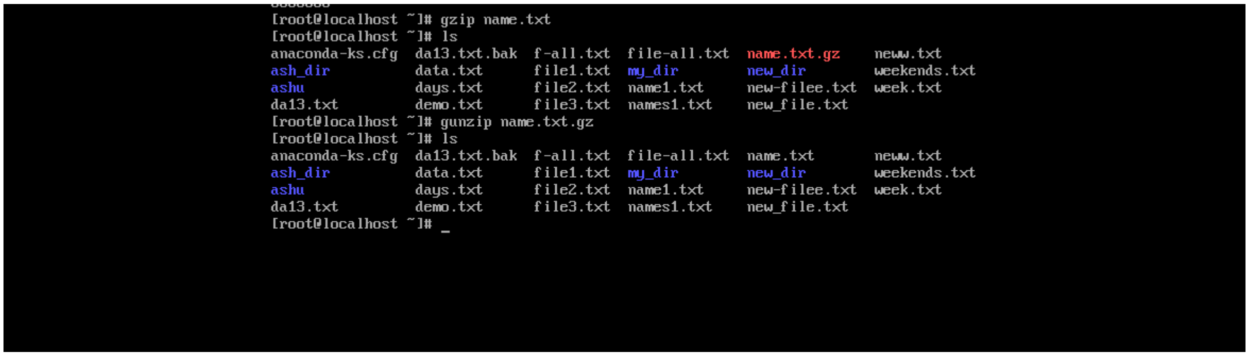
43) gzip – used to truncate the file size. It is compressing tool.

  By default original file will be replaced by the compressed file ending with extension (.gz).

Syntax: gzip filename

44) gunzip – used to decompressed the file.

 Syntax: gunzip filename



```
[root@localhost ~]# gzip name.txt
[root@localhost ~]# ls
anaconda-ks.cfg  da13.txt.bak  f-all.txt  file-all.txt  name.txt.gz    neww.txt
ash_dir          data.txt      file1.txt  my_dir        new_dir        weekends.txt
ashu             days.txt      file2.txt  name1.txt     new-filee.txt  week.txt
da13.txt         demo.txt      file3.txt  names1.txt    new_file.txt
[root@localhost ~]# gunzip name.txt.gz
[root@localhost ~]# ls
anaconda-ks.cfg  da13.txt.bak  f-all.txt  file-all.txt  name.txt       neww.txt
ash_dir          data.txt      file1.txt  my_dir        new_dir        weekends.txt
ashu             days.txt      file2.txt  name1.txt     new-filee.txt  week.txt
da13.txt         demo.txt      file3.txt  names1.txt    new_file.txt
[root@localhost ~]# _
```

