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

1. **Steps Performed in this activity?**

Follow step by step commands to performed this activity

1. **What resources / materials / equipment / tools did you use for this activity?**

VMware, MS word

1. **What skills did you acquire?**

Creation of many files, directories, copy, move, cat etc. so many commands we can learn in this activity.

1. **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

1. **ls –** ls command is used to list files or listing the content

# Syntax: ls

1. **touch-** used to create a file

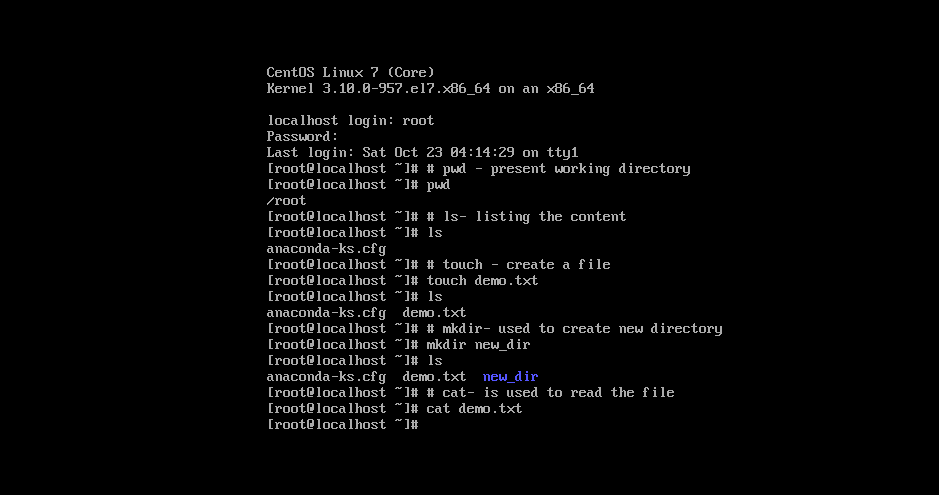
# Syntax: touch filename

1. **mkdir-** This command is used to create new directory

# Syntax: mkdir foldername

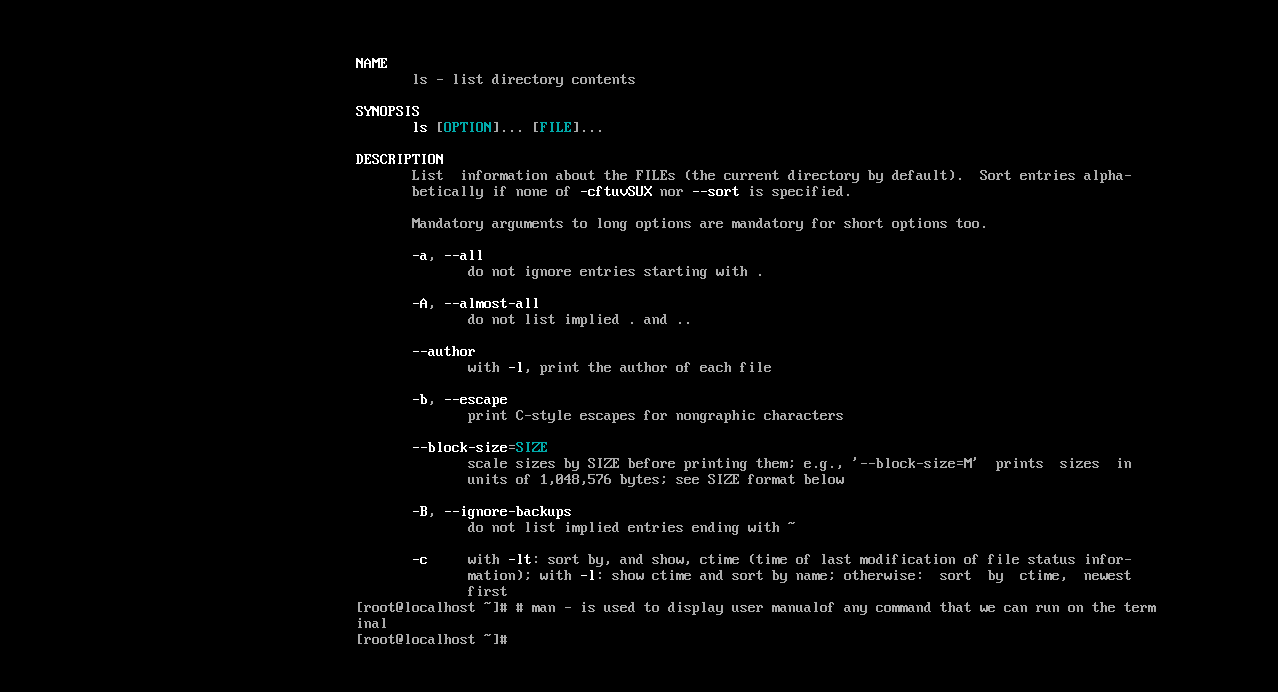
1. **cat-** cat command is used to read the file

# Syntax: cat filename



1. **man-** command is used to display user manual of any command that we can run on the terminal

