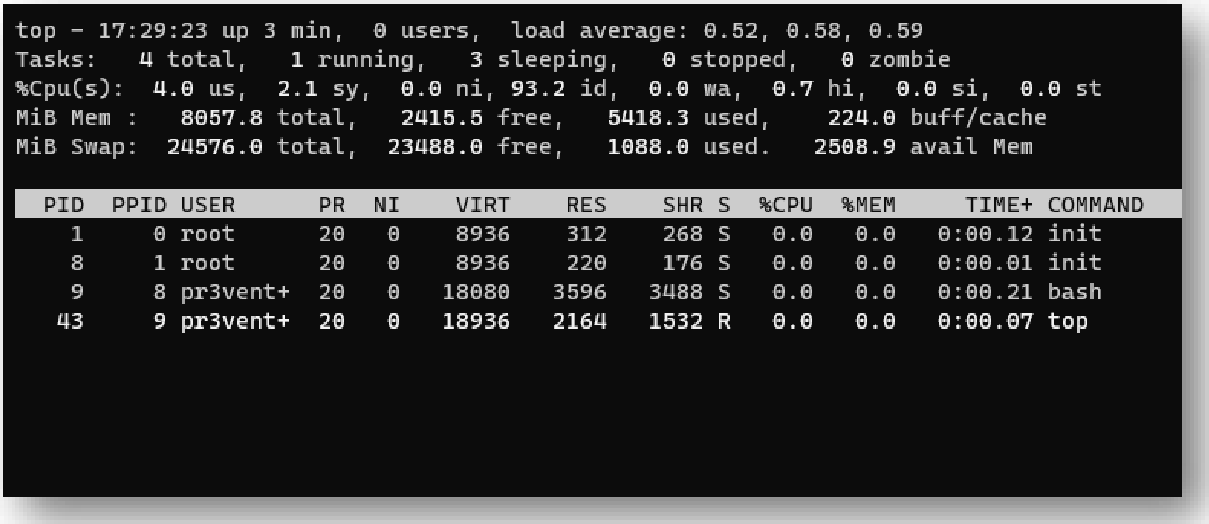
MONITORING SYSTEM

“TOP”

How can I know what process is running and who is running it?

The packages top is a programs that allow us to have a general vision of which process are running in our server. The best part is that it shows the information in a real time view and so we can have a clear idea of what is really happening in our servers at the level of processes and performance. Let´s see how top looks like.

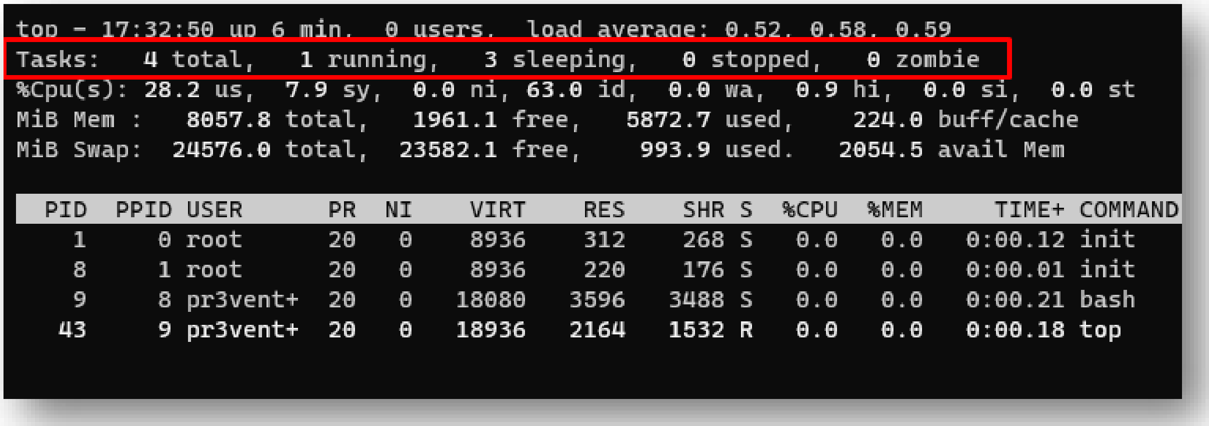
Command: “**top”**



```
top - 17:29:23 up 3 min,  0 users,  load average: 0.52, 0.58, 0.59
Tasks:   4 total,   1 running,   3 sleeping,   0 stopped,   0 zombie
%Cpu(s):  4.0 us,  2.1 sy,  0.0 ni, 93.2 id,  0.0 wa,  0.7 hi,  0.0 si,  0.0 st
MiB Mem :   8057.8 total,   2415.5 free,   5418.3 used,    224.0 buff/cache
MiB Swap:  24576.0 total,  23488.0 free,   1088.0 used.   2508.9 avail Mem

  PID  PPID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    1     0 root      20   0    8936    312    268 S   0.0   0.0   0:00.12 init
    8     1 root      20   0    8936    220    176 S   0.0   0.0   0:00.01 init
    9     8 pr3vent+  20   0   18080   3596   3488 S   0.0   0.0   0:00.21 bash
   43     9 pr3vent+  20   0   18936   2164   1532 R   0.0   0.0   0:00.07 top
```