45) ls/etc

White color- regular/ normal file

Blue color- directory

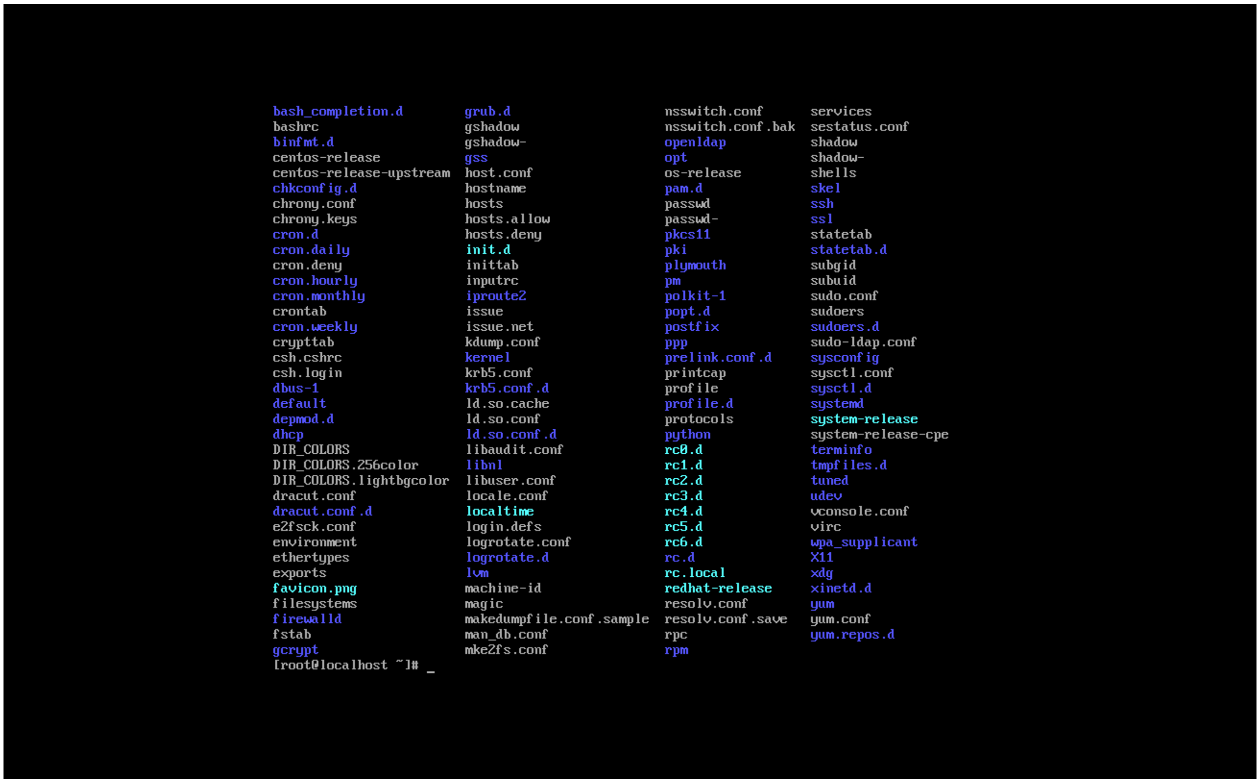
Brighte green – executable file

Bright red – archieve file / compressed file

Magenta – image file

Cyan – audio file

Sky blue – symbolic link file



```
bash_completion.d        grub.d                    nsswitch.conf      services
bashrc                   gshadow                   nsswitch.conf.bak  sestatus.conf
binfmt.d                 gshadow-                  openldap           shadow
centos-release           gss                       opt                shadow-
centos-release-upstream  host.conf                 os-release         shells
chkconfig.d              hostname                  pam.d              skel
chrony.conf              hosts                     passwd             ssh
chrony.keys              hosts.allow               passwd-            ssl
cron.d                   hosts.deny                pkcs11             statetab
cron.daily               init.d                    pki                statetab.d
cron.deny                inittab                   plymouth           subgid
cron.hourly              inputrc                   pm                 subuid
cron.monthly             iproute2                  polkit-1           sudo.conf
crontab                  issue                     popt.d             sudoers
cron.weekly              issue.net                 postfix            sudoers.d
crypttab                 kdump.conf                ppp                sudo-ldap.conf
csh.cshrc                kernel                    prelink.conf.d     sysconfig
csh.login                krb5.conf                 printcap           sysctl.conf
dbus-1                   krb5.conf.d               profile            sysctl.d
default                  ld.so.cache               profile.d          systemd
depmod.d                 ld.so.conf                protocols          system-release
dhcp                     ld.so.conf.d              python             system-release-cpe
DIR_COLORS               libaudit.conf             rc0.d              terminfo
DIR_COLORS.256color      libnl                     rc1.d              tmpfiles.d
DIR_COLORS.lightbgcolor  libuser.conf              rc2.d              tuned
dracut.conf              locale.conf               rc3.d              udev
dracut.conf.d            localtime                 rc4.d              vconsole.conf
e2fsck.conf              login.defs                rc5.d              virc
environment              logrotate.conf            rc6.d              wpa_supplicant
ethertypes               logrotate.d               rc.d               X11
exports                  lvm                       rc.local           xdg
favicon.png              machine-id                redhat-release     xinetd.d
filesystems              magic                     resolv.conf        yum
firewalld                makedumpfile.conf.sample  resolv.conf.save   yum.conf
fstab                    man_db.conf               rpc                yum.repos.d
gcrypt                   mke2fs.conf               rpm
[root@localhost ~]# _
```

46) find – used to find particular file within a directory. It supports various options to find a file such as by name, type, date & more.

