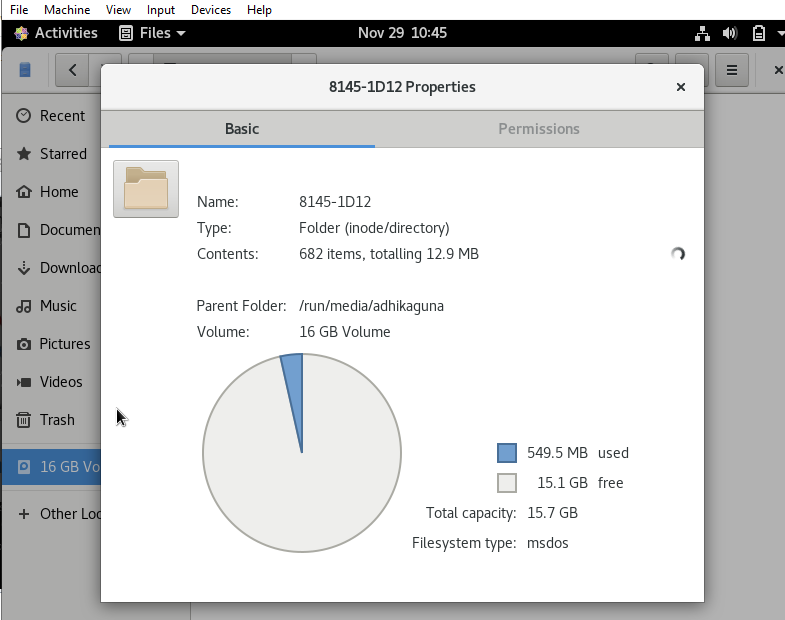
Adhika Guna – IT1 – 001201800018

Operating System Week 12

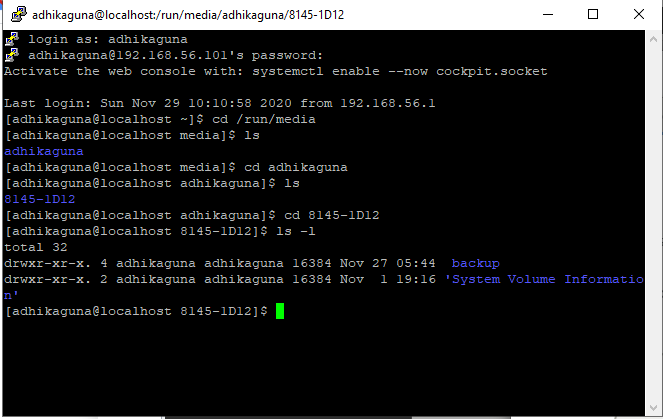
**Mount:**

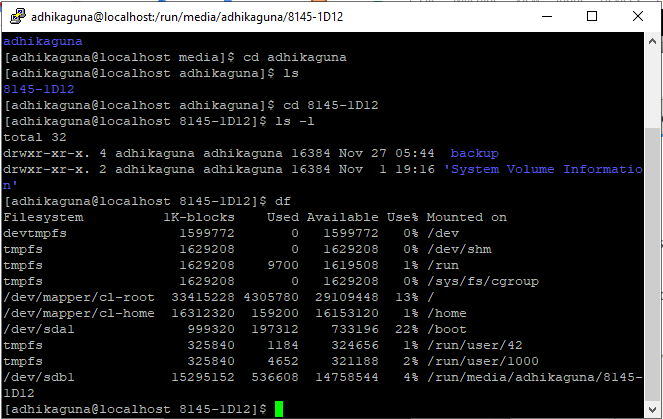
* Image below is the properties of the flash disk in the Centos. The name of the flash disk is 8145 – 1D12.



* The images below show the list of files in the flash disk on the /run/media/adhikaguna/8145-1D12 directory:

(8145-1D12 is the flash disk name.)

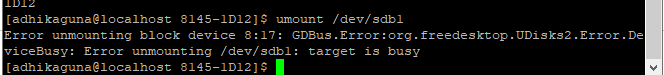




/dev/sda1 is the first disk drive and its first partition.

/dev/sdb1 is the second disk drive and its first partition.

