

# Linux Process Management(Ubuntu,Centos,OpenSuse)

---

Everything we do in linux OS is handled by a process. There are different commands for managing the process how to start and stop this process. How to run process in the foreground and Background and how to customize the process and how to schedule this process with 'corn' and 'anacorn' for future execution. Process management and Process Monitoring is a way of increasing and optimizing the performance of the server.

## Different Kinds of Process:

---

Linux basically has two kinds of Process

- 1) Automatic Process (Background Process)
- 2) Interactive Process

### 1)Automatic Process:

---

Automatic Process also known as Daemons. Automatic Process starts when the system started.(When the server is booted).This Process is not under the direct control of the User .In other it do not write output directly to the standard output

### 2) Interactive Process:

---

interactive process is the processes that started by the user using different commands. the process starts in a shell and it write the output directly to the standard output .To start an interactive process the user have to type command in a shell. and the process is started as a child process from the shell in which the user entered the command. when we terminate the child process this process give a exit status to the parent process and then it safely exit. Bu of the parent process died then it will not possible for monitoring .its called a zombie process .zombie Process is a result of the bad Programming. The systemmd process is the first process which starts other process is a child process of the process.

## Some Background Process (Daemons):

---

Process Name(Daemons)	Descriptions
<b>systemmd</b>	<p>Systemmd is the The Unix program which spawns all other processes .Which replaced the Init process after 2016 in Most of the Linux Operating system. systemmd is the top process of all the process trees.If we open a terminal and see the process trees with ‘pstree’ command.</p> <p>Installing packages for pstree commands</p> <p><b>ubuntu : sudo apt-get install psmisc</b> <b>centos : sudo yum install psmisc</b> <b>suse : sudo zypper install psmisc</b></p>
<b>cornd</b>	<p>Cornd is a job scheduler program .this program is used for schedule jobs which can be commands and shell scripts that can run periodically after time interval. It is typically used for system automation .its is very useful for downloading file from internet or downloading or sending email after a fixed time interval</p>
<b>dhcpcd</b>	<p>Its the dynamic host configuration protocol daemons .this program works as a background and automatically set the TCP/IP information to the client computer.</p>
<b>ftpd</b>	<p>This program handle the file server program. it handles the ftp request coming from the user</p>
<b>Httpd</b>	<p>Httpd is the web server daemons which handles the web server requests</p>
<b>sshd</b>	<p>Sshd the secure shell program. it handles the ssh requests from the users.</p>
<b>nfsd</b>	<p>Nfsd handles the requests of the of the user for nfs operation(nfs stands for network file system)</p>
<b>Sendmail</b>	<p>STMP daemon .it handles the STMP requests</p>

**fingerd** Provides a network interface for the finger protocol, Finger command looks up and displays information about system users.

**Ubuntu:** `sudo apt-get install finger`

**Centos:** `sudo yum install finger`

**suse:** `sudo zypper install finger`

**syslogd** System logger process that collects various system messages.

**ntpd** Ntpd is the Network Time Protocol Daemon . It manages clock synchronization across the network.

---

```
[vagrant@localhost ~]$ pstree
systemd-+-NetworkManager-+-dhclient
        |                  +-2*[{NetworkManager}]
        |
        | -agetty
        | -auditd---{auditd}
        | -chronyd
        | -crond
        | -dbus-daemon---{dbus-daemon}
        | -gssproxy---5*[{gssproxy}]
        | -master-+-pickup
        |           +-qmgr
        | -polkitd---6*[{polkitd}]
        | -rpcbind
        | -rsyslogd---2*[{rsyslogd}]
        | -sshd---sshd---sshd---bash---pstree
        | -systemd-journal
        | -systemd-logind
        | -systemd-udevd
        +-tuned---4*[{tuned}]
[vagrant@localhost ~]$
```

command `ps tree`

In a server the Daemon process is more important than a interactive process.it runs on the background and typically don't send any output to the .we have to check the log files to see what they are doing.

To see the daemon output we should see the '`/var/log/messages`' file. Because the daemons write their output to this file.

## Foreground and Background Process:

---

Basically interactive process is the foreground process and the Daemons are the background process. But sometimes we can send the foreground process to the Background.

