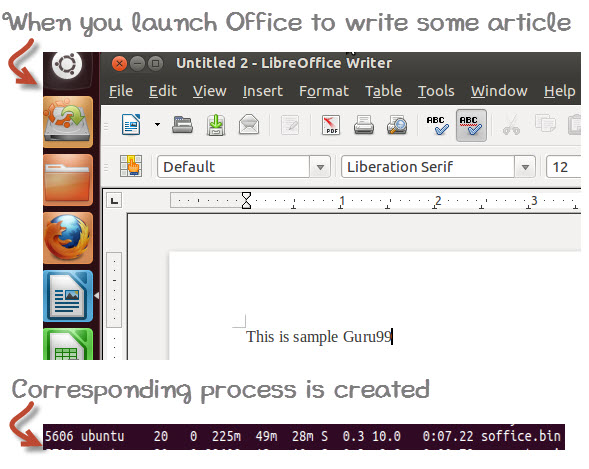
**Identify CPU/memory intensive processes, adjust process priority with renice, and kill processes.**

**System Activities**

## What is a Process?

An instance of a program is called a Process. In simple terms, any command that you give to your Linux machine starts a new process.

[](https://cdn.guru99.com/images/whatisprocessid.jpg)

Having multiple processes for the same program is possible.

Types of Processes:

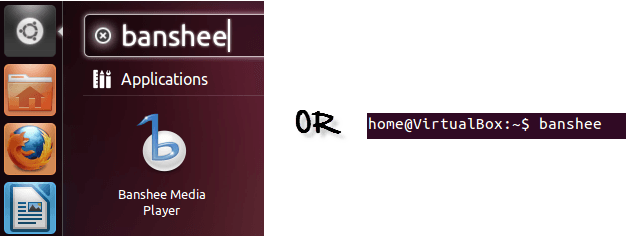
* Foreground Processes: They run on the screen and need input from the user. For example Office Programs
* Background Processes: They run in the background and usually do not need user input. For example Antivirus.

Please be patient. The Video will load in some time. If you still face issue viewing video click [here](https://www.guru99.com/faq.html#1)

## Running a Foreground Process

To start a foreground process, you can either run it from the dashboard, or you can run it from the terminal.

When using the Terminal, you will have to wait, until the foreground process runs.

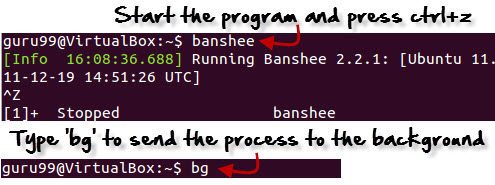
[](https://cdn.guru99.com/images/foreground.png)

## Running a Background process

If you start a foreground program/process from the terminal, then you cannot work on the terminal, till the program is up and running.

Particular, data-intensive tasks take lots of processing power and may even take hours to complete. You do not want your terminal to be held up for such a long time.

To avoid such a situation, you can run the program and send it to the background so that terminal remains available to you. Let's learn how to do this -

[](https://cdn.guru99.com/images/bg.jpg)

## Fg

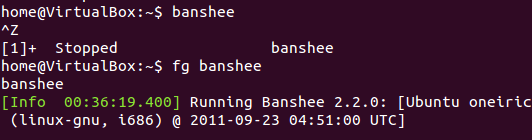
You can use the command "fg" to continue a program which was stopped and bring it to the foreground.

The simple syntax for this utility is:

fg jobname

Example

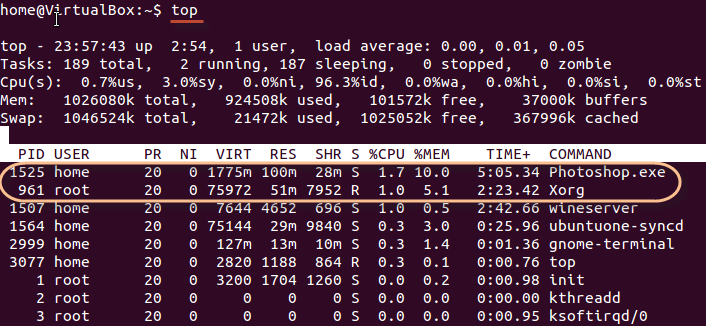
1. Launch 'banshee' music player
2. Stop it with the 'ctrl +z' command
3. Continue it with the 'fg' utility.

[](https://cdn.guru99.com/images/fg.png)

Let's look at other important commands to manage processes -

## Top

This utility tells the user about all the running processes on the Linux machine.

[](https://cdn.guru99.com/images/top.png)

Press 'q' on the keyboard to move out of the process display.

