An example is as follows. Here, `comm` stands for COMMAND and `pcpu` is percent of CPU usage:

```
$ ps -eo comm,pcpu | head
COMMAND          %CPU
init              0.0
kthreadd          0.0
migration/0       0.0
ksoftirqd/0       0.0
watchdog/0        0.0
events/0          0.0
cpuset            0.0
khelper           0.0
netns             0.0
```

## How it works...

The different parameters that can be used with the `-o` option and their descriptions are as follows:

| Parameter | Description |
|---|---|
| `pcpu` | Percentage of CPU |
| `pid` | Process ID |
| `ppid` | Parent Process ID |
| `pmem` | Percentage of memory |
| `comm` | Executable filename |
| `cmd` | Simple command |
| `user` | The user who started the process |
| `nice` | The priority (niceness) |
| `time` | Cumulative CPU time |
| `etime` | Elapsed time since the process started |
| `tty` | The associated TTY device |
| `euid` | The effective user |
| `stat` | Process state |

## There's more...

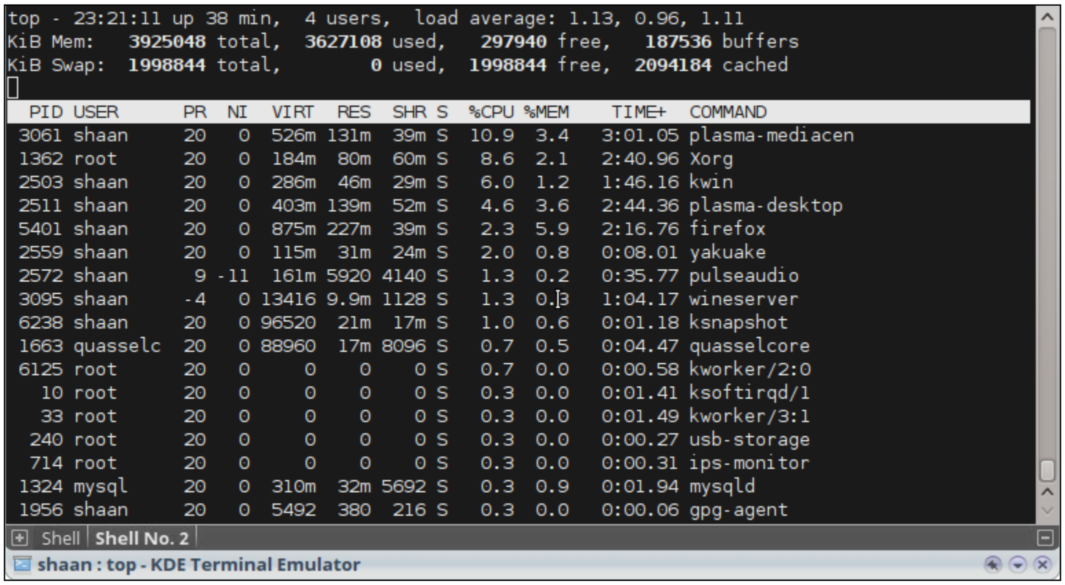
Let's go through additional usage examples of process manipulation commands.

### top

`top` is a very important command for system administrators. The `top` command will, by default, output a list of top CPU consuming processes. The output is updated every few seconds, and is used as follows:

```
$ top
```

It will display several parameters along with the top CPU consuming processes:



```
top - 23:21:11 up 38 min,  4 users,  load average: 1.13, 0.96, 1.11
KiB Mem:   3925048 total,  3627108 used,   297940 free,   187536 buffers
KiB Swap:  1998844 total,        0 used,  1998844 free,  2094184 cached

  PID USER      PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
 3061 shaan     20   0  526m 131m  39m S  10.9  3.4   3:01.05 plasma-mediacen
 1362 root      20   0  184m  80m  60m S   8.6  2.1   2:40.96 Xorg
 2503 shaan     20   0  286m  46m  29m S   6.0  1.2   1:46.16 kwin
 2511 shaan     20   0  403m 139m  52m S   4.6  3.6   2:44.36 plasma-desktop
 5401 shaan     20   0  875m 227m  39m S   2.3  5.9   2:16.76 firefox
 2559 shaan     20   0  115m  31m  24m S   2.0  0.8   0:08.01 yakuake
 2572 shaan      9 -11  161m 5920 4140 S   1.3  0.2   0:35.77 pulseaudio
 3095 shaan     -4   0 13416 9.9m 1128 S   1.3  0.3   1:04.17 wineserver
 6238 shaan     20   0 96520  21m  17m S   1.0  0.6   0:01.18 ksnapshot
 1663 quasselc  20   0 88960  17m 8096 S   0.7  0.5   0:04.47 quasselcore
 6125 root      20   0     0    0    0 S   0.7  0.0   0:00.58 kworker/2:0
   10 root      20   0     0    0    0 S   0.3  0.0   0:01.41 ksoftirqd/1
   33 root      20   0     0    0    0 S   0.3  0.0   0:01.49 kworker/3:1
  240 root      20   0     0    0    0 S   0.3  0.0   0:00.27 usb-storage
  714 root      20   0     0    0    0 S   0.3  0.0   0:00.31 ips-monitor
 1324 mysql     20   0  310m  32m 5692 S   0.3  0.9   0:01.94 mysqld
 1956 shaan     20   0  5492  380  216 S   0.3  0.0   0:00.06 gpg-agent
```

### Sorting the ps output with respect to a parameter

Output of the `ps` command can be sorted according to specified columns with the `--sort` parameter. The ascending or descending order can be specified by using the + (ascending) or - (descending) prefix to the parameter as follows:

```
$ ps [OPTIONS] --sort -paramter1,+parameter2,parameter3..
```