Symbols –

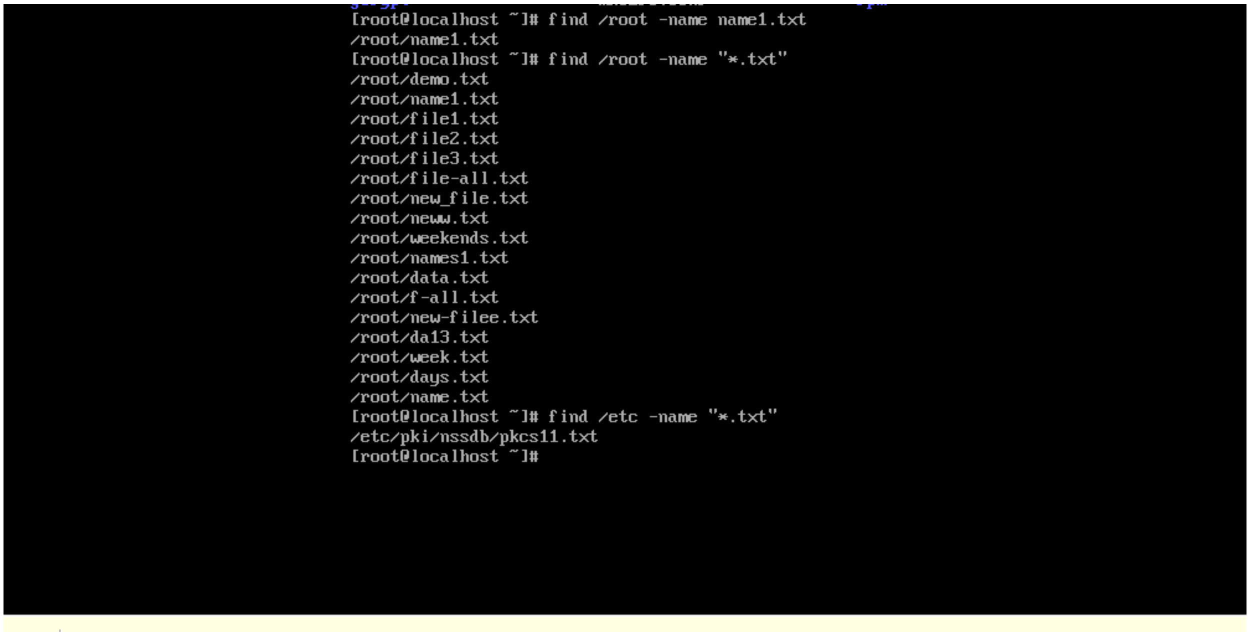
(.) for current directory name

/   for root

Example: find /root -name names.txt

find /root -name “*.txt”

find /etc -name “*.txt”



```
[root@localhost ~]# find /root -name name1.txt
/root/name1.txt
[root@localhost ~]# find /root -name "*.txt"
/root/demo.txt
/root/name1.txt
/root/file1.txt
/root/file2.txt
/root/file3.txt
/root/file-all.txt
/root/new_file.txt
/root/neww.txt
/root/weekends.txt
/root/names1.txt
/root/data.txt
/root/f-all.txt
/root/new-filee.txt
/root/da13.txt
/root/week.txt
/root/days.txt
/root/name.txt
[root@localhost ~]# find /etc -name "*.txt"
/etc/pki/nssdb/pkcs11.txt
[root@localhost ~]#
```