# Syntax: man command name

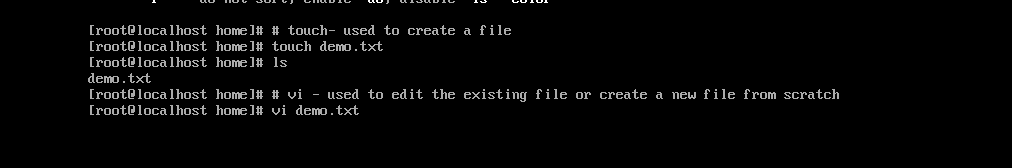


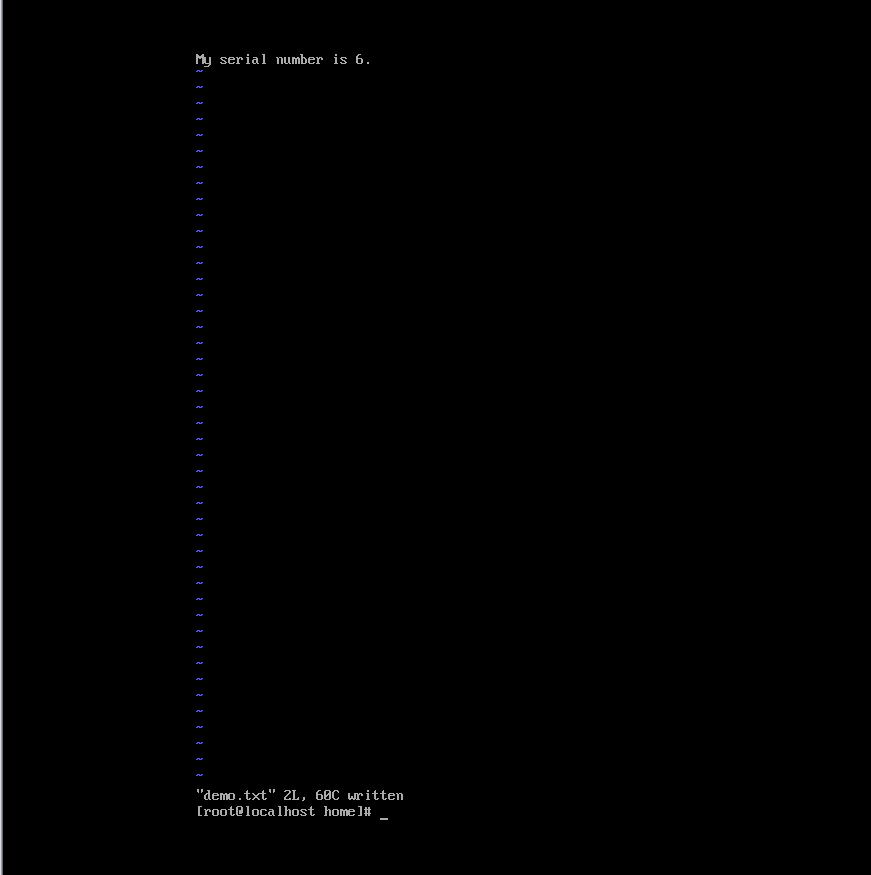
1. **touch-** used to create a new file

# Syntax: touch filename

1. **vi –** Used to edit the existing file or create a new file from scratch.

# Syntax: vi filename





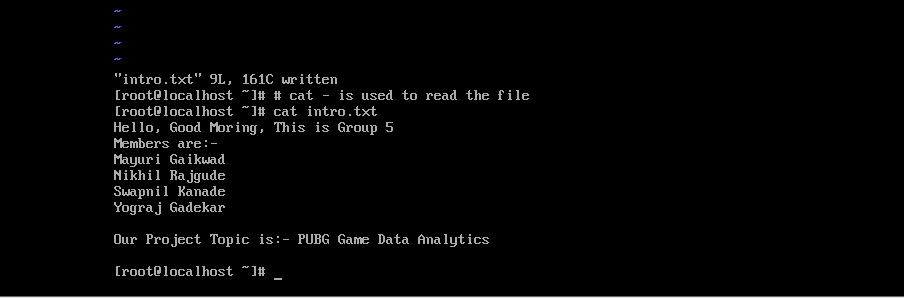
**Insert mode –** This mode enables you to insert text into file To come in insert mode you simply type **i .**

**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**)**

1. **cat-** used to read the file

# Syntax: cat demo.txt



1. **mkdir-** create new folder

# Syntax: mkdir new\_dir

1. **cd –** change the directory

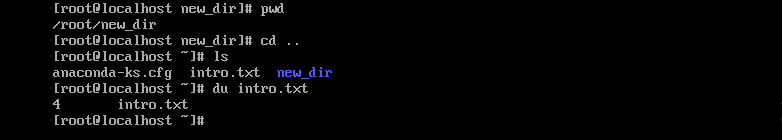
# Syntax: cd

1. **cd .. –** used to move to the parent directory of current directory

# Syntax: cd ..

1. **du -** This command is used to move to the parent directory of current directory

# Syntax: du filename



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

# Syntax: rmdir foldername

1. **rm –** command is used to remove the file

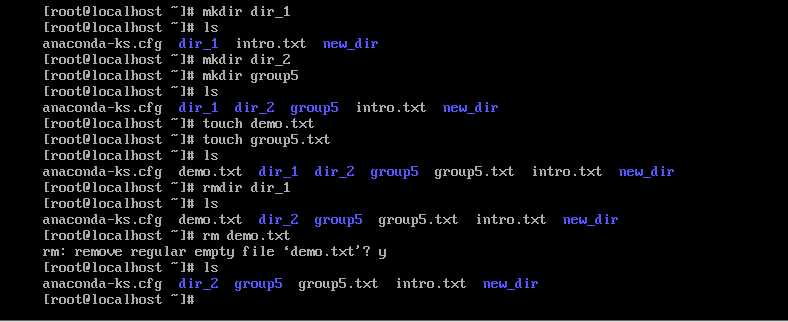
# Syntax: rm filename

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

# Syntax: rm -r dirname

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

# Syntax: rm – rf dirname



1. **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)

