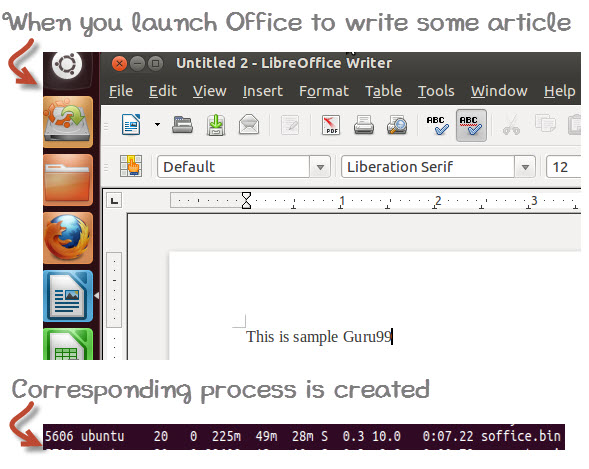
**What is a Process?**

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://www.guru99.com/images/whatisprocessid.jpg)

Having multiple processes for the same program is possible.

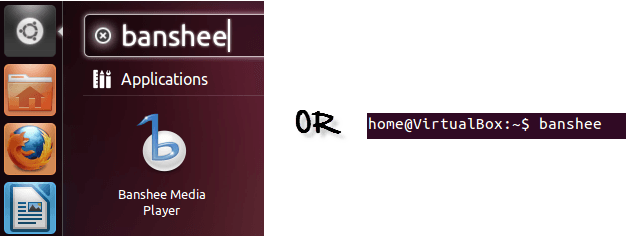
Types of Processes:

* 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.

**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.

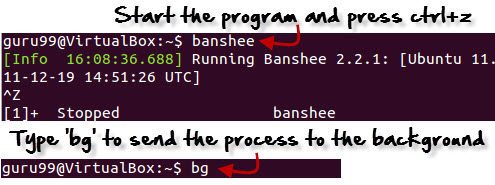
[](https://www.guru99.com/images/foreground.png)

**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://www.guru99.com/images/bg.jpg)

**Fg**

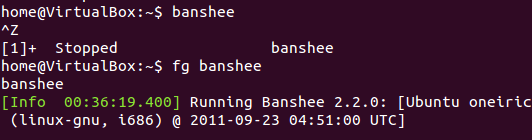
You can use the command "fg" to continue a program which was stopped and bring it to the foreground.

The simple syntax for this utility is:

fg jobname

Example

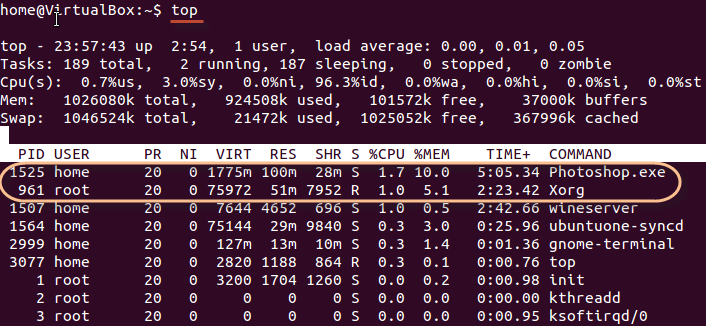
1. Launch 'banshee' music player
2. Stop it with the 'ctrl +z' command
3. Continue it with the 'fg' utility.

[](https://www.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://www.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 |

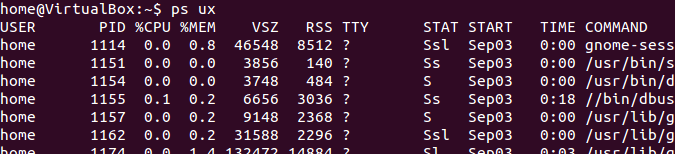
**PS**

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 -

[See More](https://eb2.3lift.com/pass?tl_clickthrough=true&redir=https%3A%2F%2Fad.doubleclick.net%2Fddm%2Ftrackclk%2FN1114924.2079317MEDIAMATHPROGRAM%2FB23910829.280282388%3Bdc_trk_aid%3D474475793%3Bdc_trk_cid%3D130179297%3Bu%3D8362036%7C6579238%7C938701497361349502%3Bkw%3D374f5f54-f2a8-4a00-a471-7f7e7505dbb6%3Bdc_lat%3D%3Bdc_rdid%3D%3Btag_for_child_directed_treatment%3D%3Btfua%3D%3Bgdpr%3D0%3Bgdpr_consent%3D%24%7BGDPR_CONSENT_755%7D&bc=1.592&pr=0.796&brid=66&bmid=3690&biid=4597&aid=131730879173817997890&tid=5952148&bcud=1592&sid=16982&ts=1603729022&cb=61706)

ps ux

[](https://www.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://www.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://www.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

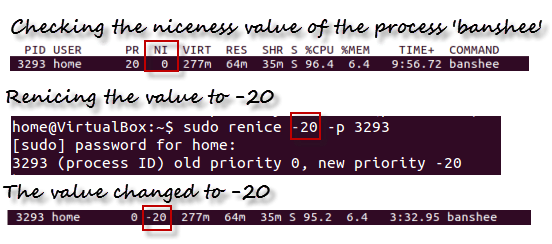
[Managing Processes in Linux/Unix: top, ps, kill, df, free, nice](https://www.guru99.com/images/changing_niceness.png)

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

renice 'nice value' -p 'PID'

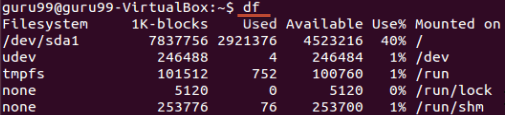
To change Niceness, you can use the 'top' command to determine the PID (process id) and its Nice value. Later use the renice command to change the value.

Let us understand this by an example.

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

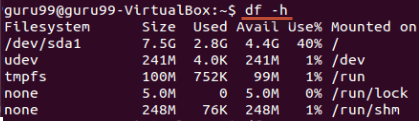
**DF**

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

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

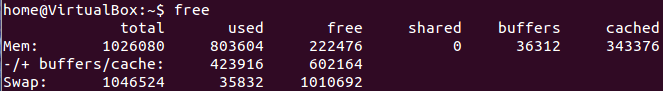
If you want the above information in a readable format, then use the command

'df -h'

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

**Free**

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

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

You can use the arguments

free -m to display output in MB

free -g to display output in GB