**Advance Linux Process Management(Ubuntu,Centos,OpenSuse)**

There are some advance process monitoring tools for Linux Operating system.

Some of them are explaining bellow

**htop:**

Most system administrator familiar with Linux have used the top command line utility to see what process is taking the most CPU or memory. There’s a similar utility called htop that is much easier to use for normal tasks. It’s interactive, real-time and most importantly its very user friendly and you can see the CPU utilization at a glance.

But to use the htop utility we have to install it first. Because By default it is not installed in the operating system

**Installing Process of htop in linux(with Different Package management):**

**Ubuntu:**

**=> sudo apt install htop**

**Centos:**

for installing in centos we just need to add an EPEL repository so yum can find it.

**=>sudo yum -y install epel-release**

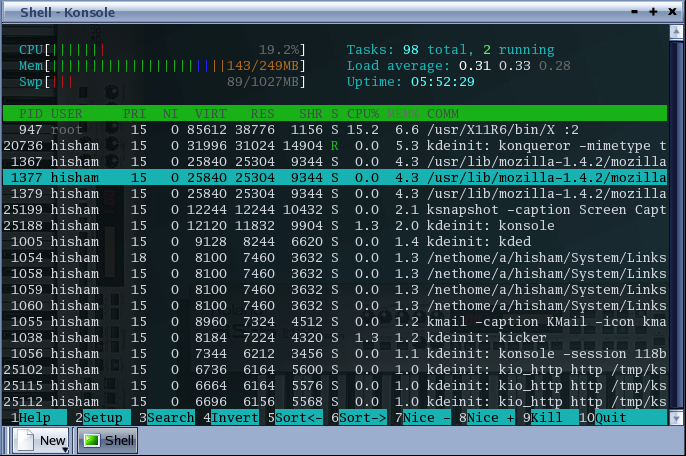
**=>sudo yum -y update**

**=>sudo yum install htop**

After a successful install we have to type

**=>sudo htop**

we should see the status of your system

****

Its almost look like top command but more interactive and more user friendly.

Lets talk about each option about the htop utility.

1) First option is the CPU which shows us the CPU utilization percentage and also in a graph mode

2) Second option is the memory option Which shows the actual memory which is used .

3) Third Option is the Swap space that is used by the system

4) Next option on the right portion is Task. It show the total ,Threads and the Running task

5) Next option shows the Load average of the system

6) Third option shows on the right shows the Uptime of the server shows the amount of time server is running

The lower Part provides details of the process just like top command lets see it again

|  |  |
| --- | --- |
| **Name Of The Header** | **Description** |
| **PID** | Every Process has a unique process id (the so called PID).the process id is very important. For example you want to kill a process then you need to provide the process id for that |
| **USER** | The name of the users the process is using .many process are run as root so you can see it quite often |
| **PRI** | It shows the priority of the process. This number is an indication that when the process will get the CPU cycles again. Lower the value higher the priority. Process with a higher priority will have the CPU cycle sooner. And lower priority process get the CPU cycle later |
| **NI** | The NICE value of the process .With the Help of the NICE value we can change the process Priority |
| **VIRT** | Total amount of Memory claimed by the process |
| **RES** | The memory size that the process is using at that moment |
| **SHR** | The amount of Shared memory that the process is sharing with other processed |
| **S** | Shows the status of the processed  ‘R’ means it is running  ‘S’ means it is in sleeping mode  ‘Z’ means its a zombie process  ‘T’ means stopped, either by a job control signal  ‘D’ means uninterruptible sleep |
| **%CPU** | The amount of CPU that is used by the last pooling cycle (which is typically 5 seconds ) |
| **%MEM** | The amount of MEMORY that is used by the last pooling cycle (which is typically 5 seconds ) |
| **TIME** | It indicates the total amount of CPU time that the process has used since it was started |
| **COMMAND** | This is the command that started the processed |

the most useful option is the option on the bottom .There are 10 option on the bottom of the screen