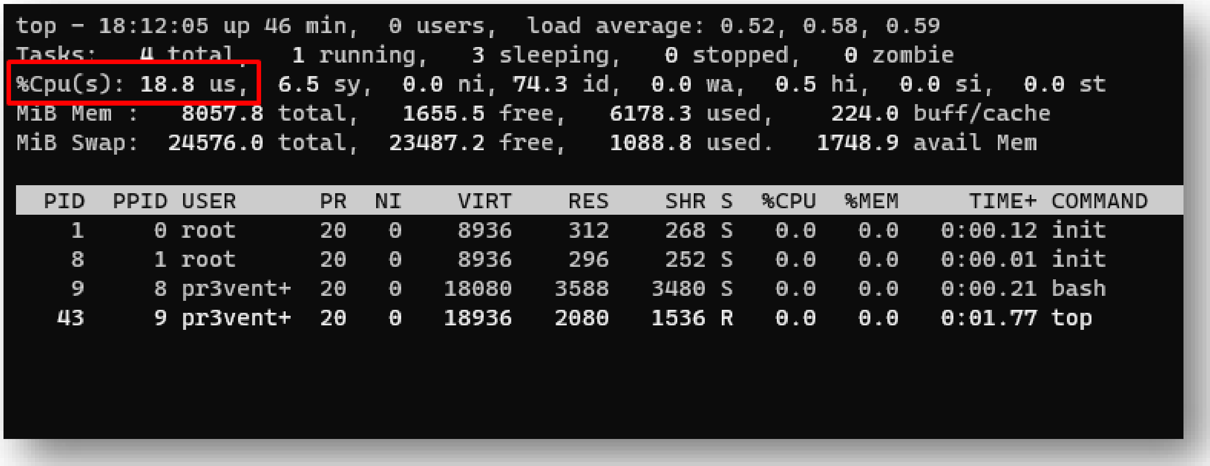
In this output we can find some interesting information such as:

1. The tasks that are currently running.



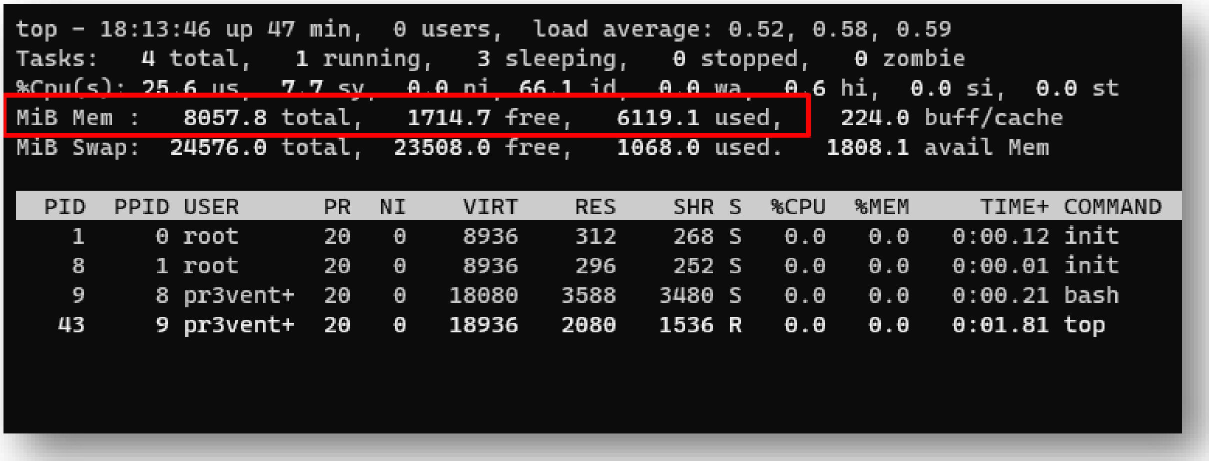
```
top - 17:32:50 up 6 min,  0 users,  load average: 0.52, 0.58, 0.59
Tasks:   4 total,   1 running,   3 sleeping,   0 stopped,   0 zombie
%Cpu(s): 28.2 us,  7.9 sy,  0.0 ni, 63.0 id,  0.0 wa,  0.9 hi,  0.0 si,  0.0 st
MiB Mem :   8057.8 total,   1961.1 free,   5872.7 used,    224.0 buff/cache
MiB Swap:  24576.0 total,  23582.1 free,    993.9 used.   2054.5 avail Mem

  PID  PPID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    1     0 root      20   0    8936    312    268 S   0.0   0.0   0:00.12 init
    8     1 root      20   0    8936    220    176 S   0.0   0.0   0:00.01 init
    9     8 pr3vent+  20   0   18080   3596   3488 S   0.0   0.0   0:00.21 bash
   43     9 pr3vent+  20   0   18936   2164   1532 R   0.0   0.0   0:00.18 top
```

2. The percentage of Cpu that is being used.



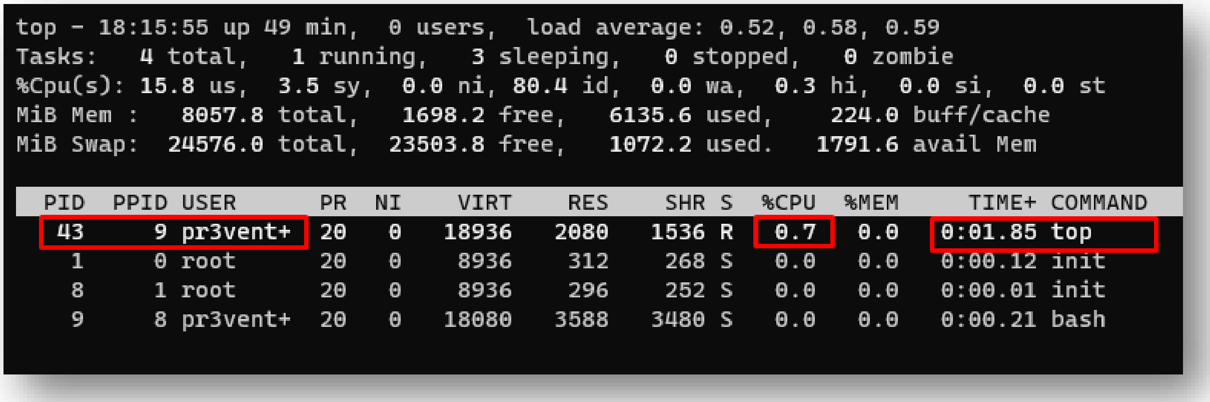
```
top - 18:12:05 up 46 min,  0 users,  load average: 0.52, 0.58, 0.59
Tasks:   4 total,   1 running,   3 sleeping,   0 stopped,   0 zombie
%Cpu(s): 18.8 us,  6.5 sy,  0.0 ni, 74.3 id,  0.0 wa,  0.5 hi,  0.0 si,  0.0 st
MiB Mem :   8057.8 total,   1655.5 free,   6178.3 used,    224.0 buff/cache
MiB Swap:  24576.0 total,  23487.2 free,   1088.8 used.   1748.9 avail Mem

  PID  PPID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    1     0 root      20   0    8936    312    268 S   0.0   0.0   0:00.12 init
    8     1 root      20   0    8936    296    252 S   0.0   0.0   0:00.01 init
    9     8 pr3vent+  20   0   18080   3588   3480 S   0.0   0.0   0:00.21 bash
   43     9 pr3vent+  20   0   18936   2080   1536 R   0.0   0.0   0:01.77 top
```

