User management:

There is 2 types of users :

1. Root Users
2. Regular Users
3. System Users

Root User:

It is the super user. root user can create/modify/delete other user information. root user has access to all file systems, directories.

We can find the root user identity by **#**  symbol in the CLI

We can find the normal user identity by $ symbol in CLI

**\*\*\*USER Creation\*\*\***

>> adding new user:

**#useradd <new\_account\_name>**

>> to set the pass word for the user

**#passwd username**

when you run this command then, these operations are performed:

**1. home directory** is created (**/home/username** by default).

**2.** These hidden files are copied into the user’s home directory,-

* bash\_logout
* bashrc
* bash\_profile

user account information is stored in the **/etc/passwd** file. This file contains a record per system user account and has the following format

**[username]:[x]:[UID]:[GID]:[Comment]:[Home directory]:[Default shell]**

>> if you want to specify user home directory then:

**#useradd -d /dir1/dir2 <username>**

>> add user without home directory

**#useradd -M <username>**

>> create user with specific uid and gid (group id)

**#useradd -u <userid> -g <groupid> <username>**

>> to switch to normal user

**#su ~ regular\_user (su means switch user)**

>> to delete the user and the home directory

**#userdel --remove username**

**\*\*\*GROUP Creation\*\*\***

>>> to create the group

**#groupadd groupname**

>>> to add the user to the group at the user creation level

**#useradd -g groupname username**

>>> to add the user to the group post user creation

**#usermod -g groupname username**

>> to add the user to primary group

**#usermod -g groupname username**

>> to add the user to secondary group

**#usermod -G groupname username**

>> to know the user id and group id details

**#id username**  it will list out the user id and group id

**Sudo user :**

Add the user details in /etc/sudoers file

**Regular Commands in Day2Day basis:**

**\*\*\*File creation and deletion\*\*\***

>> to check the present working directory (directory means folder. Hereafter folder will be referred as directory)

**#pwd**

>> to create empty file

**#touch file\_name**

>>> to see the last user session details

**#last**

>>> to check the live user session data

**#w**

>> to know the date and time stamp

**#date**

>> to create a directory

**#mkdir directory\_name**

>> to know when your system is restarted

**#uptime**

>> to list out the files and directories

**#ls -ltr**

>> to list out the files and hidden files also

**#ls -la**

>> to list out only files with the same extension

**# ls -ltr \*.txt**  🡪 it will list out all the files with .txt exetension

>> to change the directory

**#cd dirname**

>>to create directory in recursive manner

**#mkdir -p dir1/dir2/dir3/dir4** ( p means parent here)

>> to remove a file

**#rm -i file\_name**

>>to remove a file without confirmation.

**#rm -f filename**

>> to remove a directory

**#rmdir dir\_name**  or  **rm -d dirname**

>>to remove a directory recursively and forcefully

**#rm -rf dirname**

>> to remove the files with \*.txt

**#rm -fv \*.txt**

**\*\*\*cat commands\*\*\***

>> to display the content on screen

**#cat filename**

>> to copy content from one file to another file

**#cat filename1 > filename2 (**It will remove the content from file2 and copy content of file1**)**

>> to append the content of file2 with file1 content

**#cat filename1 >> filename2**

>> to show the content lines with number

**#cat -n filename**

>> to show multiple files content on screen in single short

**#cat filename1 filename2 filename3**

>> to copy content of the multiple files into single file

**#cat filename1 filename2 filename3 >> filename4**

>> to show the content of all .txt files

**#cat \*.txt**

>> to check certain string in the file

**#grep “string\_name” filename** or cat filename | grep

>> to check the number of lines in a file

**#wc -l filename** -- to check the number of lines in a file

**#wc -c filename**  🡪 to check the number of characters in a file

>> to check the first 10 line from the file

**# head filename**

**#head -20 filename --**  to check the first 20 lines from the top in file

>> to check the last 10 lines from the file

**# tail filename**

**# tail -20 filename** -- to check the last 20 lines from the bottom

>> to search any particular key string in the file

**#grep string filename**

**#grep -in filename**  🡪 to check the string in case sensitive manner with the line number

**\*\*\*VI editor\*\*\***

VI stands for Visual Instrument. This is popular text editor in Linux OS . By using this , We can create/modify the content of a file .