To understand this thing .its Important to understand thre thing

- 1) **Standard input(STDIN)**
- 2) **Standard Output (STDOUT)**
- 3) **Standard Error (STDERR)**

when a process run in the Foreground

- 1)keyboard is considered as a standard input (STDIN)
- 2)Terminal is considered as a Standard output (STDOUT) and Standard error (STDERR)

but if we send a command to the background process this three things remain the same. That means if we run a program send a program in the background we can still see the output and the error(if happens) in the terminal. If we dont want The terminal output we can redirect this output to a file by using this symbol '>'.Like

command > /somewhere

for example if we write

=> **ls > output.txt**

**[pic will be added]**

it will write the output of that command in the output.txt file inside the current directory.

## How to send a Process to Background:

---

There are two easy way to send any process to background

- 1) Putting '&' sign at the end of any commands
- 2) Using 'bg' command after interrupting the running foreground processes

1)suppose we want run the 'ping 8.8.8.8 > out.txt' command in the background we put a '&' sign at the end of the command

=>ping 8.8.8.8 > out.txt &

[pic will be added]

2) when a program is running in the Foreground we interrupt the program using 'CTRL+Z ' and then we use the 'bg' command to resume the program in the background

```
[vagrant@localhost ~]$  
[vagrant@localhost ~]$ ping 8.8.8.8 > out1.txt  
^Z  
[1]+  Stopped                  ping 8.8.8.8 > out1.txt  
[vagrant@localhost ~]$ bg  
[1]+ ping 8.8.8.8 > out1.txt &  
[vagrant@localhost ~]$  
[vagrant@localhost ~]$ jobs  
[1]+  Running                  ping 8.8.8.8 > out1.txt &  
[vagrant@localhost ~]$
```

and after sending the program to the background if we actually want to see their activity ,we use the 'jobs' command to see their running.

```
[vagrant@localhost ~]$ ping 8.8.8.8 > out1.txt &  
[1] 23599  
[vagrant@localhost ~]$ ping 8.8.4.4 > out2.txt &  
[2] 23600  
[vagrant@localhost ~]$ jobs  
[1]-  Running                  ping 8.8.8.8 > out1.txt &  
[2]+  Running                  ping 8.8.4.4 > out2.txt &  
[vagrant@localhost ~]$
```

**Bringing the processes from Background to foreground:**

---

For bringing any process for background to the foreground we use the 'fg' command. If we just enter fg it will bring the last background job to foreground. to bring a specific job to foreground we use the process id after the 'fg' command.

For example

=>fg 1

=>fg 2

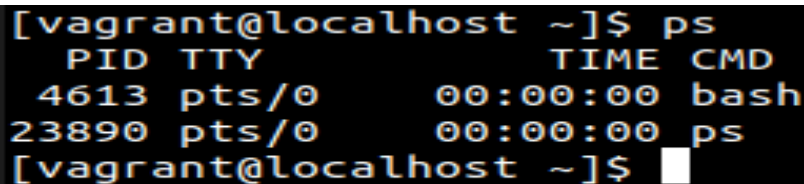
## Process management:

---

There are a lot of good performance monitoring tools for linux Operating system. Some of them are explaining bellow

### ps :

The most used and easiest command that is related to the process management is the '**ps**' command. with ps command you can view the list of process that is currently running on your system. it will show the process name and the process id(PID);



```
[vagrant@localhost ~]$ ps
  PID TTY          TIME CMD
 4613 pts/0        00:00:00 bash
23890 pts/0        00:00:00 ps
[vagrant@localhost ~]$
```

with the ps command we can try different switches like if we use the 'ps -e' new can see all the process in the system. or you can see one by one using this command

**ps -e | more**

```
[vagrant@localhost ~]$ ps -e | more
PID TTY          TIME CMD
  1 ?           00:00:03 systemd
  2 ?           00:00:00 kthreadd
  3 ?           00:00:00 ksoftirqd/0
  5 ?           00:00:00 kworker/0:0H
  6 ?           00:00:00 kworker/u2:0
  7 ?           00:00:00 migration/0
  8 ?           00:00:00 rcu_bh
  9 ?           00:00:02 rcu_sched
 10 ?           00:00:00 lru-add-drain
 11 ?           00:00:00 watchdog/0
 13 ?           00:00:00 kdevtmpfs
 14 ?           00:00:00 netns
 15 ?           00:00:00 khungtaskd
 16 ?           00:00:00 writeback
 17 ?           00:00:00 kintegrityd
 18 ?           00:00:00 bioset
```