3. The total ram memory, which is being used, and the percentage available.



```
top - 18:13:46 up 47 min,  0 users,  load average: 0.52, 0.58, 0.59
Tasks:   4 total,   1 running,   3 sleeping,   0 stopped,   0 zombie
%Cpu(s): 25.6 us,  7.7 sy,  0.0 ni, 66.1 id,  0.0 wa,  0.6 hi,  0.0 si,  0.0 st
MiB Mem :   8057.8 total,   1714.7 free,   6119.1 used,    224.0 buff/cache
MiB Swap:  24576.0 total,  23508.0 free,   1068.0 used.   1808.1 avail Mem

  PID  PPID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    1     0 root      20   0    8936    312    268 S   0.0   0.0   0:00.12 init
    8     1 root      20   0    8936    296    252 S   0.0   0.0   0:00.01 init
    9     8 pr3vent+  20   0   18080   3588   3480 S   0.0   0.0   0:00.21 bash
   43     9 pr3vent+  20   0   18936   2080   1536 R   0.0   0.0   0:01.81 top
```

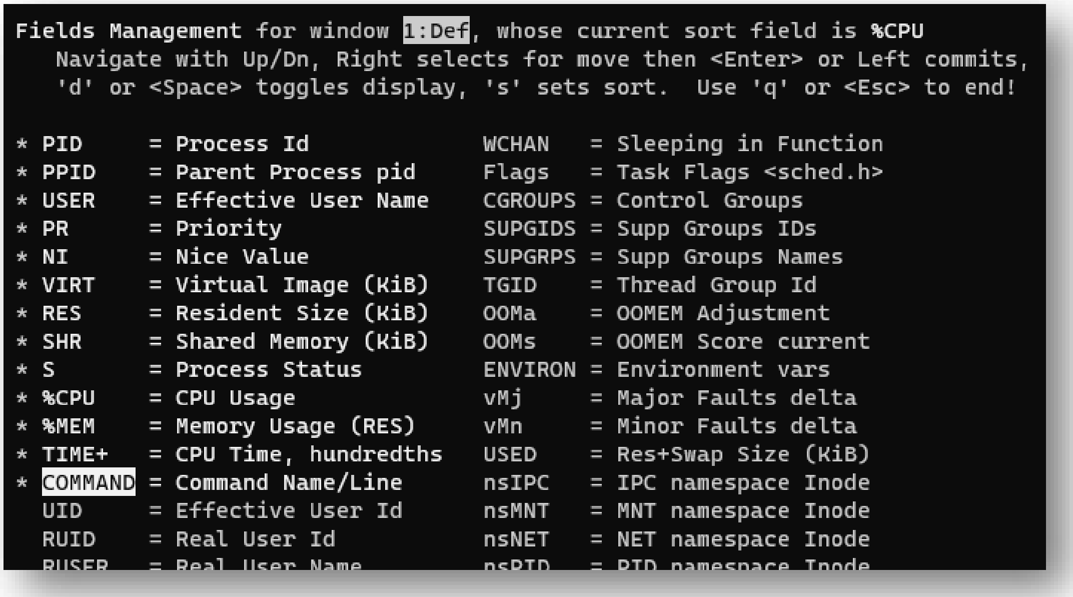
4. We can obtain the id of the process, the parent process, the user that executed the process, the occupation at the cpu level, the time the action was executed and finally the name of the program that was executed.



```
top - 18:15:55 up 49 min,  0 users,  load average: 0.52, 0.58, 0.59
Tasks:   4 total,   1 running,   3 sleeping,   0 stopped,   0 zombie
%Cpu(s): 15.8 us,  3.5 sy,  0.0 ni, 80.4 id,  0.0 wa,  0.3 hi,  0.0 si,  0.0 st
MiB Mem :  8057.8 total,  1698.2 free,  6135.6 used,   224.0 buff/cache
MiB Swap:  24576.0 total,  23503.8 free,   1072.2 used.   1791.6 avail Mem

  PID  PPID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   43     9 pr3vent+  20   0   18936   2080   1536 R   0.7   0.0   0:01.85 top
    1     0 root      20   0    8936    312    268 S   0.0   0.0   0:00.12 init
    8     1 root      20   0    8936    296    252 S   0.0   0.0   0:00.01 init
    9     8 pr3vent+  20   0   18080   3588   3480 S   0.0   0.0   0:00.21 bash
```

