**Session 1: How to Solve Big Data Problem**

**Assignment 1.2**

Student Name: Harikrishnan D

Course: Big Data Hadoop & Spark Training

Start Date:  2018-05-06

End Date:  2018-06-02

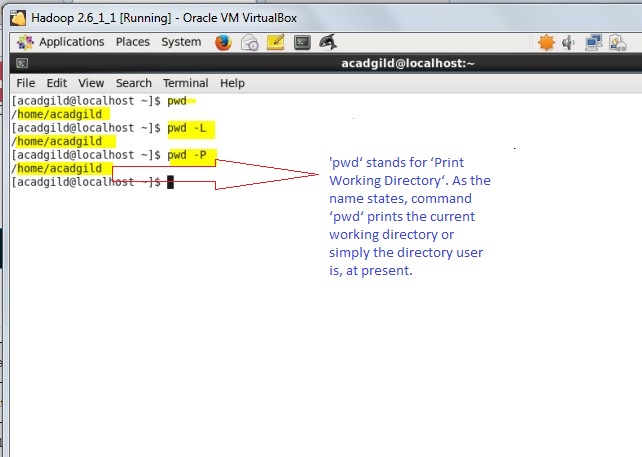
**Assignment 1.2 Question** –Explain the below linux commands with an example. Share the screenshot of each command with the output:

1. pwd
2. vi
3. touch
4. mkdir
5. rm
6. ls
7. echo
8. cat
9. who
10. cd
11. date
12. cal
13. mv
14. cp
15. which

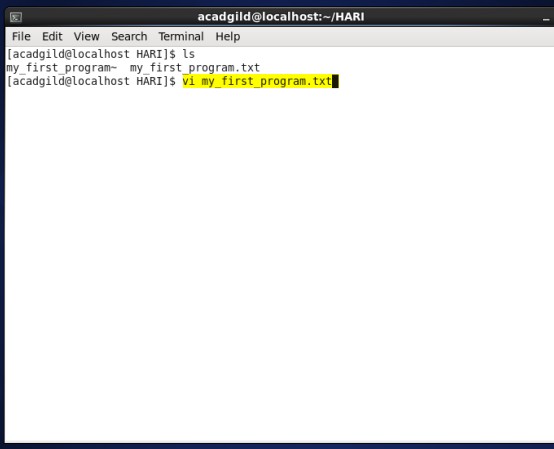
**Linux Commands Outputs:**

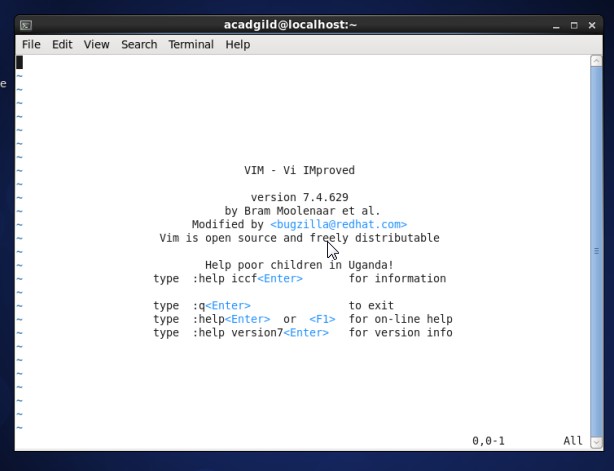
1. **pwd**

**‘**pwd’ stands for ‘Print Working Directory’. As the name states, command ‘pwd’ prints the current working directory or simply the directory user is, at present. It prints the current directory name with the complete path starting from root(/).



1. **Vi modes**

****

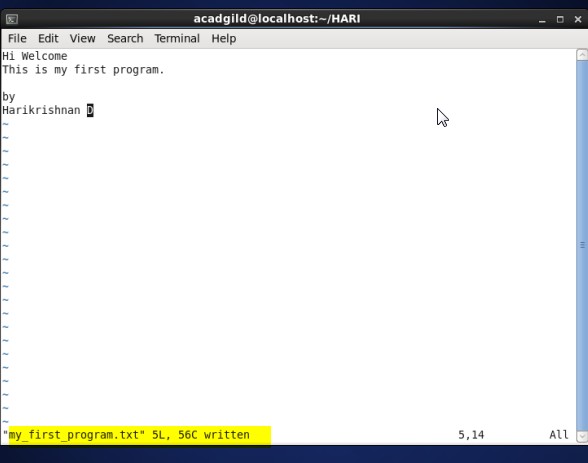
****

**Three modes in vi :-**

1. **Command mode:**

* Accessed by typing “**Esc**”
* In this mode all the keys pressed by the user are interpret to the editor command.