The terminology follows:

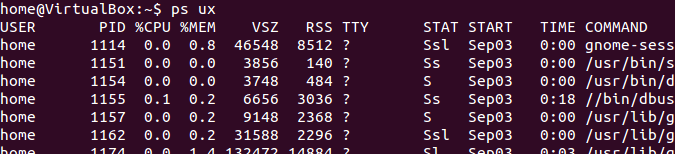
| **Field** | **Description** | **Example 1** | **Example 2** |
| --- | --- | --- | --- |
| PID | The process ID of each task | 1525 | 961 |
| User | The username of task owner | Home | Root |
| PR | Priority Can be 20(highest) or -20(lowest) | 20 | 20 |
| NI | The nice value of a task | 0 | 0 |
| VIRT | Virtual memory used (kb) | 1775 | 75972 |
| RES | Physical memory used (kb) | 100 | 51 |
| SHR | Shared memory used (kb) | 28 | 7952 |
| S | Status  There are five types:            'D' = uninterruptible sleep            'R' = running            'S' = sleeping            'T' = traced or stopped            'Z' = zombie | S | R |
| %CPU | % of CPU time | 1.7 | 1.0 |
| %MEM | Physical memory used | 10 | 5.1 |
| TIME+ | Total CPU time | 5:05.34 | 2:23.42 |
| Command | Command name | Photoshop.exe | Xorg |

## PS

This command stands for 'Process Status'. It is similar to the "Task Manager" that pop-ups in a Windows Machine when we use Cntrl+Alt+Del. This command is similar to 'top' command but the information displayed is different.

To check all the processes running under a user, use the command -

ps ux

[](https://cdn.guru99.com/images/ps.png)

You can also check the process status of a single process, use the syntax -

ps PID

[Managing Processes in Linux/Unix: top, ps, kill, df, free, nice](https://cdn.guru99.com/images/ps_pid.jpg)

## Kill

This command **terminates running processes** on a Linux machine.

To use these utilities you need to know the PID (process id) of the process you want to kill

Syntax -

kill PID

To find the PID of a process simply type

pidof Process name

Let us try it with an example.

[Managing Processes in Linux/Unix: top, ps, kill, df, free, nice](https://cdn.guru99.com/images/kill.png)

## NICE

Linux can run a lot of processes at a time, which can slow down the speed of some high priority processes and result in poor performance.

To avoid this, you can tell your machine to prioritize processes as per your requirements.

This priority is called Niceness in Linux, and it has a value between -20 to 19. The lower the Niceness index, the higher would be a priority given to that task.

The default value of all the processes is 0.

To start a process with a niceness value other than the default value use the following syntax

nice -n 'Nice value' process name

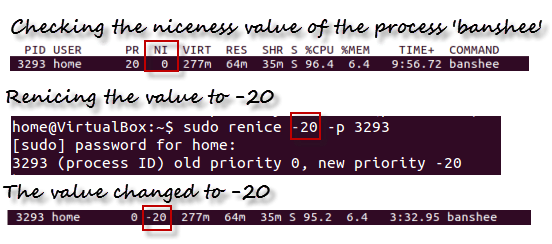
[Managing Processes in Linux/Unix: top, ps, kill, df, free, nice](https://cdn.guru99.com/images/changing_niceness.png)

If there is some process already running on the system, then you can 'Renice' its value using syntax.

renice 'nice value' -p 'PID'

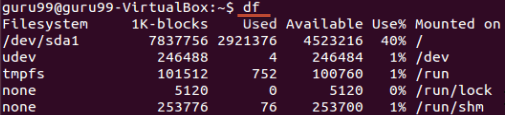
To change Niceness, you can use the 'top' command to determine the PID (process id) and its Nice value. Later use the renice command to change the value.

Let us understand this by an example.

[](https://cdn.guru99.com/images/renicing.png)

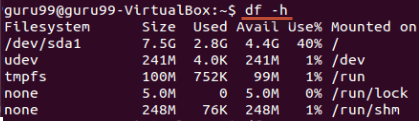
## DF

This utility reports the free disk space(Hard Disk) on all the file systems.

[](https://cdn.guru99.com/images/df.png)

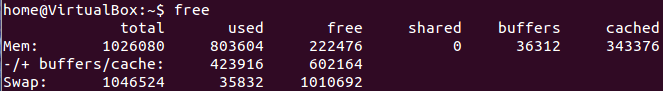
If you want the above information in a readable format, then use the command

'df -h'

[](https://cdn.guru99.com/images/df-h.png)

## Free

This command shows the free and used memory (RAM) on the Linux system.