**Modes in VI editor:**

1. command mode
2. insert mode
3. esc mode

>> to write something in the text editor go to insert mode as below

**ESC+i**

>> to save and quit the file do as below

**ESC+shift:+ wq**

>> to quit the file do as below

**ESC+shift:+q**

>> to save the file and quit forcefully

**ESC+shift:+wq!**

>> to search any word in the file using VI editor, go to command mode

**ESC+/string\_name**

>> to delete the 1 line or n number of lines go to command mode

**ESC+dd**  🡪 to delete only single line

**ESC+ndd** 🡪 to delete n lines from the cursor

>> to copy 1 lines or n number of lines

**ESC+yy**  🡪 yanking

**ESC+2yy** 🡪 to copy the 2 lines from the cursor

**ESC+p 🡪**  to paste the content whatever is copied

>> to go to the last line in the file

**ESC+G**

>> to create a line below the current line

**ESC+o** 🡪 o is small letter and it is case sensitive

**ESC+O** 🡪 to create a line above the current line

>> to go to the first letter of the current line

**ESC+^**

>> to go to last letter of the current line

**ESC+$**

**\*\*\* To know the system INFO\*\*\***

>> to know the number of cores.

**cat /proc/cpuinfo**

**lscpu**  -- to check the cpu info

>> to know the RAM details

**free -g 🡪 to know the details in GB**

**free -m 🡪 to know the details in MB**

**total used free shared buff/cache available**

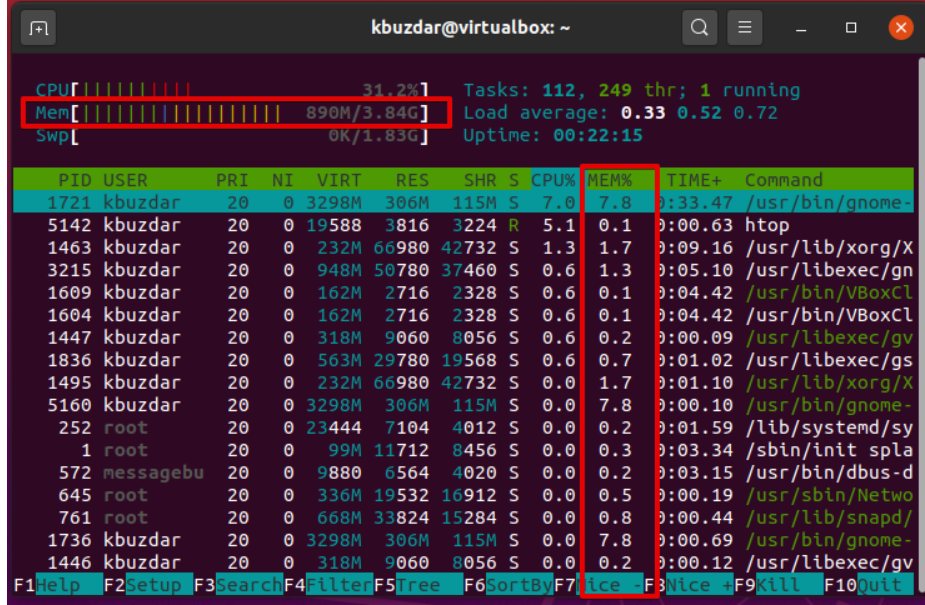
**Mem: 8075208 3204964 1310540 551232 3559704 4198340**

**Swap: 2097148 0 2097148**

* **total** - This number represents the total amount of memory that can be used by the applications.
* **used** - Used memory. It is calculated as: used = total - free - buffers - cache
* **free** - Free / Unused memory.
* **shared** - This column can be ignored as it has no meaning. It is here only for backward compatibility.
* **buff/cache** - The combined memory used by the kernel buffers and page cache and slabs. This memory can be reclaimed at any time if needed by the applications. If you want buffers and cache to be displayed in two separate columns, use the -w option.
* **available** - An estimate of the amount of memory that is available for starting new applications, without swapping.

>> to install htop **Yum install htop -y**

**htop**

****

>> to check the disk free

**df -kh**

>> to check the any particular directory disk usage

**df -ksh /directory name**

**df -ksh \*** 🡪 It will list out all the directories with respective disk usage

**\*\*\*Process Management in Linux\*\*\***

>> to check all the process in Linux

**#ps -ef**

>> to check particular process in the linux

**# ps -ef | grep process\_name**

**Example:**

1. yum install nginx -y
2. systemctl status nginx
3. systemctl start nginx
4. ps -ef | grep nginx
5. kill -9 nginx\_id
6. systemctl status nginx

**#ps -ef | grep process\_id**

**#kill -9 process\_id**  -- to kill the process

>> to find the top 10 process consuming more memory in the Linux

**ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem | head**

**ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%cpu | head**

**Process states in Linux:**

In Linux, a process is an instance of executing a program or command. While these processes exist, they will be in one of the five possible states:

* Running or Runnable (R)
* Uninterruptible Sleep (D)
* Interruptable Sleep (S)
* Stopped (T)
* Zombie (Z)