But is that all we can get? the answer is no. the good thing about top is that it is completely customizable. If we press the **"f"** key, we can access a menu where we can add or remove the data types that we want to obtain in the output of the top command.

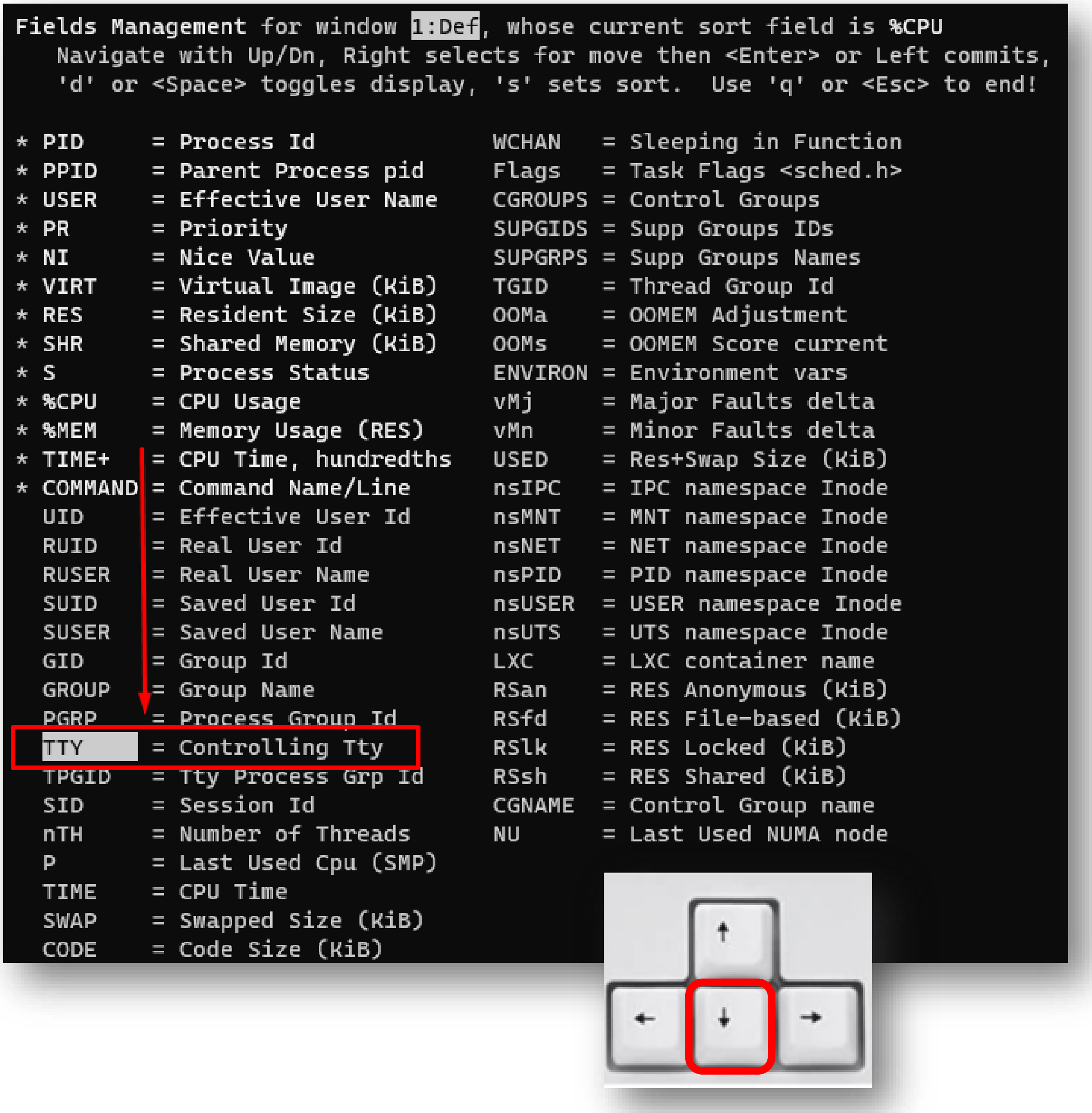
Press “f”



```
Fields Management for window 1:Def, whose current sort field is %CPU
   Navigate with Up/Dn, Right selects for move then <Enter> or Left commits,
   'd' or <Space> toggles display, 's' sets sort.  Use 'q' or <Esc> to end!

* PID     = Process Id            WCHAN   = Sleeping in Function
* PPID    = Parent Process pid    Flags   = Task Flags <sched.h>
* USER    = Effective User Name   CGROUPS = Control Groups
* PR      = Priority              SUPGIDS = Supp Groups IDs
* NI      = Nice Value            SUPGRPS = Supp Groups Names
* VIRT    = Virtual Image (KiB)   TGID    = Thread Group Id