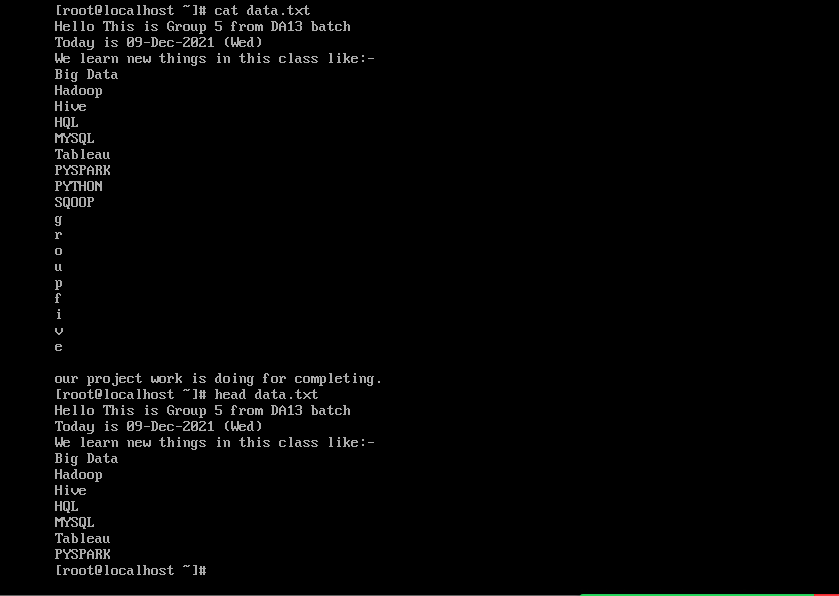
# Syntax: cp <existing file name> <new file name>

1. **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

1. **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>



**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.

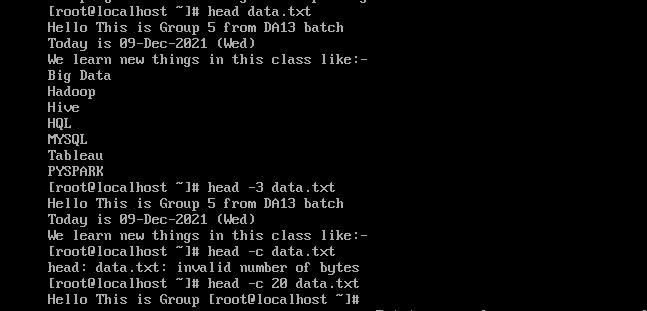
# Syntax: head <file name> <file name>

1. **head – n** option displays specified number of lines.

# Syntax: head -n <file name>

1. **head – c** command counts the number of bytes of a file.

# Syntax: head -c <number> <file name>



1. **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>

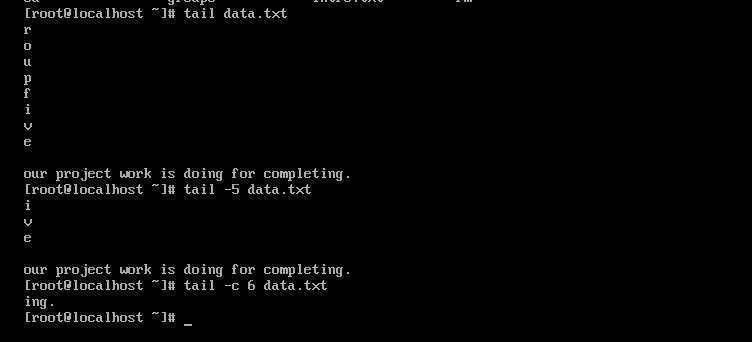
1. **tail -n** option displays the specified number of lines.

Syntax: tail -n **<number> <file** name**>**

# Example: tail -n 5 data.txt

1. **tail -c** option displays the specified number of bytes from the last.

# Syntax: tail -c <number> <file name> Example: tail -c 6 data.txt



1. **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.