* The image below shows an attempt to unmount the usb drive:

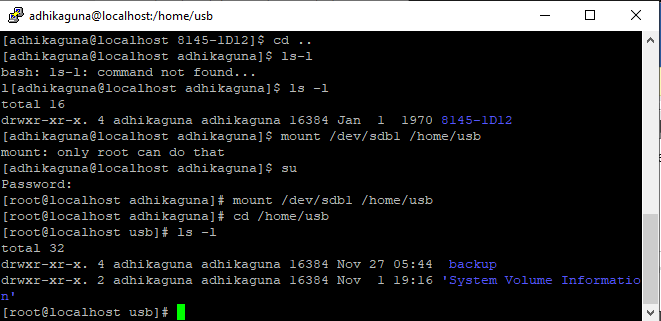


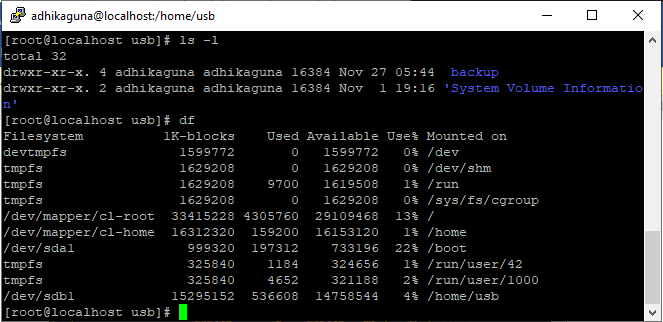
We can’t unmount the drive in this situation since the drive is in use,

we need to get out of the drive directory first so, the drive is in idle state and not in use.

After going up 1 directory, we are able to unmount the disk drive, and the disk drive is not mounted on the /run/media/adhikaguna anymore.

* The images below shows the mounting process of the usb drive on another directory:



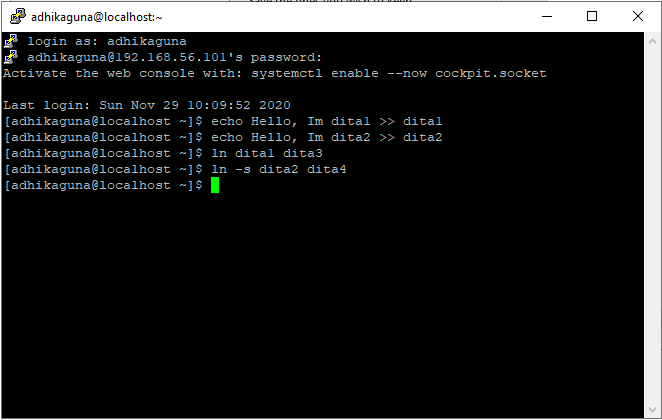


The image shown above is the process of mounting the usb drive in a newly created directory which is /home/usb.

Mount is the process to attach a drive to the file tree. In this case, we attach 8145-1D12 to the /home/usb directory.

**Link and Process:**

* The images below show the steps required for link:



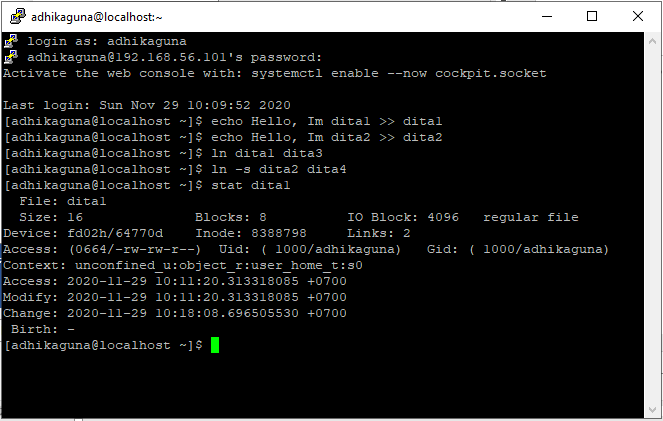
The function of echo is to fill out the file with a text, in this case dita1 content is “Hello, Im dita1” and dita2 content is “Hello, Im dita2”.

Link is a pointer to a file. It allows more than one file name to refer to the same file elsewhere. In this case dita1 and dita2 acted as a reference to dita3 and dita4 respectively. dita1 and dita3 is linked using hard link, to applied hard link we can use “ln dita1 dita3”, while dita2 and dita4 is linked using soft link, to applied soft link we can use “ln -s dita2 dita4”.

