MANAGING PROCESS

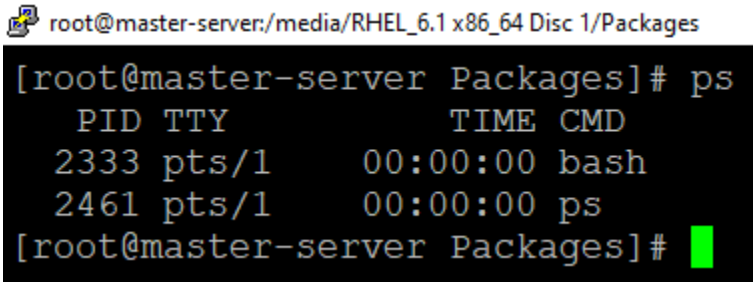
- A Linux process is a program running in the Linux system. Depending on Linux distributions, it's also known as service. In Linux community however, a Linux process is called daemon.
- When you start a program or running an application in Linux, you actually execute that program. A Linux process (a daemon), running in foreground or in the background, uses memory and CPU resources. That's why we need to manage Linux process. Keeping unused Linux process running in the system is a waste and also exposes your system to security threat.
- In Linux, every running process or daemon is given an identity number called PID (Process ID). The process id is unique. We can terminate unused program in the system by stopping its process id.

Parent and Child Process:

- The Process which starts or creates another process is called parent process and the one which got created is known as child process.
- Every process will be having a parent process except init process.
- The init process is the parent of all the process in the system. It is the first process which gets started by the kernel at the time of booting
- The PID of init will be 1.
- Only after init process gets started the remaining process are called by it, and hence it is responsible for all the remaining processes in the system.

To monitor the process using ps command

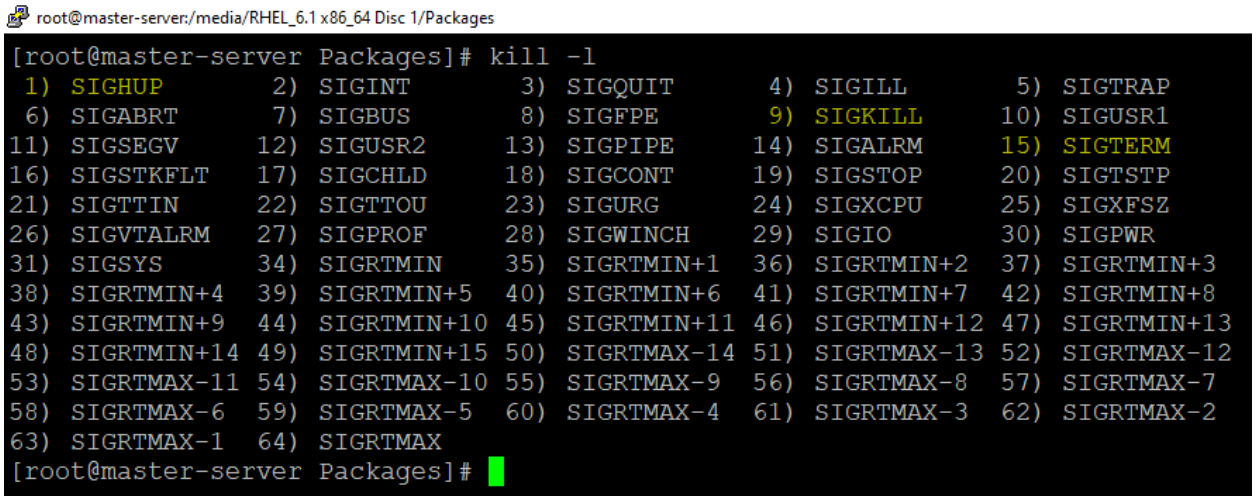
- The ps command gives the running process of the present terminal and present command. The syntax for ps command is
- #ps



- To see total number of processes running in the system
- #ps –a

Signals in Linux:

- ➔ Signals are a way of sending simple messages to processes. Most of these messages are already defined and can be found in . However, signals can only be processed when the process is in user mode. If a signal has been sent to a process that is in kernel mode, it is dealt with immediately on returning to user mode.
- ➔ There are total 64 signals in Linux, the list of all the signal can be sen by
- ➔ #kill –l