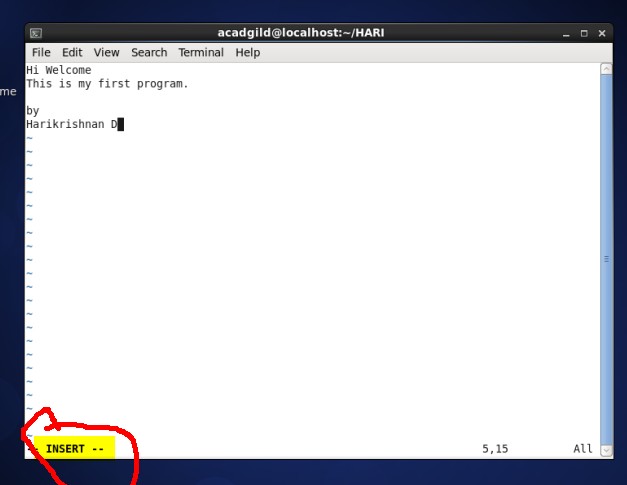
**Example:-**



1. **Input mode:-**

* Accessed by typing “**i**”
* This mode permits insertion of new text, editing of existing text or replacement of existing text.

**Example**:-



1. **Ex mode(Last Line mode):-**

* The bottom line of the vi screen is called ex mode.
* When you start vi by default it is in command mode.
* You exit the input mode by pressing the Esc key to get back to the command mode.

**Common Vi Commands:-**

* Moving Cursor Position
* You can move around only when you are in the command mode
* Arrow keys usually works(but may not)
* The Standard keys for moving cursor are:
* **h** – for left
* **l** –for right
* **j** – for down
* **k** – for up
* **w** – to move one word forward
* **b** – to move one word backward
* **$** - takes you to the end of line
* **<enter>** takes the cursor the beginning of next line
* **-** -(minus) moves the cursor to the first character in the current line
* **H** – takes the cursor to the beginning of the current screen Home position.
* **L** – moves to the Lower last line
* **M** – moves to the middle line on the current screen
* **Control-d** scrolls the screen down (half screen)
* **Control-u** scrolls the screen up (half screen)
* **Control-f** scrolls the screen forward (full screen)
* **Control-b** scrolls the screen backward (full screen)

**Editing Text commands:-**

* **x** – deletes the current character
* **d** – is the delete command but pressing only d will not delete anything you need to press a second key
* **dw** – deletes to end of word
* **dd** – deletes the current line
* **d0** – deletes the beginning of line
* The **vi** commands can be used followed by a number such as

**n<command key(s)>**

* For example **dd** deletes a line **5dd** will delete five lines.
* **yy** – (yank copy current line to buffer
* **nyy** – Where n is number of lines
* **p** – Paste the yanked lines from buffer to the line below.
* **P** – Paste the yanked lines from buffer to the line above.
* **u** – undo the changes made by editing commands
* **.** – (dot or period) repeats the last edit command
* **ZZ**  - is for save and Exit (Note this command is uppercase)

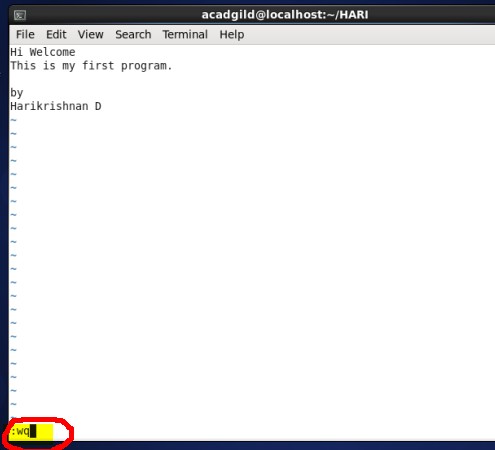
**File Savings Related Commands (ex mode)**

You need to press [Esc] key followed by the colon (:) before typing the following commands:

* **q <enter>** - Quit
* **q!** – Quit without saving changes i.e. discard changes and quit file
* **r fileName** – Read data from file called **filename**
* **w** - Save file and continue editing
* **wq** – Write and quit(save and exit)
* **x** – Same as wq command i.e. write and quit
* **w fileName** – Write to file called fileName(save as)
* **w!** **fileName** – Overwrite to file called fileName (save as forcefully)

Note: for overwrite the file need to be add **!** at end of command.