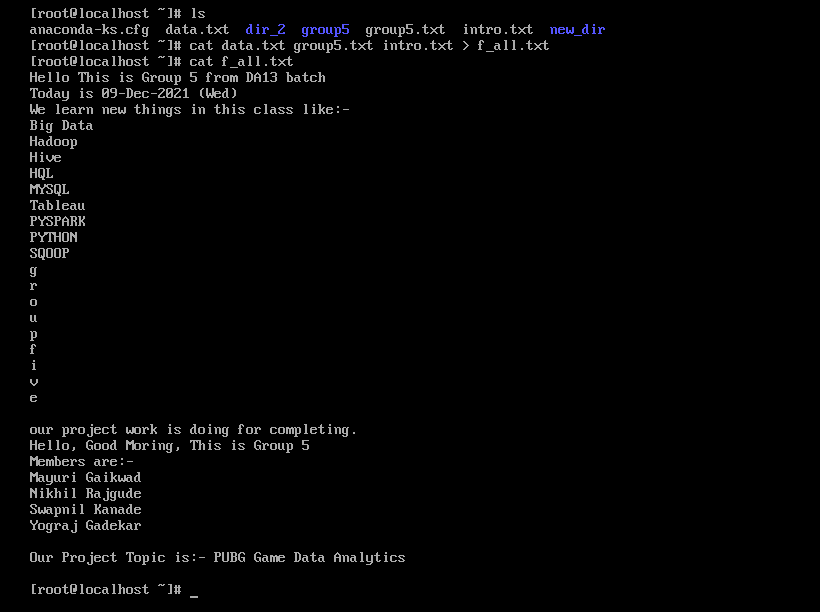
# cat -n filename.txt

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

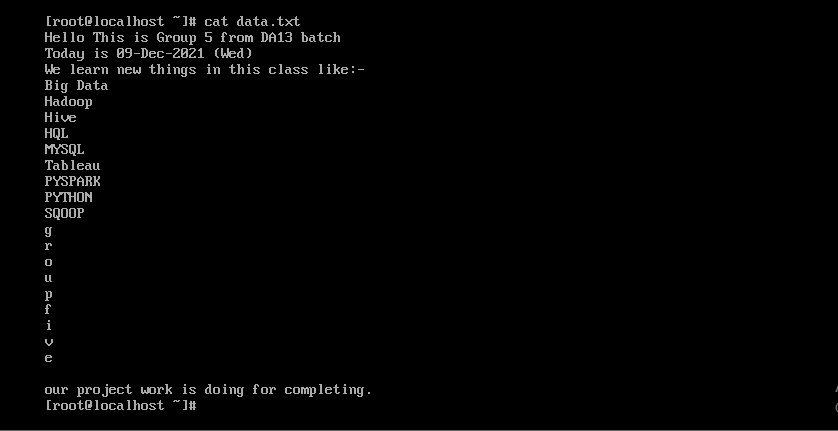
cat -b filename

1. **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.

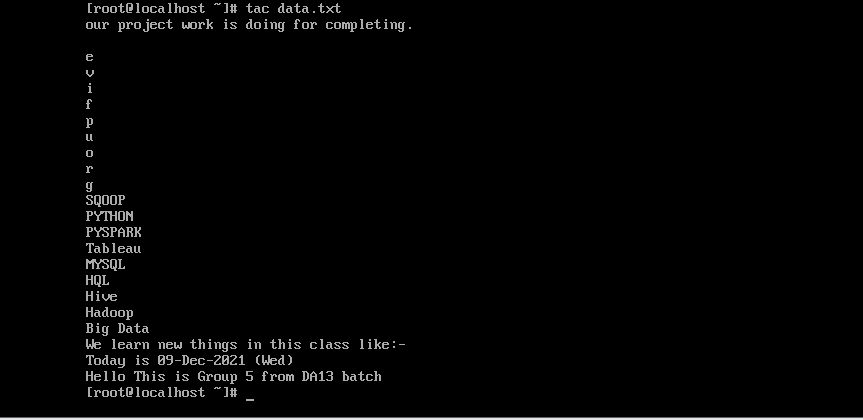
**Syntax-** tac **<file** name**>**



# cat data.txt



**tac data.txt**



1. **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>



1. **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>

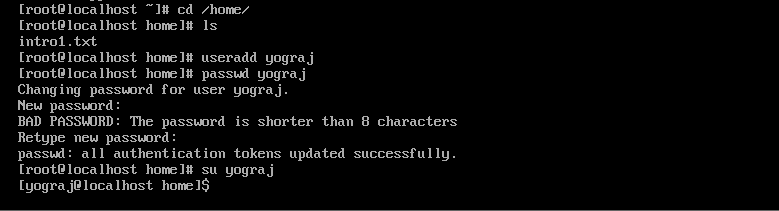
1. **su-** The su command allows you to run a shell as another user.

# Syntax: su <username>

1. **useradd-** Used to add or remove a user on a linux server

# Syntax: useradd username

1. **passwd-** A user can set the password with the command **passwd Syntax: passwd**



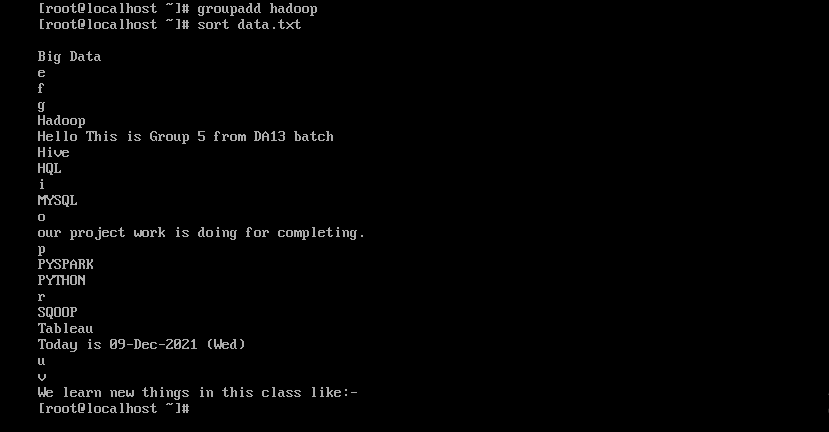
1. **groupadd-** The groupadd command creates or add a group in our system.

# Syntax: groupadd <groupName>

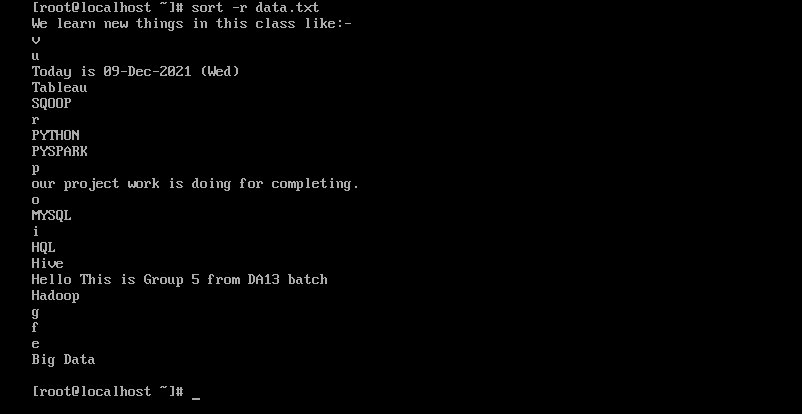
1. **sort -** The 'sort' command sorts the file content in an alphabetical order.