47) locate – used to search a file by filename. Similar to ‘find’ command

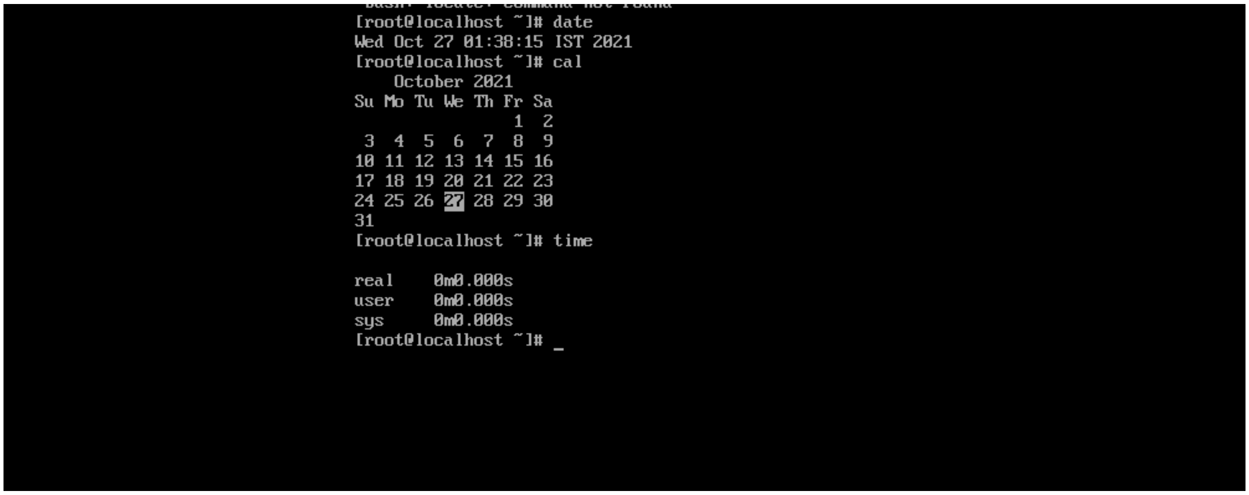
Syntax: locate <filename>

48) date – used to display date time and timezone

Syntax: date

49) cal – used to display the current month calendar with the current date highlighted

50) time- used to display and set the current system time.

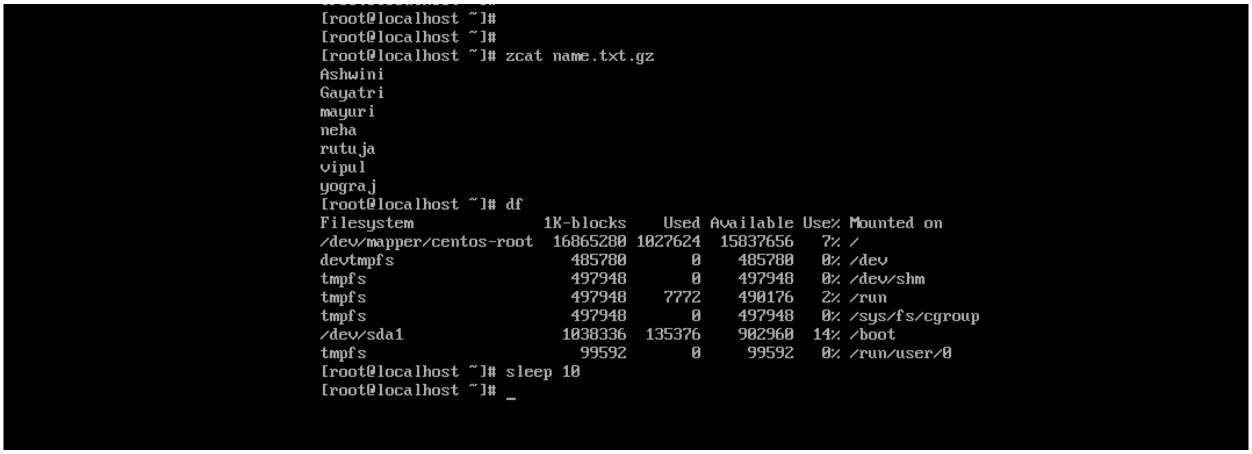


```
-bash: locate: command not found
[root@localhost ~]# date
Wed Oct 27 01:38:15 IST 2021
[root@localhost ~]# cal
    October 2021
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
[root@localhost ~]# time

real    0m0.000s
user    0m0.000s
sys     0m0.000s
[root@localhost ~]# _
```

51) zcat – used to display compressed file**.**
Syntax: zcat filename

52) df – used to display the disk space
Syntax: df

53) sleep –
Syntax- sleep <time>



```
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# zcat name.txt.gz
Ashwini
Gayatri
mayuri
neha
rutuja
vipul
yograj
[root@localhost ~]# df
Filesystem              1K-blocks    Used Available Use% Mounted on
/dev/mapper/centos-root  16865280 1027624  15837656   7% /
devtmpfs                   485780       0    485780   0% /dev
tmpfs                      497948       0    497948   0% /dev/shm
tmpfs                      497948    7772    490176   2% /run
tmpfs                      497948       0    497948   0% /sys/fs/cgroup
/dev/sda1                 1038336  135376    902960  14% /boot
tmpfs                       99592       0     99592   0% /run/user/0
[root@localhost ~]# sleep 10
[root@localhost ~]# _
```

54) mount- used to connect an external device file system to the systems file system.
Syntax- mount -t type <device> <directory>

55) clear- used to clear the terminal screen.
Syntax- clear

56) exit- used to exit the shell where it is currently running.
Syntax- exit