[](https://cdn.guru99.com/images/free.png)

You can use the arguments

free -m to display output in MB

free -g to display output in GB

## Summary:

* Any running program or a command given to a Linux system is called a process
* A process could run in foreground or background
* The priority index of a process is called Nice in Linux. Its default value is 0, and it can vary between 20 to -19
* The lower the Niceness index, the higher would be priority given to that task

| **Command** | **Description** |
| --- | --- |
| **bg** | To send a process to the background |
| **fg** | To run a stopped process in the foreground |
| **top** | Details on all Active Processes |
| **ps** | Give the status of processes running for a user |
| **ps PID** | Gives the status of a particular process |
| **pidof** | Gives the Process ID (PID) of a process |
| **kill PID** | Kills a process |
| **nice** | Starts a process with a given priority |
| **renice** | Changes priority of an already running process |
| **df** | Gives free hard disk space on your system |
| **free** | Gives free RAM on your system |

To get an instantaneous image of a server activity (use ‘**virt-top**‘ on a **KVM** hypervisor), type:

# top

To get details about processes, type:

# ps -edf

**Process Priority**

To start a process (here **script.sh**) with a low priority, type:

# nice -n 10 ./script.sh

To change the priority (here **+5**) of an already running process, get its PID (Process ID) through top or ps (here **789**) and type:

# renice +5 789

**Alternatively**:

# renice +5 `pgrep script.sh`

**Process Deletion**

To kill the process, get its PID through top or ps (here **789**) and type:

# kill -9 789

**Alternatively**:

# pkill script.sh

**System Reporting**

To display details about IO activities, type:

# iostat

To show network card activities, type:

# netstat -i

To display socket activities, type:

# netstat -a

To get details about virtual memory activities (memory, swap, run queue, cpu usage, etc) every 5 second, type:

# vmstat 5

To get a full report of a server activity, type:

# sar -A

**Using the Linux ps utility**

This utility is used to output a snapshot of your current processes. ps displays information about various active processes. There are many flags you can use along with ps, below I will list a few examples.

To see every process on the system using standard syntax:  
ps -e  
ps -ef  
ps -eF  
ps -ely [Plain Text](http://www.devblog.co/linux-rhcsa-identify-cpu-memory-intensive-processes-adjust-process-priority-with-renice-and-kill-processes/)

To see every process on the system using BSD syntax:  
ps ax  
ps axu[Plain Text](http://www.devblog.co/linux-rhcsa-identify-cpu-memory-intensive-processes-adjust-process-priority-with-renice-and-kill-processes/)

You can find more examples using the man pages as such.

man ps[Plain Text](http://www.devblog.co/linux-rhcsa-identify-cpu-memory-intensive-processes-adjust-process-priority-with-renice-and-kill-processes/)

**Using the top utility**

The top utility displays a dynamic real-time view of the running tasks on your system. It also displays system summary information as far as CPU usage, memory usage, your uptime and more.

In order to use top simply type top in your terminal.

top[Plain Text](http://www.devblog.co/linux-rhcsa-identify-cpu-memory-intensive-processes-adjust-process-priority-with-renice-and-kill-processes/)

After top is running you will be able to interact with the program by using various commands. For instance if you needed to sort by a particular column you could use shift f. This would then display options for you to choose from for your sort.

You can also kill process or change their priority in top by using either the letter k or r. Top also has a help menu that you can access once you’re running the program. Type the ? key in order to access it.

**Using the kill utilities**

The kill utility can come in handy when you need to kill a particular process. You can call a process using the process id or the process name. There are 2 tools that you need to be familiar with, kill and killall.

**Using kill**

If you wanted to use kill to end a process based on the process id you would use the following.

kill 2012[Plain Text](http://www.devblog.co/linux-rhcsa-identify-cpu-memory-intensive-processes-adjust-process-priority-with-renice-and-kill-processes/)

The number 2012 you would replace with the process id of the application you want to kill. You could also kill a process based on the name. For instance, lets say we wanted to stop the process named firefox. We would use the following.

killall firefox[Plain Text](http://www.devblog.co/linux-rhcsa-identify-cpu-memory-intensive-processes-adjust-process-priority-with-renice-and-kill-processes/)

Both tools also offer various flags to handle certain processes in certain ways. For instance the -9 flag can be added to either kill or killall and it will kill the signal immediately.

**Using the renice utility.**

The renice utility allows you to prioritize your processes. This is helpful when you want to make sure a particular process finishes or runs compared to other processes. The range for the priorities are from -20 to 19. The lower the the number the higher the priority.

Note\* Only root can set the priority to less then 0.

In order to use renice you would do the following.

renice -20 process\_id[Plain Text](http://www.devblog.co/linux-rhcsa-identify-cpu-memory-intensive-processes-adjust-process-priority-with-renice-and-kill-processes/)

The above command sets the process with the highest priority of -20. You would replace process\_id with your actual process id.