Setting up the Priority of a Process

- ➔ In Linux we can set guidelines for the CPU to follow when it is looking at all the tasks it has to do. These guidelines are called niceness or nice value. The Linux niceness scale goes from -20 to 19. The lower the number the more priority that task gets. If the niceness value is high number like 19 the task will be set to the lowest priority and the CPU will process it whenever it gets a chance. The default nice value is zero.
- ➔ To schedule a priority of a process before starting it
- ➔ To set a priority to a process before starting it, the syntax is
- ➔ #nice –n <nice value range(-20 to 19) <command>

Monitoring the process using top command

- ➔ When you need to see the running processes on your Linux in real time, you have top as your tool for that.

Monitoring all process using top command

To monitor all processes in the system use the following command

#top

#top

root@master-server:/media/RHEL_6.1 x86_64 Disc 1/Packages

```
top - 19:35:34 up 52 min,  3 users,  load average: 0.00, 0.00, 0.00
Tasks: 179 total,   1 running, 178 sleeping,   0 stopped,   0 zombie
Cpu(s):  1.0%us,  1.3%sy,  0.0%ni, 97.2%id,  0.2%wa,  0.1%hi,  0.2%si,  0.0%st
Mem:   2040528k total,   700476k used,  1340052k free,    31824k buffers
Swap:  4095992k total,        0k used,  4095992k free,   293252k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 2467 root      20   0 15008 1224  876 R  1.9  0.1   0:00.03 top
    1 root      20   0 19336 1508 1212 S  0.0  0.1   0:03.06 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.02 kthreadd
```

The first line in top:

```
top - 02:23:18 up 1 day, 13:57,  3 users,  load average: 0.01, 0.00, 0.23
```

- **"02:23:18 "** is the current time; **"up 1 day"** shows how long the system has been up for; **"3 user"** how many users are logged in; "**load average: 0.01, 0.00, 0.23 "** the load average of the system (1minute, 5 minutes, 15 minutes).

The second line in top:

```
Tasks: 273 total,   1 running, 272 sleeping,   0 stopped,   0 zombie
```

- Shows the number of processes and their current state.

The third line in top:

```
Cpu(s):  0.4%us,  0.5%sy,  0.0%ni, 98.8%id,  0.3%wa,  0.0%hi,  0.0%si,  0.0%st
```

- Shows CPU utilization details. "9.5%us" user processes are using 9.5%; "31.2%sy" system processes are using 31.2%; "27.0%id" percentage of available cpu; "7.6%wa" time CPU is waiting for IO.

The fourth and fifth lines in top:

```
Mem:    543948k total,   526204k used,    17744k free,    11748k buffers
Swap:  2097144k total,    49064k used,  2048080k free,   129928k cached
```

- **"543948k total"** is total memory in the system; **"526204K used"** is the part of the RAM that currently contains information; **"17744k free"** is the part of RAM that contains no information; **"17748K buffers and 129928k cached"** is the buffered and cached data for IO.

By default, top starts by showing the following task's property:

Field	Description
PID	Process ID
USER	Effective User ID
PR	Dynamic priority
NI	Nice value, also known as base priority
VIRT	Virtual Size of the task. This includes the size of process's executable binary, the data area and all the loaded shared libraries.
RES	The size of RAM currently consumed by the task. Swapped out portion of the task is not included.
SHR	Some memory areas could be shared between two or more task, this field reflects that shared areas. The example of shared area are shared library and SysV shared memory.
S	Task status
%CPU	The percentage of CPU time dedicated to run the task since the last top's screen update.
%MEM	The percentage of RAM currently consumed by the task.
TIME+	The total CPU time the task has been used since it started. "+" sign means it is displayed with hundredth of a second granularity. By default, TIME/TIME+ doesn't account the CPU time used by the task's dead children.
Command	Showing program names

Interacting with TOP:

Now that we are able to understand the output from TOP lets learn how to change the way the output is displayed.

Just press the following key while running top and the output will be sorted in real time.

- ➔ M – Sort by memory usage
- ➔ P – Sort by CPU usage
- ➔ T – Sort by cumulative time
- ➔ z – Color display
- ➔ k – Kill a process
- ➔ q – quit
- ➔ r – to renice a process
- ➔ h – help