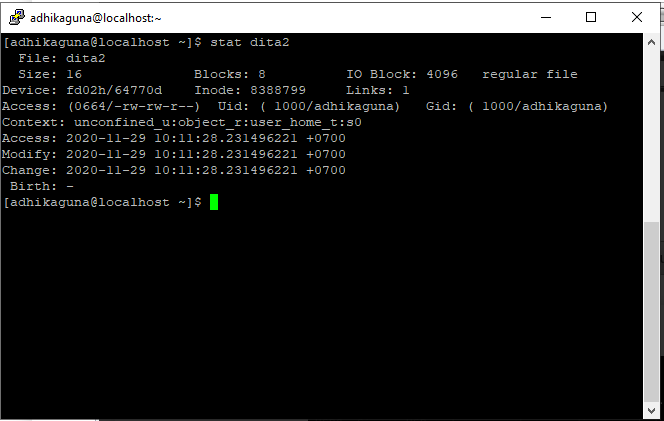
Difference between hard link and soft link:

* + Hard link file is assigned the same Inode value as the original, thus they reference the same physical file location.
  + Hard link file is still linked even if the original files are moved throughout the file system and the file still can be accessed even though the original file is deleted.
  + Hard link files are unable to cross different file systems.
  + Soft link file is assigned a separate Inode value that points to the original file.
  + Soft link file cannot show its content if the original file is deleted
  + Soft link file can cross different file systems.
* The image below shows the stat of dita1, dita2, dita3, dita4:

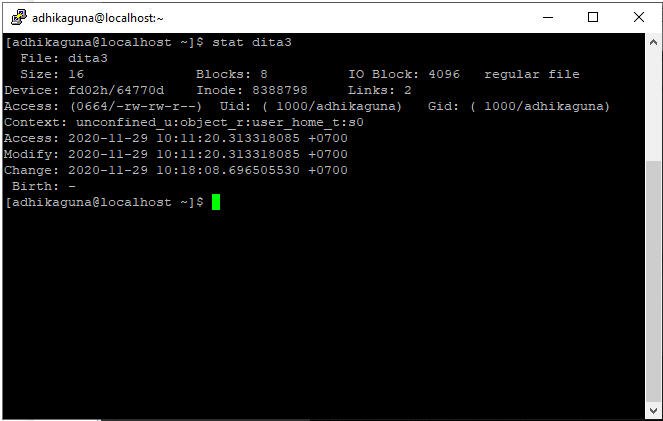
Stat dita1:



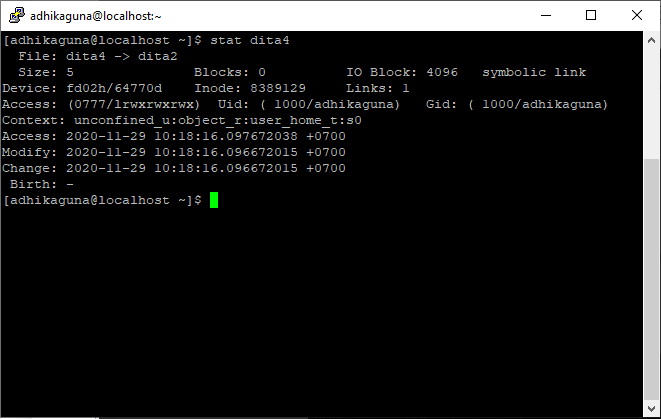
Stat dita2:



Stat dita3:



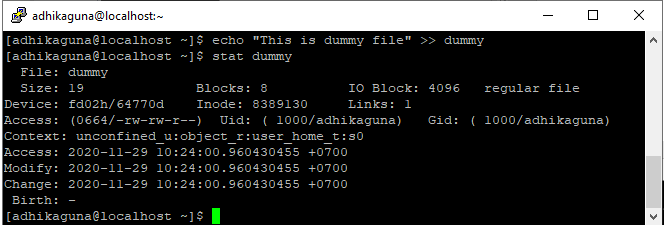
Stat dita4:



The stat command gives the information such as the file Inode, its links and the properties such as Access, Modify and Change.

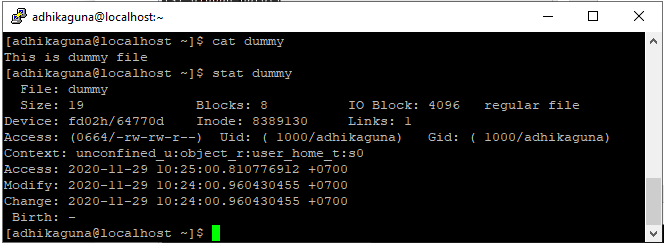
* The image below shows stat for an example file:

In this case the name the example file is “dummy”.



