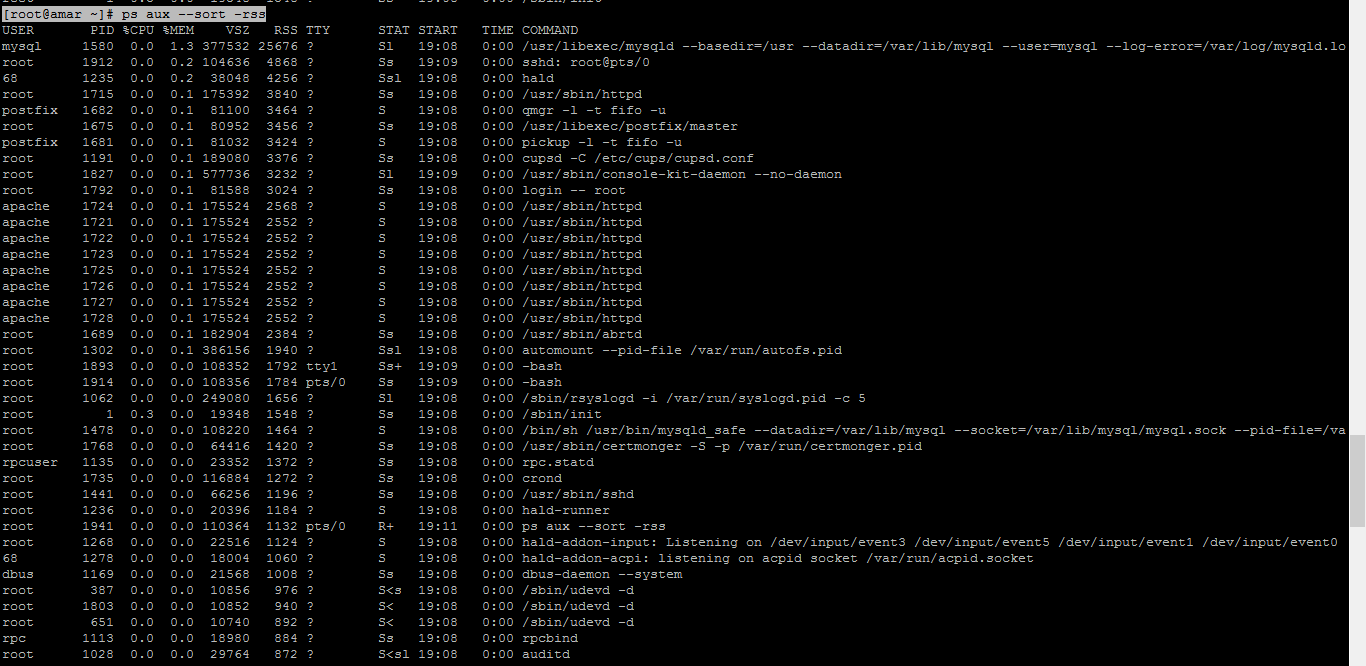
**Memory used by a process(RAM)**

Answer: -

[root@amar ~]# ps aux --sort -rss

That ps command gives me this output:



We can also use the below command

We’ll use ps, awk, head and sort with a pipe, to find out which application is consuming our RAM

[root@amar ~]# ps aux | awk '{print $2, $4, $11}' | sort -k2rn | head -n 20

**total number of open files by a process:**

**Answer :-**

[root@amar ~]# ps -aef | grep httpd



We can also use the be

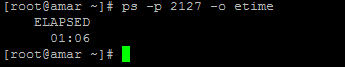
ps aux | grep httpd | awk '{ print $2 }'

**Running duration of a process**

**Ans:-**

**Here 2127 is the process id**

**[root@amar ~]# ps -p 2127 -o etime**



**What is file descriptor**

file descriptor is a number that uniquely identifies an open file in a computer's operating system. It describes a data resource, and how that resource may be accessed.

**How to kill a process**

Kill process forcefully

[root@amar ~]# kill -9 2269

Kill process gracefully

[root@amar ~]# kill -HUP 2269

Exmaple :-

**What are the signals**

pkill -u shs loopy

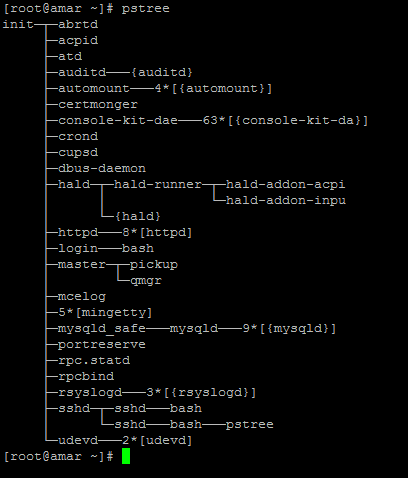
Whenever you generate a signal using any form of kill command, the operating system interrupts the target process' normal flow of execution. What happens next depends on the signal being sent.

The pkill and killall commands came about to make terminating process easier by identifying processes by name and other criteria. Going back to our “loopy” example, you could kill the process in one fell swoop, rather than looking up the process ID and then killing it. While the effect may be the same in both cases, the end result depends on whether other processes by the same name are running and whether you have sufficient access to terminate them.

What's easiest depends on what you're trying to do. Is your focus on one particular user, one particular process, or a wider range of users and processes? In general, using additional options allows your kill commands to be more targeted.

**What is parent process ID in linux**

Parent and Child Processes. Each unix process has two ID numbers assigned to it: The Process ID (pid) and **the Parent process ID (ppid**). Each user process in the system has a parent process. Most of the commands that you run have the shell as their parent.