|  |  |
| --- | --- |
| **Name** | **Description** |
| **F1** | Its the help option. it contains the descriptions of every other option and short codes |
| **F2** | Setup option with this option you can customize the appearance of the htop utility .you can also set the color of the output and your desired option with this option. you can set which column should be there and which column should not |
| **F3** | With this option you can search a particular process just type F3 and the name of the process to find it. |
| **F4** | You can filter the process with this command. if you write a process name and it will show all the process s name with the same command name |
| **F6** | F6 is the sort option .you can sort the process by different options. you can sort the process by PID,USER,Priority,Time etc |
| **F7** | F7 is used to decrease the Nice value of any process the low the Nice value the greater the priority |
| **F8** | F8 is used to increase the Nice value of any process the higher the Nice value the lower the priority |
| **F9** | Its the kill command you select a process and press F9 it will show you a list of signal the you want to send to that process. That's how you can kill any process |
| **F10** | Exit command for htop |

You can also find the process filtered bu user from the commands just like we use like top command.

**=>htop -u <username>**

**Fuser:**

The fuser command is basically used to identify processes using files, directories, or sockets. The tool basically displays the PIDs of processes that are using the file whose name is passed as argument to the command. Suppose you are given a task to identify the processes that are using a particular file,’fuser’ command lets you identify processes based on the files (or directories, or sockets) they are accessing. For block special devices, the command lists the processes that use any file on that device.Not only that, the tool also allows you to kill these processes, so you don't have to use the kill or killall commands separately.

Fuser command output displays a list of PID of process followed by a letter indicating how the process use the file. cause the fuser command not only displays the process but also the type of access the as well.

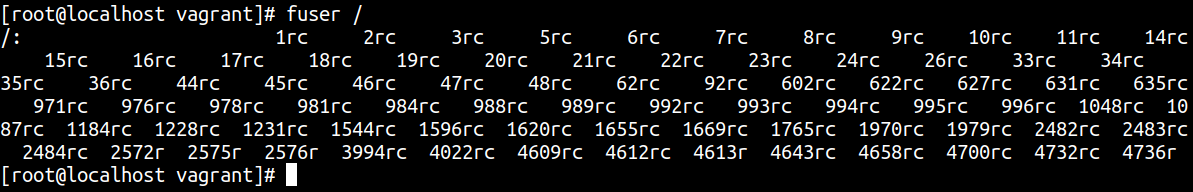
Each type of access denoted by a letter

|  |  |
| --- | --- |
| **item** | **Description** |
| **c** | Uses the file as a current directory. |
| **e** | Uses the file as as a programs executable object. |
| **r** | Uses the file as the root directory |
| **m** | Uses the file as a shared library (or other loadable object) |

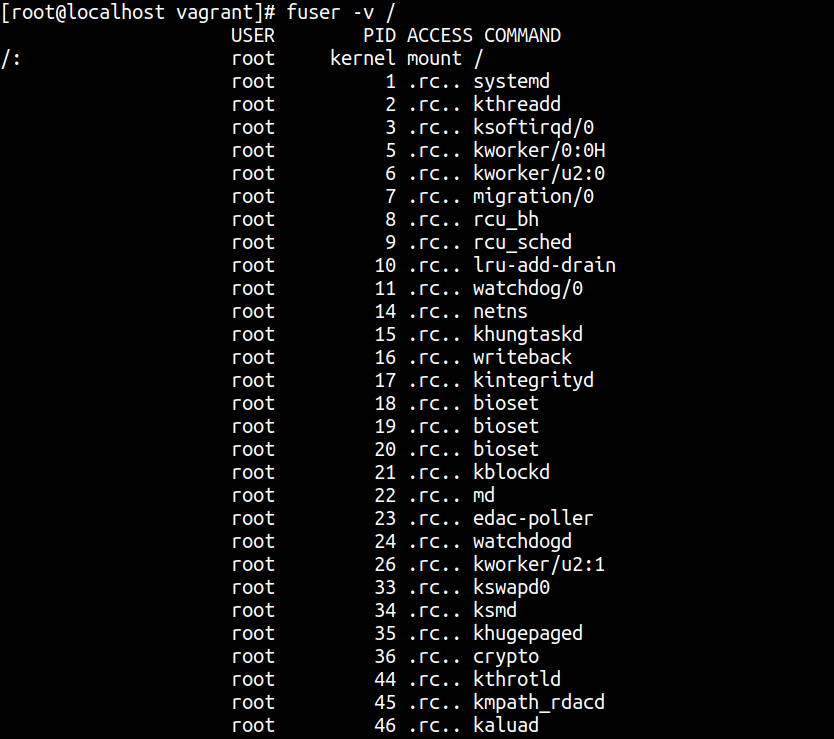
[Remember Linux consider everything as a file]