The stat in file have Access, Modify, and Change properties. Access is the indicator for when the file is last accessed, this can change if the user is doing anything that involving accessing this file and then the indicator will change.

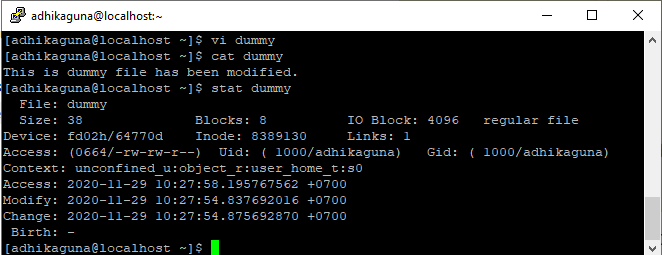
The image below shows a change to Access:



Next, we’ll change the file content so we will see the different in the file properties.

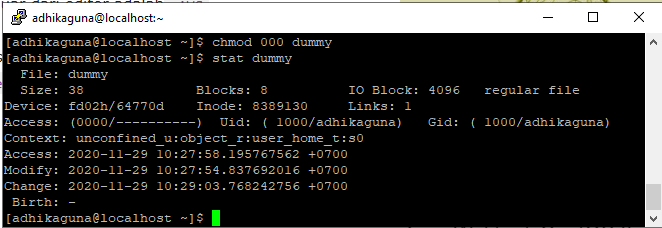
Modify is the indicator for when the file is last modified, this can change if the user has modified the content of the file.

* The image below shows a change to Modify:

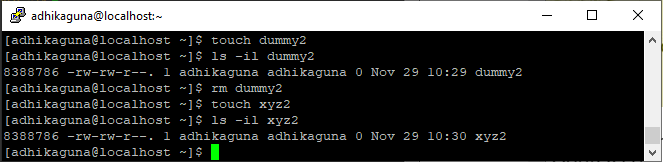


Change property is the indicator for when the file is last changed, this can change if the user has changed the content of the file, the permission of the file etc.

The image below shows the change of Change property:



* The image below shows the requirement in slide 5 on link\_process\_linux3 power point file:

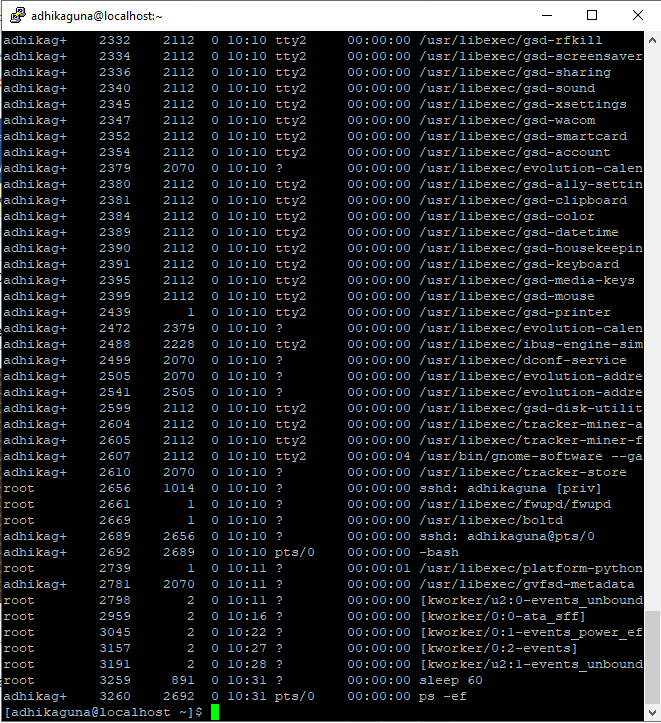


Conclusion on Inode:

* The image below shows the requirement in slide 6 and 7 on link\_process\_linux3 power point file:

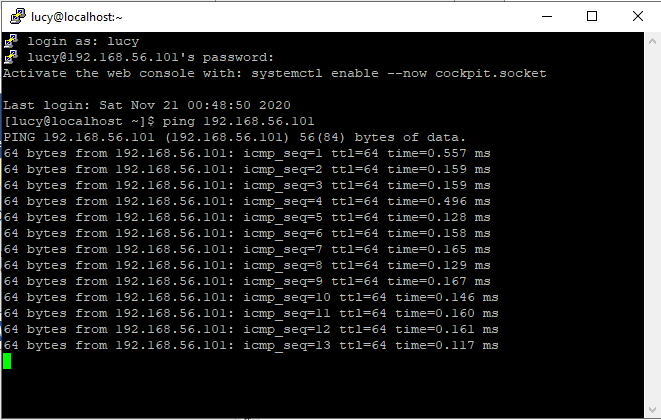
This is the result after typing ps -ef in user adhikaguna:

Command “ps” is mean process so, when we type ps it’s shown the process that are running, and -ef in this case is shown all process that are running along with the full format information that are corresponding to the process.