# Syntax: sort <fileName>

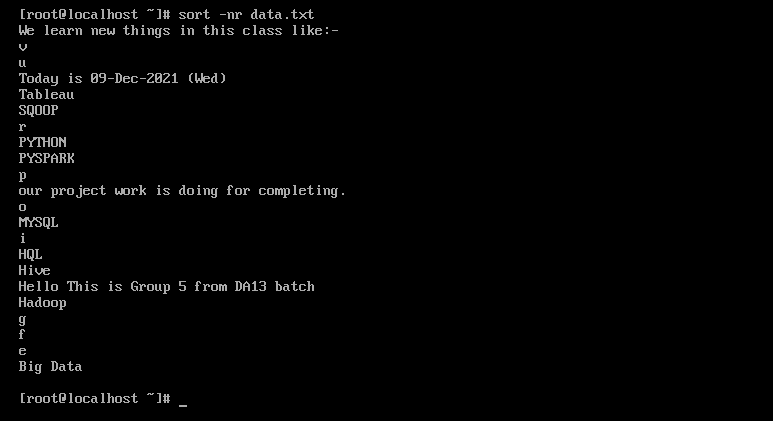
**Example: sort data.txt**



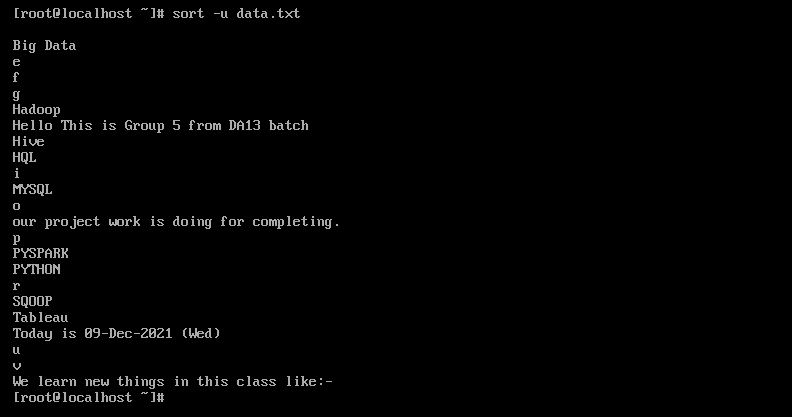
**sort -r** option for sorting in reverse order



**sort -nr** option for numeric data in reverse order



**sort -u** option to sort & remove duplicates



**sort -k** option to find particular column

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

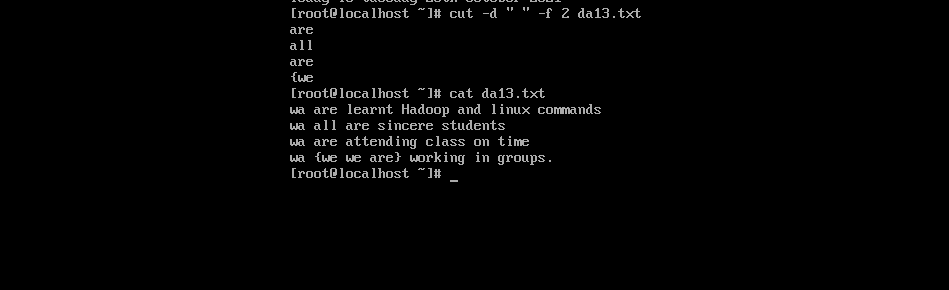
1. **cut –** used to cutting out the section from specified files

