Jobs is a shell builtin. It tells you about the jobs that the current shell is managing. It can give you information that is internal to the shell, like the job numbers (which you can use in shortcuts like fg %2) and the original command line as it appeared before variable expansions.

ps is an external command which can tell you about all the processes running on the system. (By default it only shows a small subset, but there are options to select larger sets of processes to display.) It doesn't know about the shell-internal stuff.

**Linux / Unix: jobs Command**

Job control is nothing but the ability to stop/suspend the execution of processes (command) and ontinue/resume their execution as per your requirements. This is done using your operating system and shell such as bash/ksh or POSIX shell.

## Purpose

Displays **status of jobs** in the current shell session.

## Syntax

The basic syntax is as follows:

jobs

OR

jobs jobID

o display the process ID or jobs for the job whose name begins with “p,” enter:  
$ jobs -p %p  
OR  
$ jobs %p

### How do I list only processes that have changed status since the last notification?

First, start a new job as follows:  
$ sleep 100 &  
Now, only show jobs that have stopped or exited since last notified, type:  
$ jobs -n  
Sample outputs:

[5]- Running sleep 100 &

### Display lists process IDs (PIDs) only

Pass the -p option to jobs command to display PIDs only:  
$ jobs -p

### How do I display only running jobs?

Pass the -r option to jobs command to display only running jobs only, type:  
$ jobs -r  
Sample outputs:

[1] Running gpass &

[2] Running gnome-calculator &

[3]- Running gedit fetch-stock-prices.py &

### How do I display only jobs that have stopped?

Pass the -s option to jobs command to display only stopped jobs only, type:  
$ jobs -s  
Sample outputs:

[4]+ Stopped ping cyberciti.biz

## obs command options

From the bash(1) command man page:

|  |  |
| --- | --- |
| Option | Description |
| **-l** | Show process id’s in addition to the normal information. |
| **-p** | Show process id’s only. |
| **-n** | Show only processes that have changed status since the last notification are printed. |
| **-r** | Restrict output to running jobs only. |
| **-s** | Restrict output to stopped jobs only. |
| **-x** | COMMAND is run after all job specifications that appear in ARGS have been replaced with the process ID of that job’s process group leader./td> |

### A note about /usr/bin/jobs and shell builtin

Type the following type command to find out whether jobs is part of shell, external command or both:  
$ type -a jobs  
Sample outputs:

jobs is a shell builtin

jobs is /usr/bin/jobs

**Process : PS**

1. If you run **ps command** without any arguments, it displays processes for the current shell.

**$ps**

1. Display every active process on a Linux system in generic (Unix/Linux) format.

$ps –A

$ps –e

1. Display all processes in **BSD** format.

$ps au

$ps aux

1. To perform a full-format listing, add the **-f** or **-F** flag.

**$ps –ef**

1. You can select all processes owned by you (runner of the **ps command**, root in this case), type:

$ps –x

1. To display a user’s processes by real user ID (**RUID**) or name, use the **-U** flag.

**$ps –fU uname**

1. The command below enables you to view every process running with **root** user privileges (real & effective ID) in user format.

**$ps –U root –u root**

1. If you want to list all processes owned by a certain group (real group ID (**RGID**) or name), type.

**$ps –fG aphace**

**$ps –fG aphace :owned by**

To select process by **PPID**, type.

$ps –f --ppid 122

Check execution time of a process.

**$ ps -eo comm,etime,user | grep httpd**

**Find top running processes by highest memory and CPU usage in Linux.**

**$ ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem | head**

**OR**

**$ ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%cpu | head**

**To kill an Linux processes/unresponsive applications or any process that is consuming high CPU time.**

**First, find the PID of the unresponsive process or application.**

**$ ps -A | grep -i stress**

Kill Process

$ kill -9 2583 2584