**EXAMPLE:-**



1. **Touch Command:**

The **touch command** can be used to modify the access/modification timestamps of files. It is more often used to actually just create an empty file quickly.

**Touch Command Options :-**

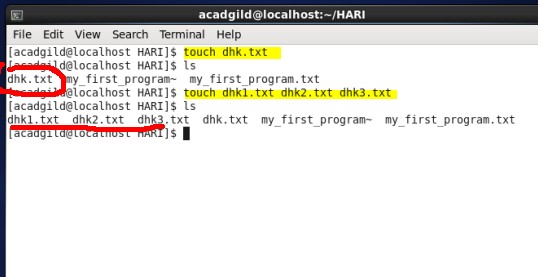
* **-a**, change the access time only
* **-c**, if the file does not exist, do not create it
* **-d**, update the access and modification times
* **-m**, change the modification time only
* **-r**, use the access and modification times of file
* **-t**, creates a file using a specified time

1. **Create a blank file:-**

Example : $ touch dhkfile.txt

1. **Create multiple files with touch:-**

Example : $touch dhkfile1.txt dhkfile2.txt dhkfile3.txt

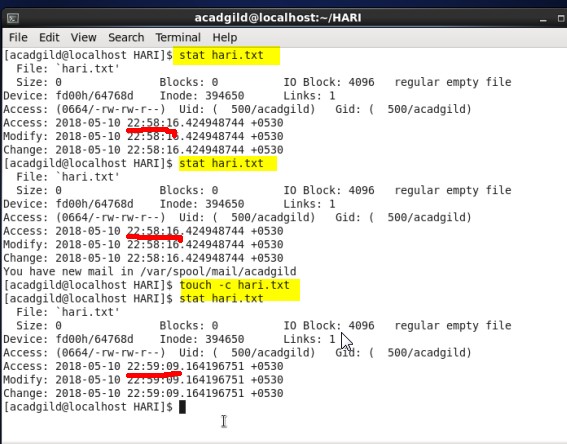


1. **Avoid creating new files:-**

Example : $touch –c hari.txt

1. **Change file access time –‘a’**

Example : $touch –a hari.txt



1. **Change the modified time –‘m’**

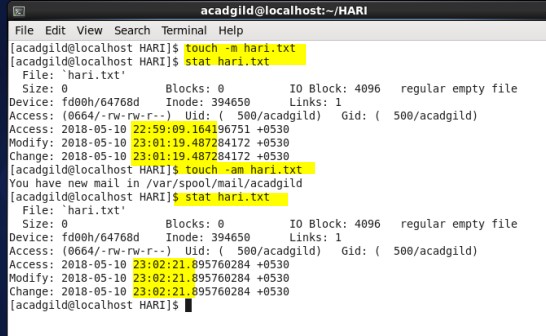
Example : $touch –m hari.txt

1. **Change the modify time of multiple files using wildcard:-**

Example : $touch –m \*.txt

1. **Change access and modification time together:-**

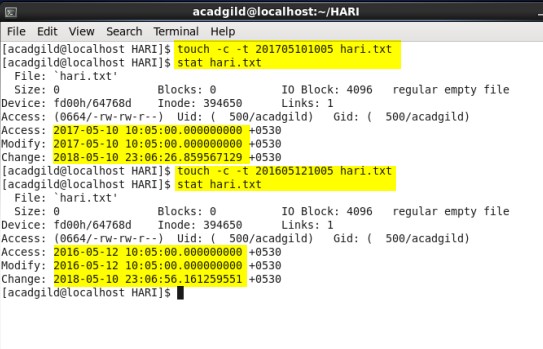
Example : $touch –am hari.txt

****

1. **Set a specific access/modify time instead of current time:-** ( Format is [[CC]YY]MMDDhhmm[.ss] )

Example : $touch –c –t 18051023 hari.txt

**\*Note -** If you omit the c option, a new file will be created with the given datetime if it does not exist.

****

1. **Use the timestamp of another file as reference:-**

Example : $touch –r ref.txt hari.txt

1. **Specify datetime as a string:-**
2. It provides the date only. The time is automatically set to 00:00.