# 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.



1. **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>

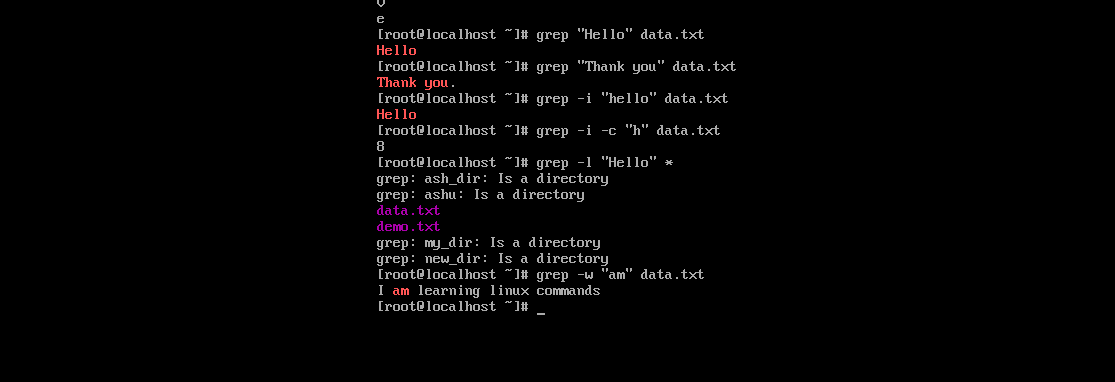
**Example: grep “Hello” demo.txt**

**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



1. **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

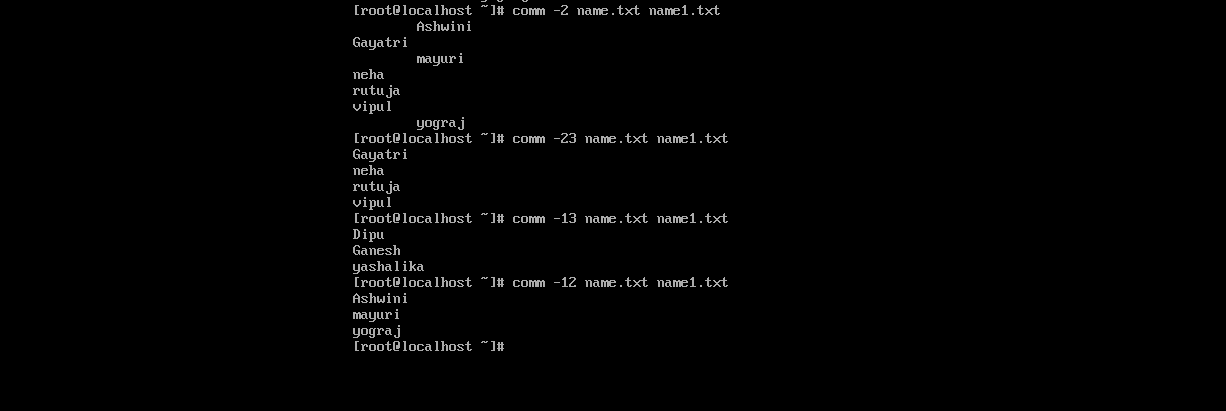


**To display single column Syntax:**

**comm -23** (To display first column)

**comm -13** (To display second column)

**comm -12** (To display third column)



1. **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

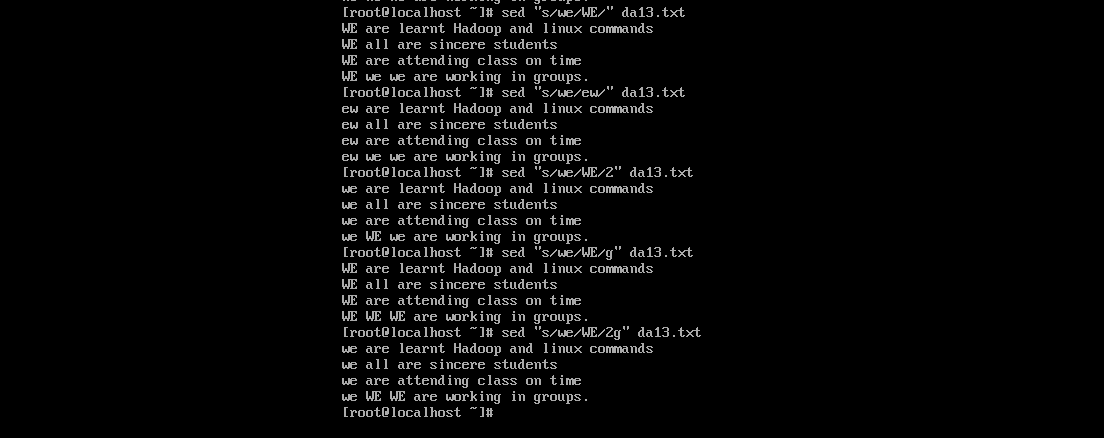
**-n** used to display only replaced line

**-p** for print

**d** used for deleting

**.bak** (for backup)

**-i.bak**



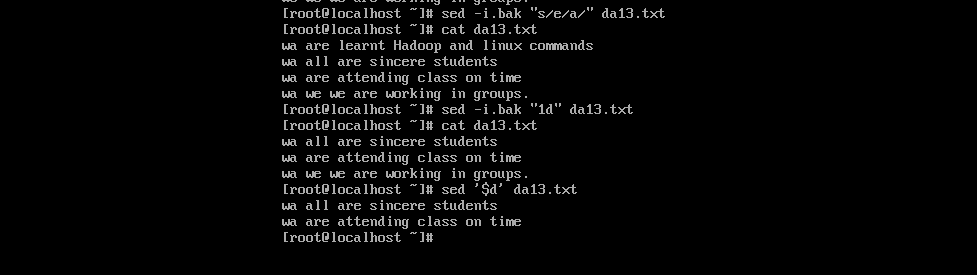
**sed “s/we/we all/” da13.txt (replacing a particular word) sed “s/we/we all/p” da13.txt (print the line)**

**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)**

**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)**



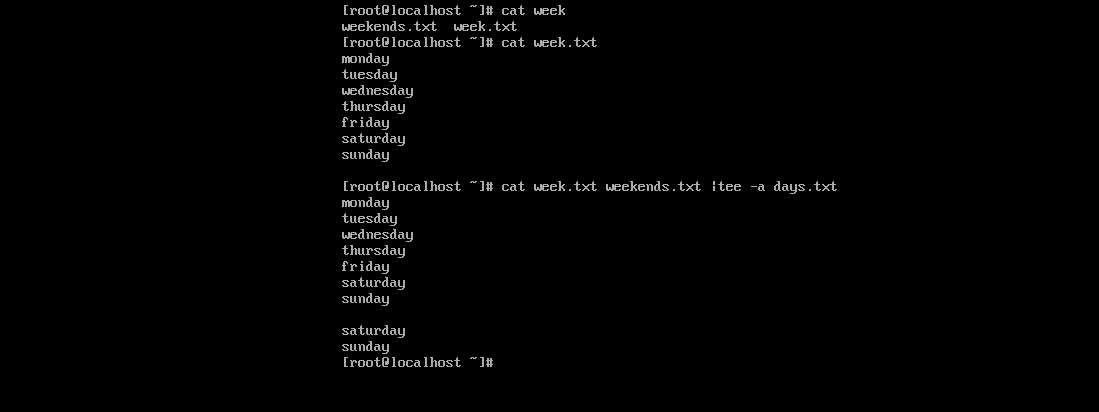
1. **tee-** tee command is quite similar to the 'cat' command, with only one difference. It puts stdin on stdout and also put them into a file.

It is one of the most used commands with other commands through piping.

