top command

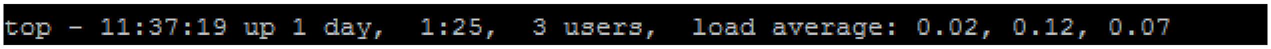
Know what is happening in “real time” on your systems is in my opinion the basis to use and optimize your OS. The `top` command can help us, this is a very useful system monitor that is really easy to use, and that can also allows us to understand why our OS suffers and which process use most resources. The command to be run on the terminal is:

```
$ top
```

And we’ll get a screen similar to the one on the right:

Let’s see now every single row of this output to explain all the information found within the screen.

1° Row — top

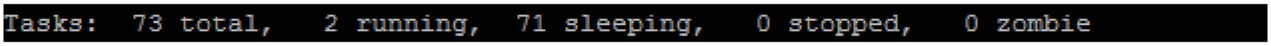


```
top - 11:37:19 up 1 day,  1:25,  3 users,  load average: 0.02, 0.12, 0.07
```

This first line indicates in order:

- current time (11:37:19)
- uptime of the machine (up 1 day, 1:25)
- users sessions logged in (3 users)
- average load on the system (load average: 0.02, 0.12, 0.07) the 3 values refer to the last minute, five minutes and 15 minutes.

2° Row – task

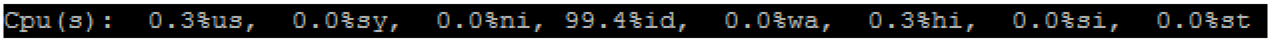


```
Tasks:  73 total,   2 running,  71 sleeping,   0 stopped,   0 zombie
```

The second row gives the following information:

- Processes running in totals (73 total)
- Processes running (2 running)
- Processes sleeping (71 sleeping)
- Processes stopped (0 stopped)
- Processes waiting to be stoppati from the parent process (0 zombie)

3° Row – cpu

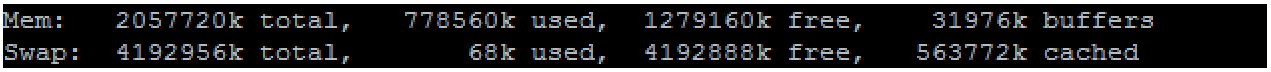


```
Cpu(s):  0.3%us,  0.0%sy,  0.0%ni, 99.4%id,  0.0%wa,  0.3%hi,  0.0%si,  0.0%st
```

The third line indicates how the cpu is used. If you sum up all the percentages the total will be 100% of the cpu. Let’s see what these values indicate in order:

- Percentage of the CPU for user processes (0.3%**us**)
- Percentage of the CPU for system processes (0.0%**sy**)
- Percentage of the CPU processes with priority upgrade *nice* (0.0%**ni**)
- Percentage of the CPU not used (99,4%**id**)
- Percentage of the CPU processes waiting for I/O operations(0.0%**wa**)
- Percentage of the CPU serving hardware interrupts (0.3% **hi** — Hardware IRQ
- Percentage of the CPU serving software interrupts (0.0% **si** — Software Interrupts
- The amount of CPU ‘stolen’ from this virtual machine by the hypervisor for other tasks (such as running another virtual machine) this will be 0 on desktop and server without Virtual machine. (0.0%**st** — Steal Time)

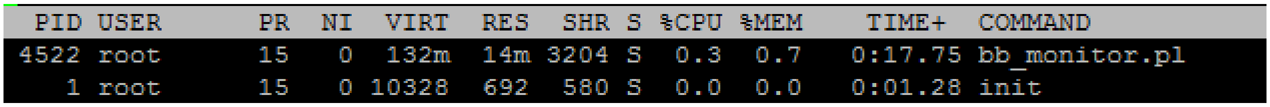
4° and 5° Rows – memory usage



```
Mem:   2057720k total,   778560k used,  1279160k free,    31976k buffers
Swap:  4192956k total,       68k used,  4192888k free,   563772k cached
```

The fourth and fifth rows respectively indicate the use of physical memory (RAM) and swap. In this order: Total memory in use, free, buffers cached.

Following Rows — Processes list



```
  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 4522 root      15   0  132m  14m 3204 S  0.3  0.7   0:17.75 bb_monitor.pl
    1 root      15   0 10328  692  580 S  0.0  0.0   0:01.28 init
```

And as last thing ordered by CPU usage (as default) there are the processes currently in use. Let’s see what information we can get in the different columns:

- **PID** – l'ID of the process(4522)
- **USER** – The user that is the owner of the process (root)
- **PR** – priority of the process (15)
- **NI** – The "NICE" value of the process (0)
- **VIRT** – virtual memory used by the process (132m)
- **RES** – physical memory used from the process (14m)
- **SHR** – shared memory of the process (3204)
- **S** – indicates the status of the process: **S**=sleep **R**=running **Z**=zombie (S)
- **%CPU** – This is the percentage of CPU used by this process (0.3)
- **%MEM** – This is the percentage of RAM used by the process (0.7)
- **TIME+** –This is the total time of activity of this process (0:17.75)
- **COMMAND** – And this is the name of the process (bb_monitor.pl)

Conclusions

Now that we have seen in detail all the information that the command "top" returns, it will be easier to understand the reason of excessive load and/or the slowing of the system