Suppose you want to see which process is currently using the root directory

**=>fuser /**



but this is only showing the PID and its hard to understand .so we add verbose flag (- v ) lets the result now



[killing process]

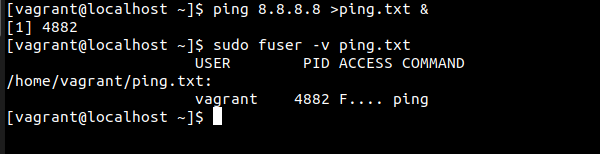
suppose you want to know which process is using a specific file.for example create a file ping.txt and

store the output of the ping [www.google.com](http://www.google.com/)

**=>pi ng** [**www.google.com**](http://www.google.com/) **> ping.txt &**

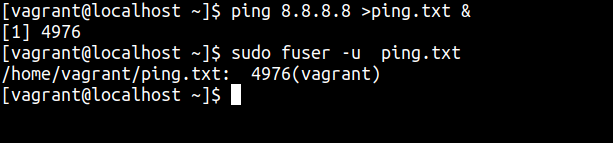
So now process created y the ping command is currently using this file lets check with the fuser command

**=> fuser -v ping.txt**



to list the process number and user login names of process .The -u flag is username

**=> fuser -u ping.txt**



like top or htop command we can also send the kill signal to the process that are currently using the process. Then you have to use the -k switch with the command.

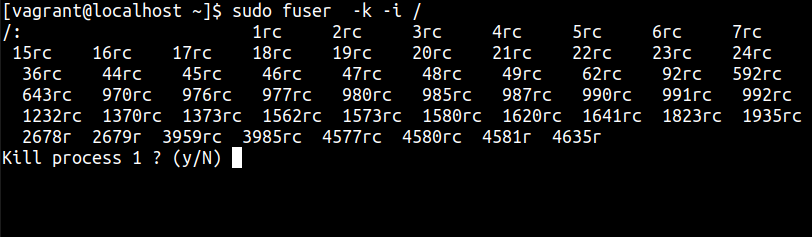
=>sudo fuser -k <filesystem>

To terminate all of the processes using a given file system, enter:

=>sudo fuser -kxuc /dev/hd1

if you want to kill the process interactively then you have to add -i switches

=>fuser -v -k -i <filesystem>



[The fuser command is used to determine the processes that are using a file system. If the file system is a network file system (NFS) and the NFS server is not responding, the fuser command might hang. To avoid such a situation, you can set the FUSER\_VERSION environment variable to 1.]

nohup:

Basically when you logout of the system all the process under this user will terminate but There is a command called nohup which executes another command and force the system to continue running it even the session the disconnected. nohup prevents the system from being aborted automatically when a user logout

**=>nohup <command> <command argument>**

There are some important properties of nohup command

1)The nohup command redirects the **standard input** to ***/dev/null*** therefore terminal input is not possible when running command using nohup

2)**Standard output** will be redirected to a file called **nohup.out** .So all the result of that command will be logged to this file

3)And **standard error** will be redirect to the terminal.

You can also the output to any file you want by redirecting the output to a file

**=>nohup command > file**