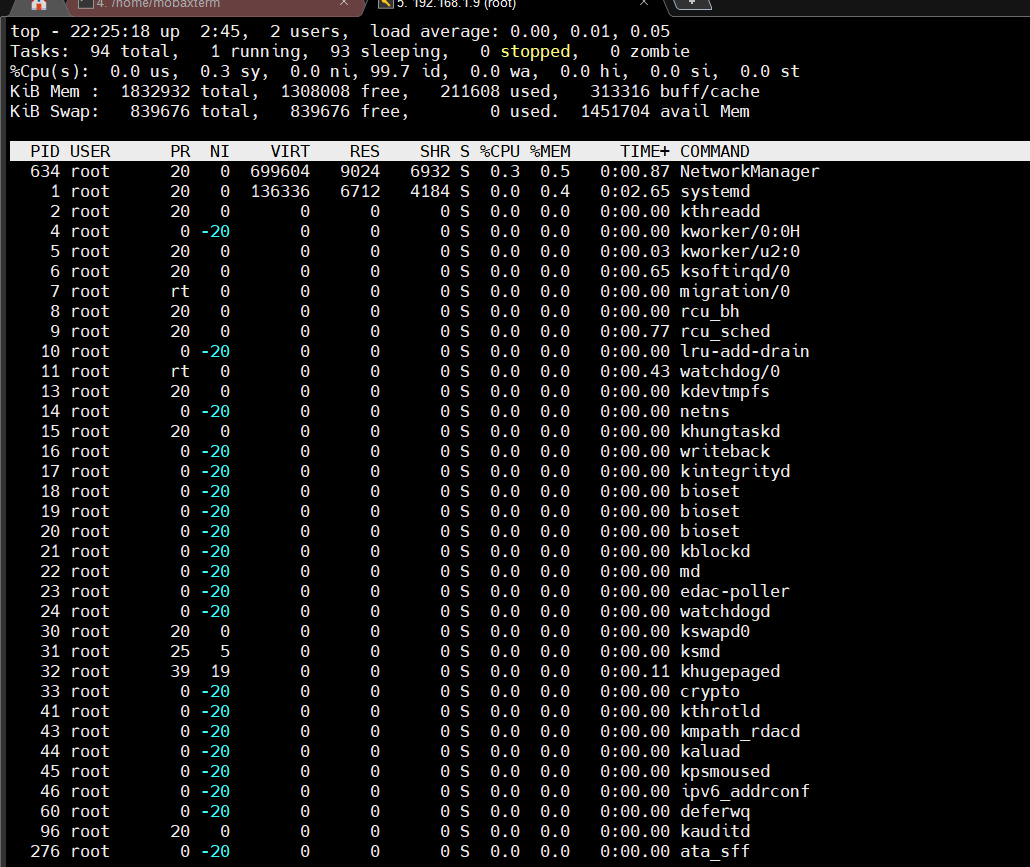
The process table in Linux (such as in nearly every other operating system) is simply **a data structure in the RAM of a computer**. It holds information about the processes that are currently handled by the OS. pointers to the executable machine code of a process.

**What is zombie process:**

A zombie process is **a process whose execution is completed but it still has an entry in the process table**. Zombie processes usually occur for child processes, as the parent process still needs to read its child's exit status

**\*\*\*TOP Command\*\*\***

**#top:**

****

**Note:**  In case your server is loaded with full of process , You can not even execute this top command

In that time , You can use cat /etc/loadavg

[root@localhost home]# cat /proc/loadavg

1. 0.01 0.05 2/111 5344

PID – process ID

USER – user details

PR – default priority set the system

NI -- Nice value set by us to any particular process . it ranges from -20 to 19

nice is used to invoke a utility or shell script with a particular CPU priority, thus giving the process CPU time than other processes. A niceness of **-20 is the highest priority and 19 is the lowest priority**. The default niceness for processes is inherited from its parent process and is usually 0.