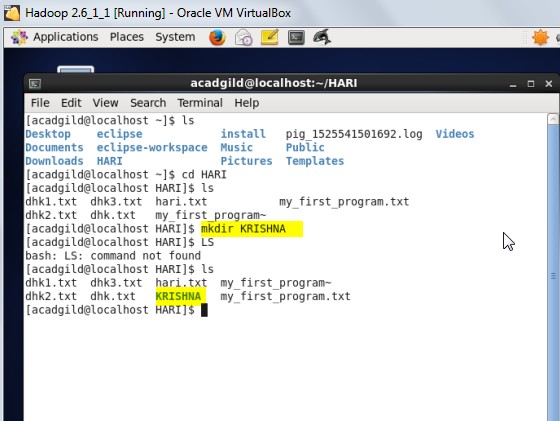
**Example :** $touch –c –d ’10 May’ hari.txt

1. It provides just the time, and the current date will be selected.

**Example :** $touch –d ’10:23’ hari.txt

1. **mkdir**

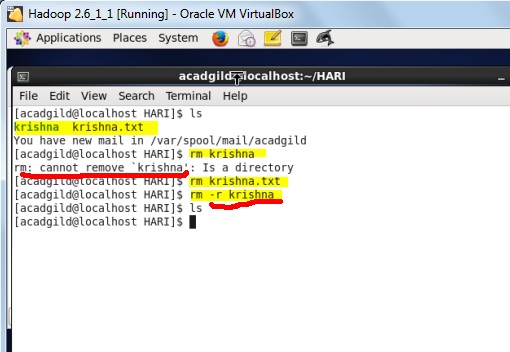
If the specified DIRECTORY does not already exist, mkdir creates it.

****

1. **rm**

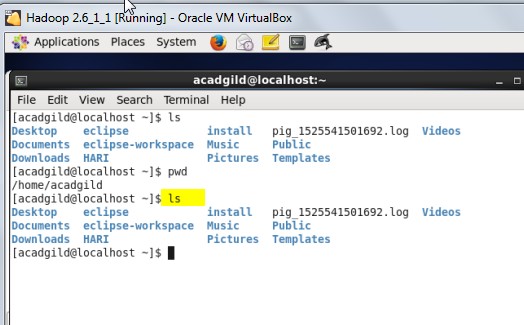
The **rm** command removes (deletes) files or directories.**rm** removes each specified FILE. By default, it does not remove directories; see Removing Directories below for details. The removal process unlinks a file name in a file system from data on the storage device, and marks that space as usable by future writes. In other words, removing files increases the amount of available space on your disk. The data itself is not destroyed, but after being unlinked, it becomes inaccessible. Remove your files wisely! The effects of an **rm** operation cannot be undone.

Example: a ‘**krishna’** and ‘**krishna.txt’** has been created under the directory ‘**HARI’**, the text files and the directory is going to be removed using **rm** command.



1. **ls**

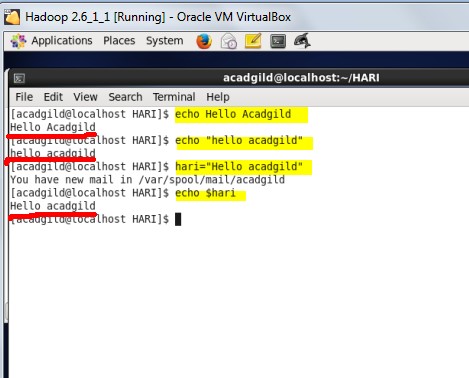
**ls** is a command to list files,



We have 15 other Basic ‘ls’ Command Examples in Linux, i.e. ls –I, ls –a, ls –lh, ls –F, ls –r, ls –R, ls –ltr, ls –IS, ls –i, ls - -version, ls - -help, ls -1 /tmp, ls –n.