# Syntax- tee <options> <file name>

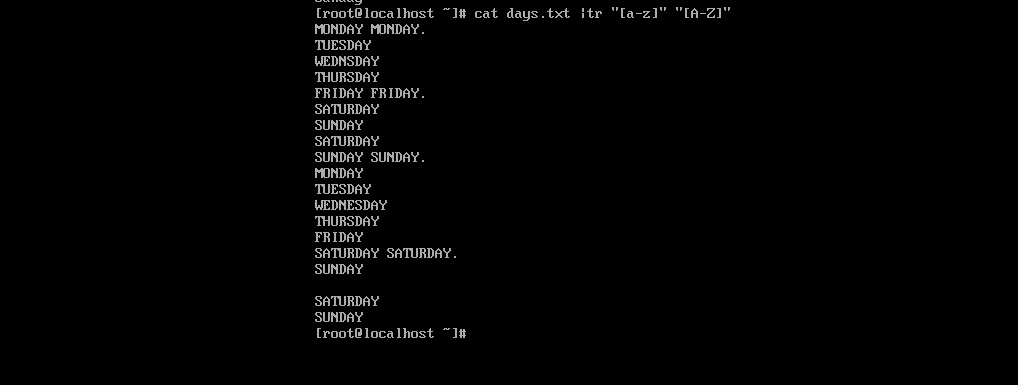
**-a, --append:** It is used to append the data to the given files, it does not overwrite data.

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



1. **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>

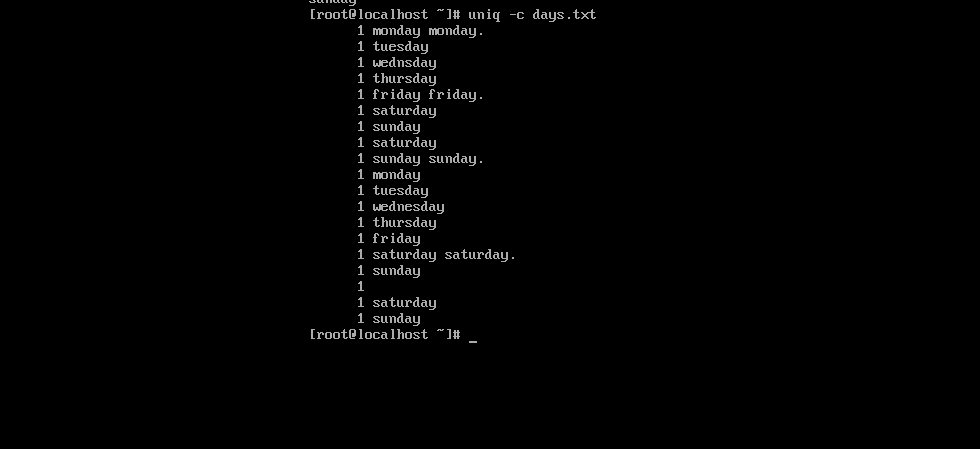


1. **uniq -** uniq command is used to remove all the repeated lines from a file.

**Syntax: uniq [OPTION]... [INPUT] Example: uniq -c days.txt (count)**

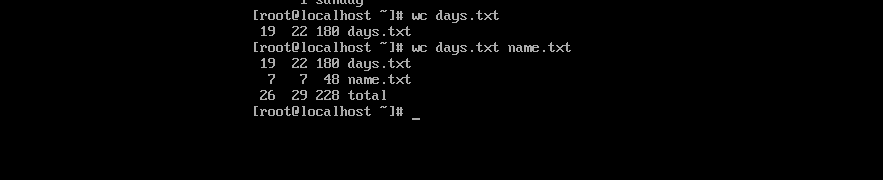
**uniq -d days.txt (show only repeated option uniq -D days.txt (print repeated lines)**

**uniq -u days.txt (only unique)**



1. **wc –** used to count the line,words,& characters in a file.

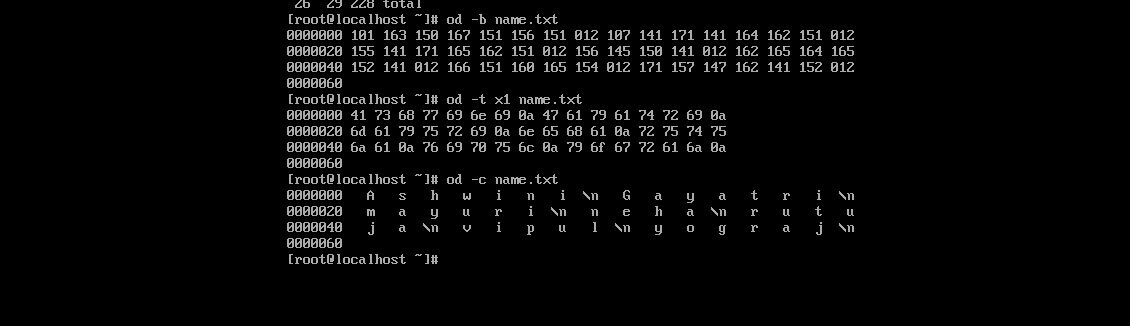
# Syntax: wc filename



1. **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>**



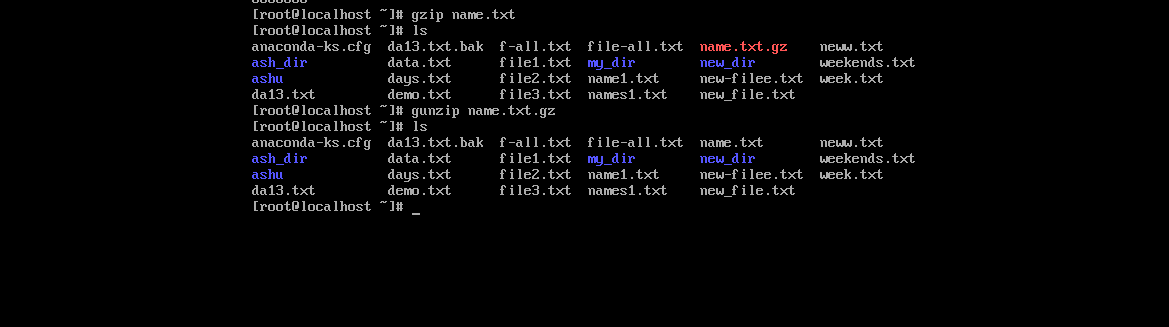
1. **gzip –** used to truncate the file size. It is compressing tool.