if we want to list the output with a full format we should use this commands

‘ps -ef’.

```
[vagrant@localhost ~]$ ps -ef | more
UID          PID    PPID  C STIME TTY          TIME CMD
root          1      0  0 02:19 ?           00:00:03 /usr/lib/systemd/systemd --switc
hed-root --system --deserialize 21
root          2      0  0 02:19 ?           00:00:00 [kthreadd]
root          3      2  0 02:19 ?           00:00:00 [ksoftirqd/0]
root          5      2  0 02:19 ?           00:00:00 [kworker/0:0H]
root          6      2  0 02:19 ?           00:00:00 [kworker/u2:0]
root          7      2  0 02:19 ?           00:00:00 [migration/0]
root          8      2  0 02:19 ?           00:00:00 [rcu_bh]
root          9      2  0 02:19 ?           00:00:02 [rcu_sched]
root         10      2  0 02:19 ?           00:00:00 [lru-add-drain]
root         11      2  0 02:19 ?           00:00:00 [watchdog/0]
root         13      2  0 02:19 ?           00:00:00 [kdevtmpfs]
root         14      2  0 02:19 ?           00:00:00 [netns]
root         15      2  0 02:19 ?           00:00:00 [khungtaskd]
root         16      2  0 02:19 ?           00:00:00 [writeback]
root         17      2  0 02:19 ?           00:00:00 [kintegrityd]
root         18      2  0 02:19 ?           00:00:00 [bioset]
```

Or you can use the ps -el

you can also use the

=>ps -ax

or

=>**ps -aux**

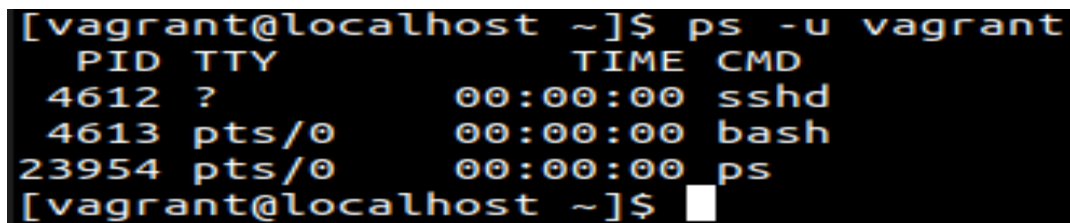
this two command will show the process list in BSD and long BSD format respectively .

### **Find processes by Users using ps command:**

---

It is possible to find process by Users using ps command.so as an administrator you can monitor what other user are doing .

=>**ps -u <username>**



```
[vagrant@localhost ~]$ ps -u vagrant
  PID TTY          TIME CMD
  4612 ?            00:00:00 sshd
  4613 pts/0        00:00:00 bash
 23954 pts/0        00:00:00 ps
[vagrant@localhost ~]$
```

In this picture there are Three process are shown. One of them is sshd cause I am actually connected to this virtual machine using ssh connectivity.

### **uptime:**

---

uptime command gives us information about how long the server is up and gives details about the load-average as well

this output starts with

- 1) current time
- 2) up time ( how long the server is up and running)
- 3) currently logged user into this sever
- 4) load average

the last one is the most important parameter .it shows three different numbers

first number	load average for the last minuets
Second number	load average for the last 5 minutes
Third number	load average for the last 15 minutes



the load averages is displayed by a number that indicates the current activity of the process queue. the value actually indicates the number of the process queues that are waiting to be handled by the CPU of your system.

We can get some insight of the system by reading this number  
if the number is 1 ,it means the CPU is fullu occupied but there is no process waiting in the queue

if it is more than 1 then it has a list of process that a lining up that have to be processed .In this case the User can experience some delays .but it is difficult to say anu critical value because it depends on the server hardware configuration .typically 1 is considered as an Ideal number .if the server has dual core or two cupu then the ideal number will be 2 .and if the server is hyper-threading enabled with 32 CPU then the ideal number will be 64.

```
[vagrant@localhost ~]$ uptime
06:33:58 up 4:14, 1 user, load average: 0.00, 0.01, 0.05
[vagrant@localhost ~]$
```

## free:

---

with Free command the server give you information about the current physical RAM and swap space. The less swap space is user the better cause swapping is bad. Because swapping is basically use the physical space to compensate the lack of physical memory which is extremely slower then the actual RAM