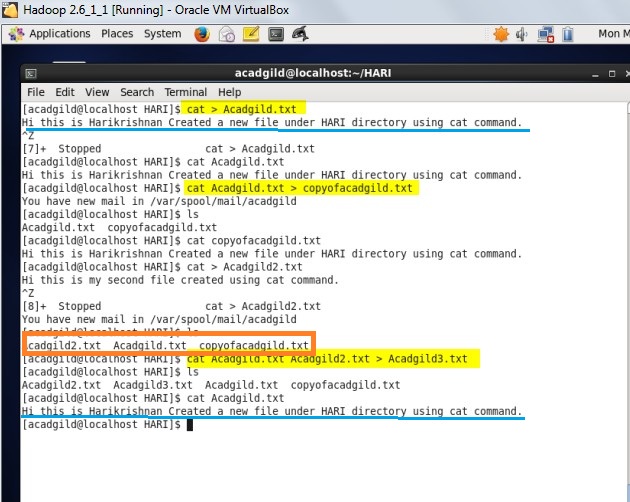
1. **echo**

The echo Command. **echo** is a built-in command in the bash and C shells that writes its arguments to standard output. A shell is a program that provides the command line (i.e., the all-text display user interface) on Linux and other Unix-like operating systems

****

1. **cat**

The cat (short for “**concatenate**“) command is one of the most frequently used command in Linux/Unix like operating systems. **cat** command allows us to create single or multiple files, view contain of file, concatenate files and redirect output in terminal or files.

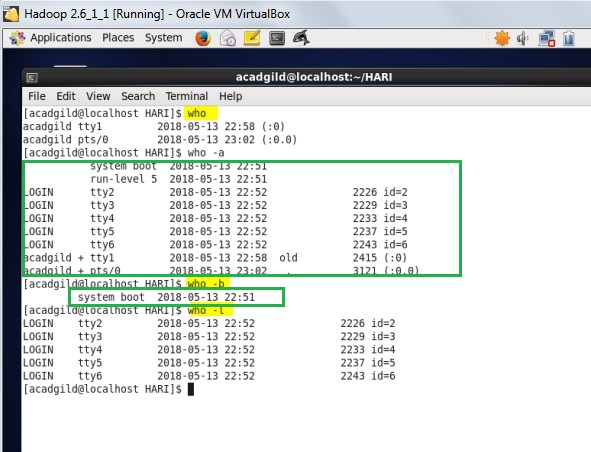


1. **who**

As a Linux user, sometimes it is required to know some basic information like:

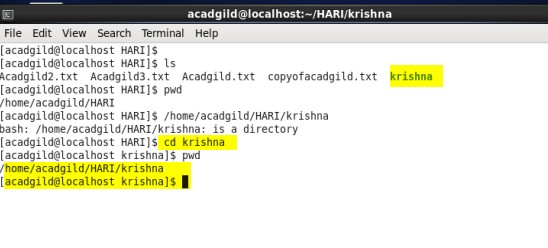
* Time of last system boot
* List of users logged-in
* Current run level etc

Though this type of information can be obtained from various files in the Linux system but there is a command line utility 'who' that does exactly the same for you.

****

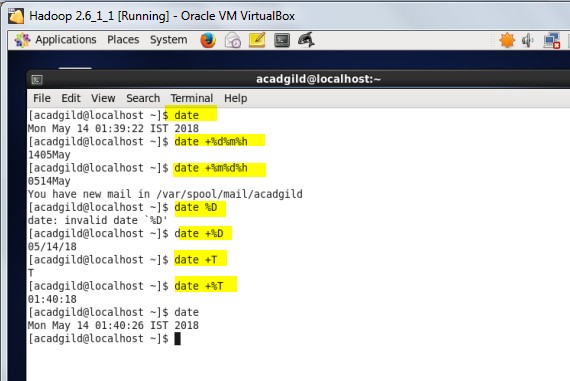
1. **cd**

The **cd** command is used to change the current directory (i.e., the directory in which the user is currently working) in Linux and other Unix-like operating systems