* RES     = Resident Size (KiB)   OOMa    = OOMEM Adjustment
* SHR     = Shared Memory (KiB)   OOMs    = OOMEM Score current
* S       = Process Status        ENVIRON = Environment vars
* %CPU    = CPU Usage             vMj     = Major Faults delta
* %MEM    = Memory Usage (RES)    vMn     = Minor Faults delta
* TIME+   = CPU Time, hundredths  USED    = Res+Swap Size (KiB)
* COMMAND = Command Name/Line     nsIPC   = IPC namespace Inode
  UID     = Effective User Id     nsMNT   = MNT namespace Inode
  RUID    = Real User Id          nsNET   = NET namespace Inode
  RUSER   = Real User Name        nsPID   = PID namespace Inode
```

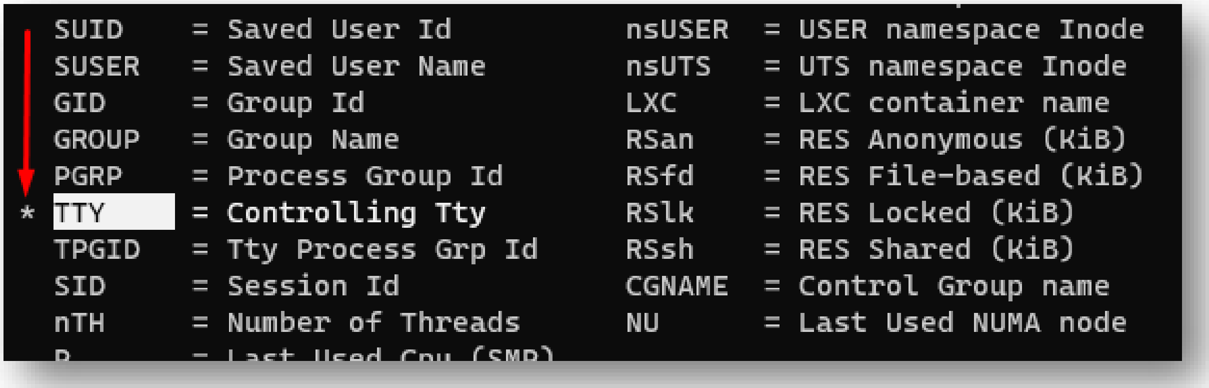
How does this menu work? Simple, with the arrow keys of the keyboard, we navigate each of the options and those that are active in the output of the command are those marked with the asterisk symbol (*)