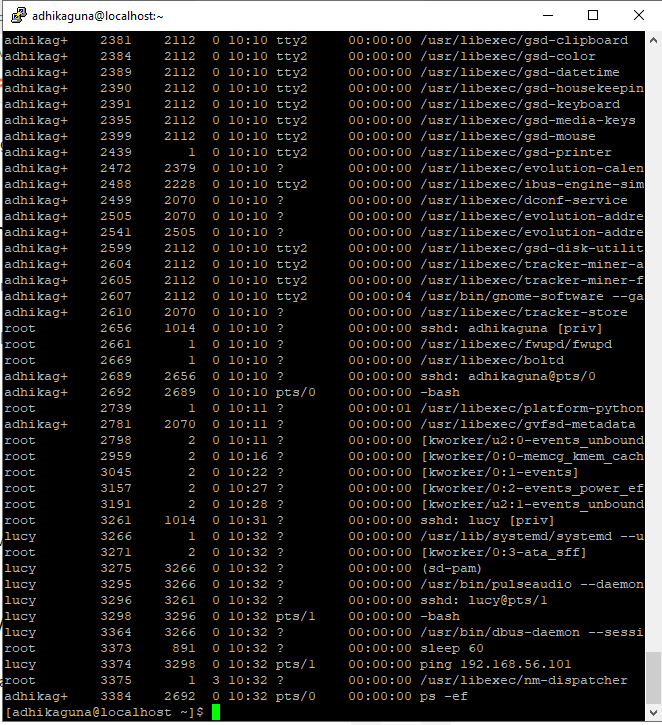


The image below shows another user ping to 192.168.56.101:

In this case user “lucy” ping to 192.168.56.101



The image below shows ps -ef after lucy has done ping:

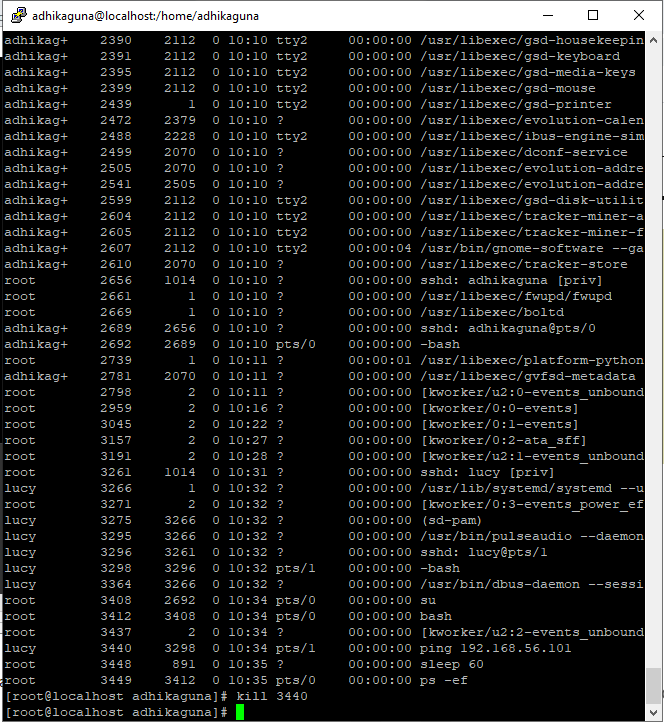


After another user has done ping, we can see a new process that is done by another user, in this case the user is Lucy.

If we want to kill the ping process, we need to choose the correct **PID** (Process Id), in this case the process that we want to kill is the ping from user “lucy” and had a **PID** 3440

In order to kill a process, we need to type “kill <PID>”, and using the su (super user).

The image below shows how to stop the process by killing it using its **PID**:



* The image below shows the process from user “lucy” automatically terminated after the main user kill the process.

