

Understanding the Process/job control commands in Linux

Whats a job in Linux

A job is a process that the shell manages. Each job is assigned a sequential job ID. Because a job is a process, each job has an associated PID. There are three types of job statuses:

1. **Foreground:** When you enter a command in a terminal window, the command occupies that terminal window until it completes. This is a foreground job.
2. **Background:** When you enter an ampersand (&) symbol at the end of a command line, the command runs without occupying the terminal window. The shell prompt is displayed immediately after you press Return. This is an example of a background job.
3. **Stopped:** If you press **Control + Z** for a foreground job, or enter the stop command for a background job, the job stops. This job is called a stopped job.

Job Control Commands

Job control commands enable you to place jobs in the foreground or background, and to start or stop jobs. The table describes the job control commands.

Option	Description
jobs	Lists all jobs
bg % n	Places the current or specified job in the background, where n is the job ID
fg % n	Brings the current or specified job into the foreground, where n is the job ID
Control-Z	Stops the foreground job and places it in the background as a stopped job

Note: The job control commands enable you to run and manage multiple jobs within a shell. However, you can use the job control commands only in the shell where the job was initiated.

Running a Job in the Background

To run a job in the background, you need to enter the command that you want to run, followed by an **ampersand (&)** symbol at the end of the command line. For example, run the sleep command in the background.

```
$ sleep 100 &  
[1] 1302  
$
```

The shell returns the job ID, in brackets, that it assigns to the command and the associated PID. With the job ID, you can use the job control commands to manage the job whereas the kernel uses PIDs to manage jobs.

When a background job is complete and you press Return, the shell displays a message indicating the job is done.

```
[1] + Done          sleep 100 &
$
```

Managing the background jobs

You can use the **jobs** command to list the jobs that are currently running or suspended in the background.

```
$ jobs
[1]+  Running          sleep 100 &
```

You can use the **fg** command to bring a background job to the foreground.

```
$ fg % 1
sleep 100
```

Note: The foreground job occupies the shell until the job is completed, suspended, or stopped and placed into the background.

You can use the '**Control+Z**' keys and **bg** command to return a job to the background. **The Control+Z keys suspend the job, and place it in the background as a stopped job.** **The bg command runs the job in the background.** For example:

1. Using CTRL+Z

```
$ sleep 100
^Z
[1]+  Stopped          sleep 100
$ jobs
[1]+  Stopped          sleep 100
```

2. Using bg

```
$ bg % 1
[1]+ sleep 100 &
$ jobs
[1]+  Running          sleep 100 &
```

Note: When you place a stopped job either in the foreground or background, the job restarts.

Top Command

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

```
home@VirtualBox:~$ top
top - 23:57:43 up 2:54, 1 user, load average: 0.00, 0.01, 0.05
Tasks: 189 total, 2 running, 187 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.7%us, 3.0%sy, 0.0%ni, 96.3%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1026080k total, 924508k used, 101572k free, 37000k buffers
Swap: 1046524k total, 21472k used, 1025052k free, 367996k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 1525 home       20   0 1775m 100m  28m  S   1.7  10.0   5:05.34 Photoshop.exe
   961 root        20   0 75972  51m 7952  R   1.0   5.1   2:23.42 Xorg
 1507 home       20   0  7644 4652  696  S   1.0   0.5   2:42.66 wineserver
 1564 home       20   0  75144 29m 9840  S   0.3   3.0   0:25.96 ubuntuone-syncd
 2999 home       20   0  127m  13m  10m  S   0.3   1.4   0:01.36 gnome-terminal
 3077 home       20   0  2820 1188  864  R   0.3   0.1   0:00.76 top
    1 root        20   0   3200 1704 1260  S   0.0   0.2   0:00.98 init
    2 root        20   0      0    0    0  S   0.0   0.0   0:00.00 kthreadd
    3 root        20   0      0    0    0  S   0.0   0.0   0:00.95 ksoftirqd/0
```

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
```

```
home@VirtualBox:~$ ps ux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
home     1114   0.0   0.8  46548  8512 ?        Ssl   Sep03   0:00 gnome-sess
home     1151   0.0   0.0   3856   140 ?        Ss    Sep03   0:00 /usr/bin/s
home     1154   0.0   0.0   3748   484 ?        S     Sep03   0:00 /usr/bin/d
home     1155   0.1   0.2   6656  3036 ?        Ss    Sep03   0:18 //bin/dbus
home     1157   0.0   0.2   9148  2368 ?        S     Sep03   0:00 /usr/lib/g
home     1162   0.0   0.2  31588  2296 ?        Ssl   Sep03   0:00 /usr/lib/g
home     1174   0.0   0.0  132472 14884 ?        Sl    Sep03   0:03 /usr/lib/g
```

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

```
ps PID
```

```
guru99@VirtualBox:~$ ps 1268
  PID TTY          STAT TIME  COMMAND
 1268 ?           S<l    0:02 /usr/bin/pulseaudio --start --log-target=syslog
```

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.

```
home@VirtualBox:~$ pidof Photoshop.exe
1525
home@VirtualBox:~$ kill 1525
```

DF

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

```
guru99@guru99-VirtualBox:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda1        7837756 2921376   4523216  40% /
udev             246488      4    246484   1% /dev
tmpfs            101512     752    100760   1% /run
none              5120        0      5120   0% /run/lock
none             253776     76    253700   1% /run/shm
```

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

```
'df -h'
```

```
guru99@guru99-VirtualBox:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        7.5G  2.8G  4.4G  40% /
udev             241M  4.0K  241M   1% /dev
tmpfs            100M  752K   99M   1% /run
none              5.0M      0   5.0M   0% /run/lock
none             248M  76K   248M   1% /run/shm
```

Free

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

```
home@VirtualBox:~$ free
              total        used        free      shared    buffers     cached
Mem:          1026080      803604      222476          0       36312      343376
-/+ buffers/cache:      423916      602164
Swap:          1046524       35832     1010692
```

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