The way to add data or delete existing data is to assign the asterisk (*) using the space bar to be located in each option. In this case I scroll down to the TTY option since I want to add this information to be able to see in which tty this process is being executed



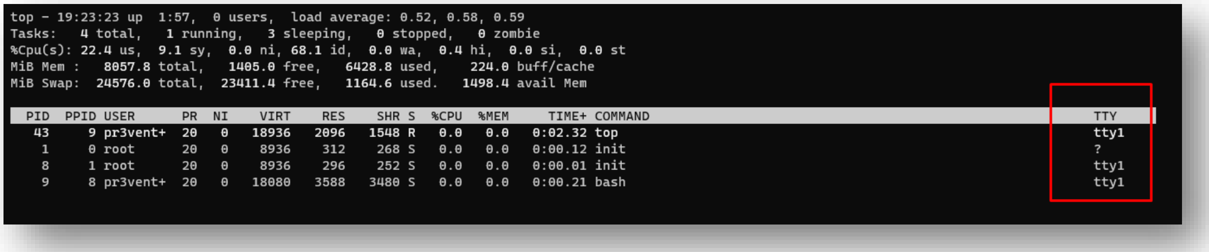
```
Fields Management for window 1:Def, whose current sort field is %CPU
   Navigate with Up/Dn, Right selects for move then <Enter> or Left commits,
   'd' or <Space> toggles display, 's' sets sort.  Use 'q' or <Esc> to end!

* PID     = Process Id             WCHAN   = Sleeping in Function
* PPID    = Parent Process pid     Flags   = Task Flags <sched.h>
* USER    = Effective User Name    CGROUPS = Control Groups
* PR      = Priority               SUPGIDS = Supp Groups IDs
* NI      = Nice Value             SUPGRPS = Supp Groups Names
* VIRT    = Virtual Image (KiB)    TGID    = Thread Group Id
* RES     = Resident Size (KiB)    OOMa    = OOMEM Adjustment
* SHR     = Shared Memory (KiB)    OOMs    = OOMEM Score current
* S       = Process Status         ENVIRON = Environment vars
* %CPU    = CPU Usage              vMj     = Major Faults delta
* %MEM    = Memory Usage (RES)     vMn     = Minor Faults delta
* TIME+   = CPU Time, hundredths   USED    = Res+Swap Size (KiB)
* COMMAND = Command Name/Line      nsIPC   = IPC namespace Inode
  UID     = Effective User Id      nsMNT   = MNT namespace Inode
  RUID    = Real User Id           nsNET   = NET namespace Inode
  RUSER   = Real User Name         nsPID   = PID namespace Inode
  SUID    = Saved User Id          nsUSER  = USER namespace Inode
  SUSER   = Saved User Name        nsUTS   = UTS namespace Inode
  GID     = Group Id               LXC     = LXC container name
  GROUP   = Group Name             RSan    = RES Anonymous (KiB)
  PGRP    = Process Group Id       RSfd    = RES File-based (KiB)
  TTY     = Controlling Tty        RSlk    = RES Locked (KiB)
  TPGID   = Tty Process Grp Id     RSsh    = RES Shared (KiB)
  SID     = Session Id             CGNAME  = Control Group name
  nTH     = Number of Threads      NU      = Last Used NUMA node
  P       = Last Used Cpu (SMP)
  TIME    = CPU Time
  SWAP    = Swapped Size (KiB)
  CODE    = Code Size (KiB)
```