**Print pid of current shell**

$$ is the PID of the current shell.



**How to clear a log file of running process**

**Step:-1**

The correct procedure is to move the logfile, then signal Apache to tell it to reopen the logfiles.

Apache is signalled using the SIGHUP (-1) signal. e.g.

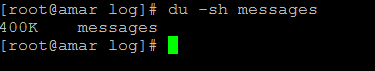
mv access\_log access\_log.old

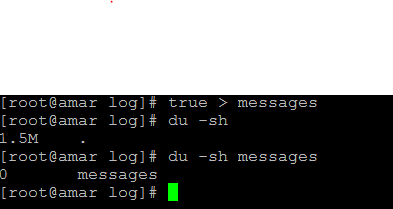


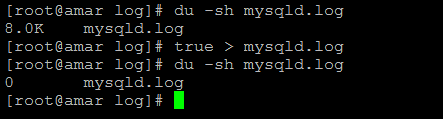
kill -1 `cat httpd.pid`

Note: httpd.pid is a file containing the process id of the Apache httpd daemon, Apache saves this in the same directory as the log files.

**Step:2**







When also use the **echo command** for the same

**What will happen if you delete a log file of running process**

**Example:- 1**

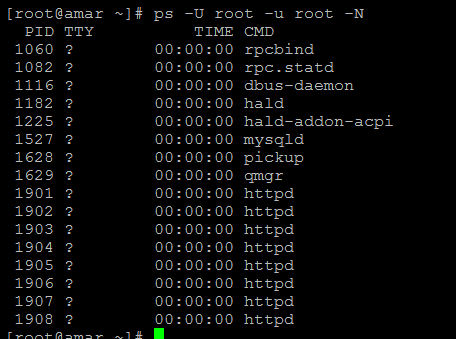
If we are using the httpd service and we have delete the access.log .

Application logs will stop working, Unless we will start the service of the application

**How do you check all the running process in the system?**

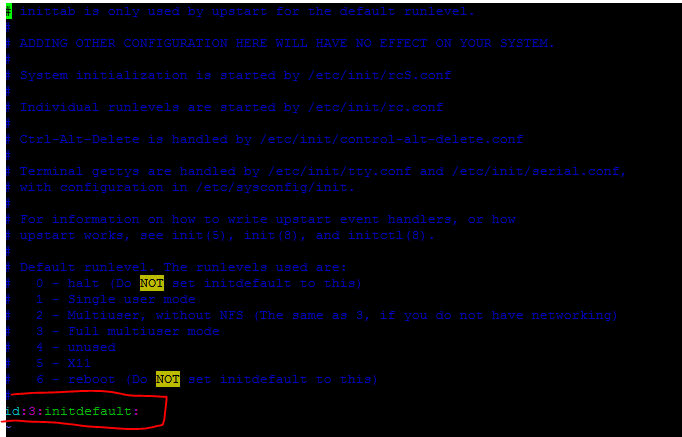
The **top** command display and update sorted information about processes.

We also use :- [root@amar ~]# ps -U root -u root -N



**What init process is responsible for**

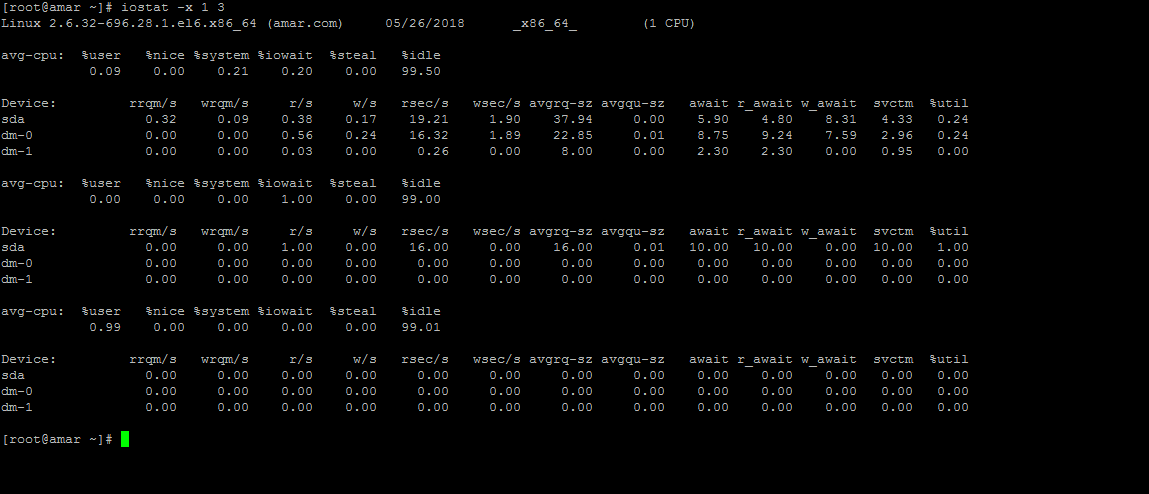
Executes the system to boot into the run level as specified in /etc/initab



**How do you check those process that are waiting for the resources**

[root@amar ~]# iostat -x 1 3

**Output come like**



**What are Running, Waiting, Stopped and Zombie processes.**

**Zombie Process: -** Zombies are DEAD processes. All processes eventually die, and when they do they become zombies. They consume almost no resources, which is to be expected because they are dead! The reason for zombies is so the zombie's parent (process) can retrieve the zombie's exit status and resource usage statistics. The parent signals the operating system that it no longer needs the zombie by using one of the wait() system calls.

**Stopped Processes: -**