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

# Syntax: gzip filename

1. **gunzip –** used to decompressed the file.

**Syntax: gunzip filename**



1. **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**



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

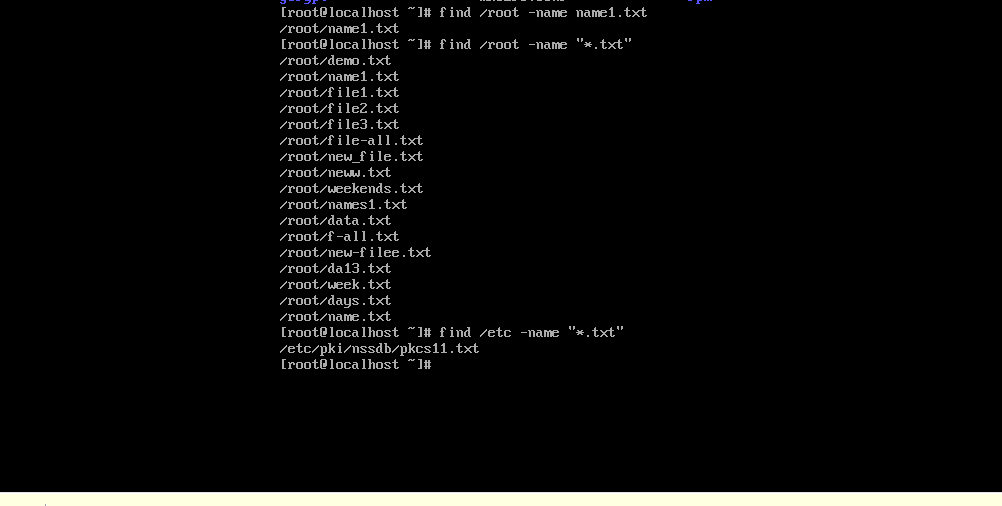
# Symbols –

**(.)** for current directory name

/ for root

Example: find /root -name names.txt find /root -name “\*.txt”

find /etc -name “\*.txt”



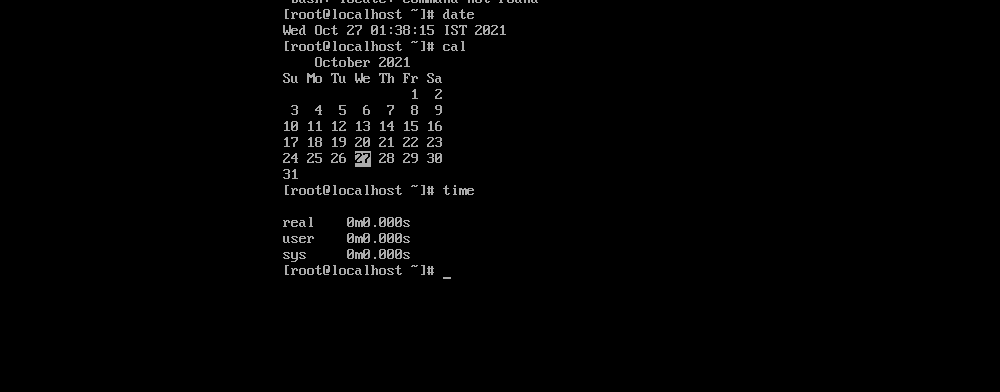
1. **locate –** used to search a file by filename. Similar to ‘find’ command

# Syntax: locate <filename>

1. **date –** used to display date time and timezone

# Syntax: date

1. **cal –** used to display the current month calendar with the current date highlighted
2. **time-** used to display and set the current system time.

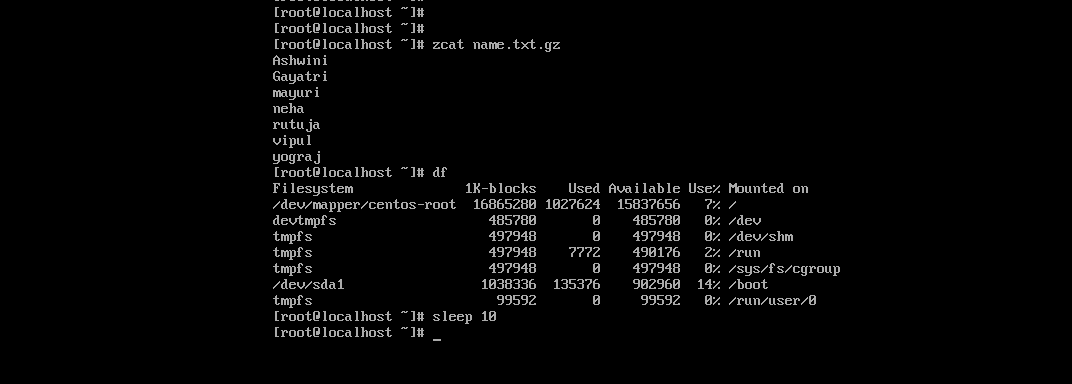


1. **zcat –** used to display compressed file**. Syntax: zcat filename**
2. **df –** used to display the disk space

**Syntax: df**

1. **sleep –**

**Syntax- sleep <time>**



1. **mount-** used to connect an external device file system to the systems file system.

# Syntax- mount -t type <device> <directory>

1. **clear-** used to clear the terminal screen.

# Syntax- clear

1. **exit-** used to exit the shell where it is currently running.

**Syntax- exit**