****

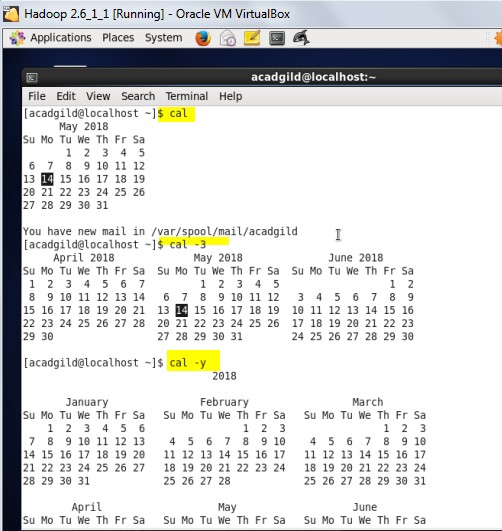
1. **date**

The **date** command is used to print out, or change the value of, the system's time and date information.



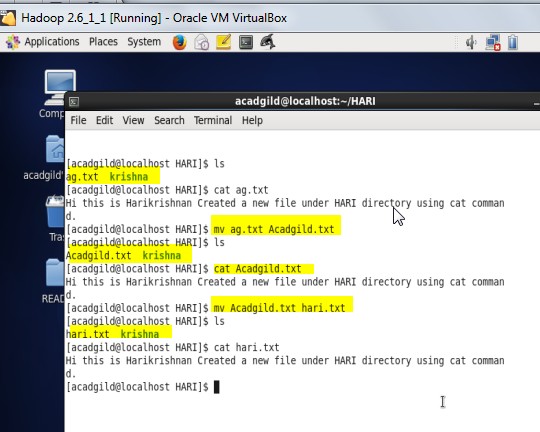
1. **cal**

Display a conveniently-formatted calendar from the command line.



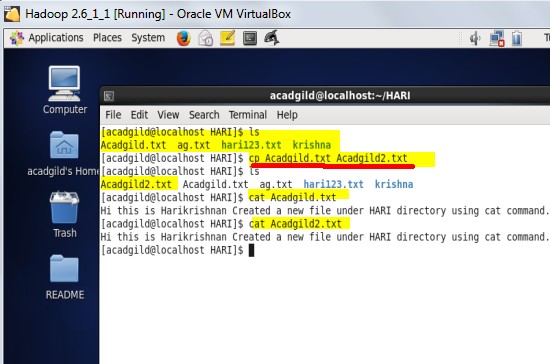
1. **mv**

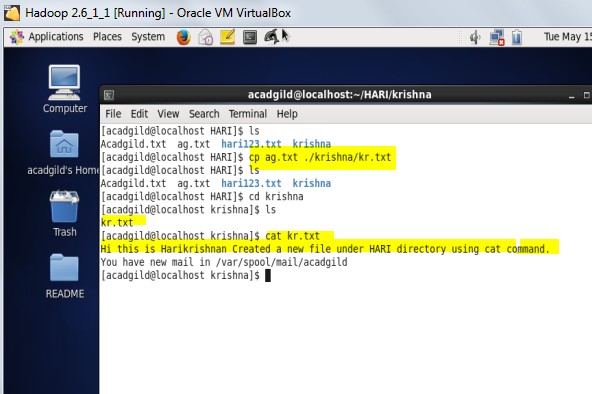
Linux mv command. **mv** command is used to move files and directories.



1. **cp**

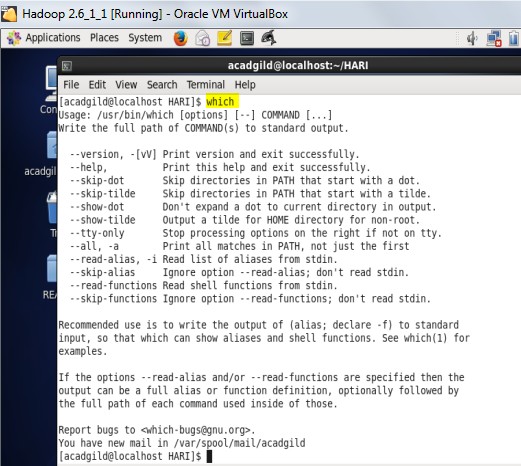
cp is a Linux shell command to copy files and directories.





1. **which**

Which command is very small and simple command to locate executables in the system.It allows user to pass several command names as arguments to get their paths in the system. “**which**” commands searches the path of executable in system paths set in $PATH environment variable.



**THANK YOU**