VIRT – it is the number of bytes used by resources

RES - Reserved and not swappable memory for the process in RAM

SHR -- amount of memory shared by different resources in RAM

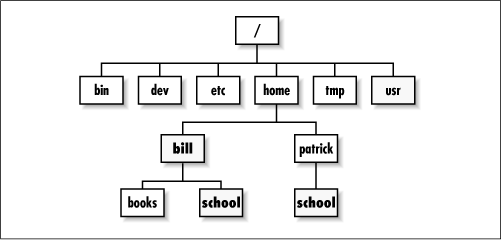
%CPU – CPU utilization by each process

%MEM -- % RAM utilized by each process

TIME – time stamp of that process created

CMD -- command because of which this process started

**\*\*\* Directory Structure\*\***



**/ --** Super directory

**/Etc --** configuration directory

**/bin** -- binary files of single user

**/sbin**  -- binary files , usually not related to normal users . It is used for super user related

**/home --** home directory for the normal users

**/tmp --** it is common folder for all the users .just like public folder in windows

**/usr** -- user program related folder

**/boot**  -- all bootable files are stored here

**\*\*\*Inode Concept\*\*\***

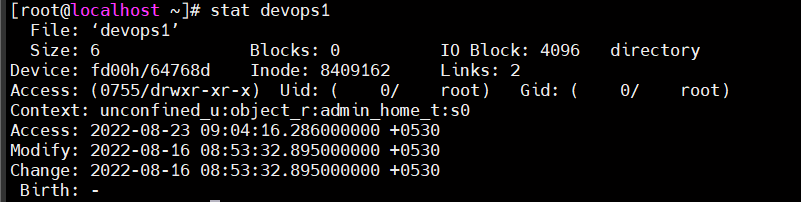
**Inode -**  Inode is nothing but a Index Node .

Whenever you create a new file your OS needs to keep a lot of information about the file, in UNIX like OS this information like file owner , file permissions etc, are stored in a data structure known as inode. So in simple words inode is data structure that holds information of a file.

**How to find the Inode of a file?**

Use the **stat**  command to check the inode of a file

**#stat filename**

****

**Modify time:**  it is about the content change in the file called as modified time , it is also called as mtime (Only when content is changed)

**Change time:**  last time , the file’s inode is changed (like permissions and ownership )

**\*\*\*Soft link and Hard link\*\*\***

**Soft link:**

**On-Directory:**

ln -s {origin directory path} {symbolic dir path}

**example:**

1. cd /root
2. mkdir -p a/b/c/d/e
3. cd a/b/c/d/e
4. touch samplefile
5. cd /root
6. mkdir f
7. **ln -s a/b/c/d/e f**
8. cd e and ls -ltr

**On-file:**

ln -s softlink\_originfile symbolic\_file

**example:**

* 1. cd /root/a/b/c/d/
  2. touch samplefile
  3. cd /root
  4. **ln -s /root/a/b/c/d/samplefile checkfike**
  5. vi checkfile

**Hard Link:**

**On-Directory:**

**Note:**  We can not create hard link on the directory

**On-File:**

ln hardlinkfilepath symbolic\_path

**Imp Points:**

1. **what if delete softlink , will it affect the original file?**

If you delete the the symbolic file , it will not affect the original file

But if you delete the original file , It will affect the symbolic file

1. **what is the inode number for softlink and hardlink?**

Softlink for the inode number is different from the symbolic file

Inode for the hard link is different from the original file

**\*\*\*Access Management\*\*\***

There are 3 types of permission exist in Linux Management

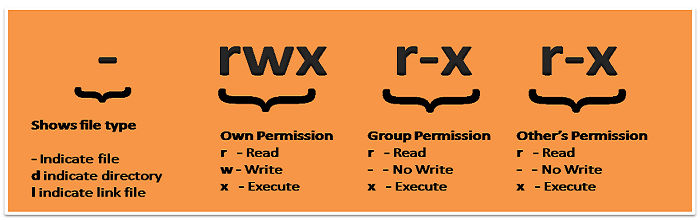
Read – 4

Write – 2

Execute – 1

Default permission for the directory is 755

Default permission for the file is 644



>> change the user , group permission , other group permission

**#chmod 777 filename/directory\_name**

**Example:**

1. Useradd devops
2. Useradd aws
3. Groupadd tcs
4. Usermod -g tcs devops
5. Usermod -g tcs aws
6. Su – devops
7. Cd /tmp
8. Touch sample
9. Chmod 664 sample (change from 644 to 644 )
10. Su – aws
11. Cd /tmp
12. Vi sample.txt

**chown:**

>>> to change the ownership of any file

**#chown owner\_name filename**

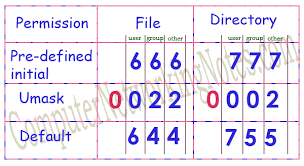
>>> to change the group of any file

**#chgrp group\_name filename**

**Umask:**

Default umask value for the root - 0022

Default umask value for the normal user – 0002



**ACL**

Setfacl:

**setfacl -m u:user\_name:rwx filename** -- to update special permission for any specific user on any specific user

**setfacl -m g:group:rwx filename**  -- to update special permission for any specific group on any specific group

Getfacl filename:

getfacl filename -- to get the file and respective permission details in linux

**\*\*\*sticky bit\*\*\***

**Sticky Bit** is mainly used on folders in order to avoid deletion of a folder and its content by other users though they having write permissions on the folder contents. If Sticky bit is enabled on a folder, the folder contents are deleted by only owner who created them and the root user. No one else can delete other users’ data in this folder

Command to set the sticky bit:

**#chmod +t dir\_name**  --- to set the sticky bit on the directory

**Find command:**