There are also cache memory and the buffer memory

Cache Memory: The memory that can be freed instantaneously for process

buffer Memory: The memory is the memory used by the processes and cant be freed without terminating the process .

```
[vagrant@localhost ~]$ free
              total        used        free      shared  buff/cache   available
Mem:          1014972       76908       636732         6792       301332       765736
Swap:         2097148           0       2097148
[vagrant@localhost ~]$
```

top:

The most useful and most used command that gives you nearly all the information is the top command.

```
top - 06:46:41 up 4:26, 1 user, load average: 0.15, 0.05, 0.06
Tasks: 79 total, 1 running, 78 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.3 us, 0.3 sy, 0.0 ni, 99.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 1014972 total, 635976 free, 77348 used, 301648 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 765272 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3	root	20	0	0	0	0	S	0.3	0.0	0:00.79	ksoftirqd/0
1	root	20	0	127972	6484	4096	S	0.0	0.6	0:03.43	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
6	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kworker/u2:0
7	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	20	0	0	0	0	S	0.0	0.0	0:03.00	rcu_sched
10	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	lru-add-drain
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.28	watchdog/0
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
14	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	netns
15	root	20	0	0	0	0	S	0.0	0.0	0:00.01	khungtaskd
16	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	writeback
17	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kintegrityd
18	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	bioaset
19	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	bioaset
20	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	bioaset

Lets break it down row by row:

**First row** : The first row of the output shows the exact output of the uptime commands

**Second row** : The second row shows the number of total task, then the number of running task then number of task that are sleeping mode and the last one is the zombie process. (zombie process is the process that is stopped but unable to give the exit status to its parent process)

**Third row** : third row consists of these values

Name Of The Header	Description
<b>us</b>	us stands for user space .it represents the CPU activity in user space .this activity is actually started by the different commands of the username
<b>sy</b>	it represent the CPU activity in system space. This are actually kernel routine. They often conduct their work on behalf the Daemons.
<b>ni</b>	ni indicates the amount of time that are spent by processing the low priority process
<b>id</b>	CPU inactivity .High value actually shows that system is doing nothing
<b>wa</b>	It indicates the amount of time the CPU is waiting for the input (I/O hardware that are connected to your system like hard disk,keyboard ,mouse)
<b>hi</b>	he amount of time that the CPU spent for communicating with the hardware.for example if you read data from a flash drive then at that time this value will be high

**si**

the amount of time that the CPU spent for communicating with the software .normally it should be low

**st**

This parameter indicate the amount of time that is stolen by the Virtualization hypervisor from a virtual machine. If your system has no virtual machine this value will be 0

---

**Fourth row:** It shows the exact output of the ‘free’ command (memory statistics of the current system)

**Fifth row:** This part is the lower part of the top window. It provides details about a process That is most the most active in terms of CPU usages

Name Of The Header	Description
<b>PID</b>	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
<b>USER</b>	The name of the users the process is using .many process are run as root so you can see it quite often
<b>PR</b>	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

<b>NI</b>	The NICE value of the process .With the Help of the NICE value we can change the process Priority
<b>VIRT</b>	Total amount of Memory claimed by the process
<b>RES</b>	The memory size that the process is using at that moment
<b>SHR</b>	The amount of Shared memory that the process is sharing with other processed
<b>S</b>	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
<b>%CPU</b>	The amount of CPU that is used by the last pooling cycle (which is typically 5 seconds )
<b>%MEM</b>	The amount of MEMORY that is used by the last pooling cycle (which is typically 5 seconds )
<b>TIME</b>	It indicates the total amount of CPU time that the process has used since it was started
<b>COMMAND</b>	This is the command that started the processed

---

## Command for Killing Process:

---

There are different commands for terminating processes

Process Termination command	Description
<b>Kill</b>	It is the most commonly used with a numerical argument (SIGKILL) if no signal is referred the default signal (15) is sent to the process
<b>killall</b>	If we want to kill more than one process then we can use killall command. for example if we use killall httpd then it will kill all the instances of the Apache server.
<b>top</b>	We can kill a process using top command. from the top interface press 'k'. you will be asked the PID of the process Enter it then you have to enter the signal to send the process. specify the numerical value it will terminate the process
<b>Pkill</b>	<p>Pkill is the command for terminating process based on other information of the process. it allows the administrator to find process by its details for example</p> <p><b>'pkill -U 501'</b> will kill all the process owned by the user 501</p>