So I have to press the space bar and this will activate this information in the command output.



```
   SUID     = Saved User Id            nsUSER   = USER namespace Inode
   SUSER    = Saved User Name          nsUTS    = UTS namespace Inode
   GID      = Group Id                 LXC      = LXC container name
   GROUP    = Group Name               RSan     = RES Anonymous (KiB)
   PGRP     = Process Group Id         RSfd     = RES File-based (KiB)
 * TTY      = Controlling Tty          RSlk     = RES Locked (KiB)
   TPGID    = Tty Process Grp Id       RSsh     = RES Shared (KiB)
   SID      = Session Id               CGNAME   = Control Group name
   nTH      = Number of Threads        NU       = Last Used NUMA node
   P        = Last Used Cpu (SMP)
```

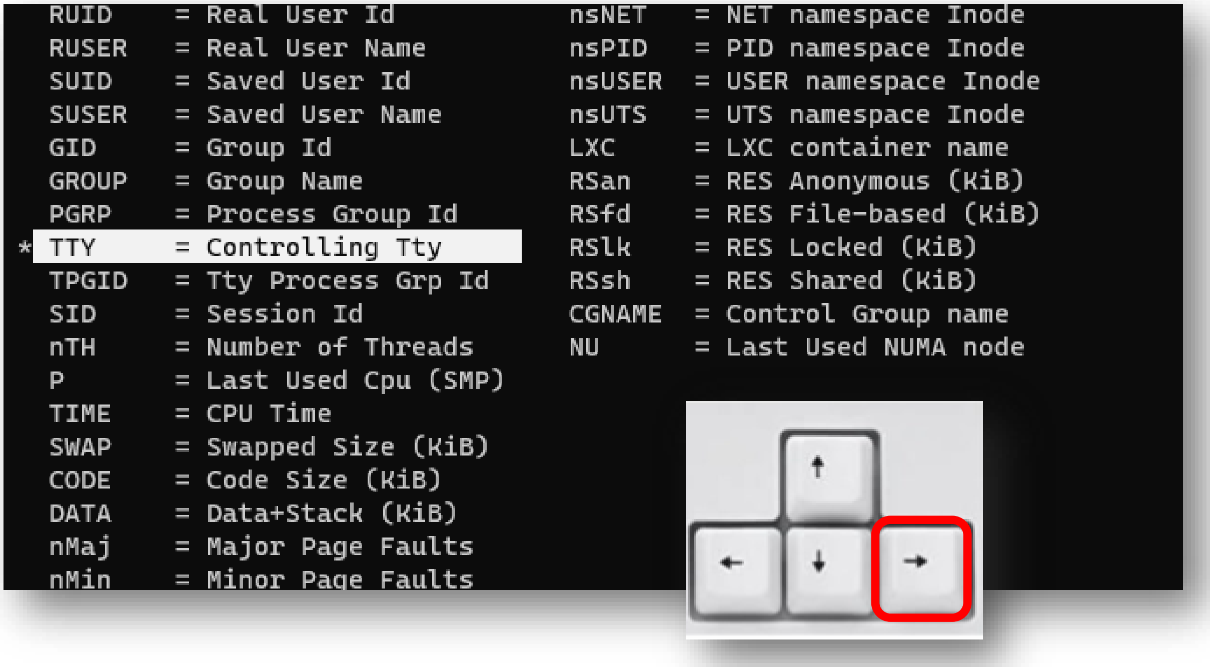
Then, we press the escape or esc key and return to the top output.



```
top - 19:23:23 up  1:57,  0 users,  load average: 0.52, 0.58, 0.59
Tasks:   4 total,   1 running,   3 sleeping,   0 stopped,   0 zombie
%Cpu(s): 22.4 us,  9.1 sy,  0.0 ni, 68.1 id,  0.0 wa,  0.4 hi,  0.0 si,  0.0 st
MiB Mem :   8057.8 total,   1405.0 free,   6428.8 used,    224.0 buff/cache
MiB Swap:  24576.0 total,  23411.4 free,   1164.6 used.   1498.4 avail Mem

  PID  PPID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND                                         TTY
   43     9 pr3vent+  20   0   18936   2096   1548 R   0.0   0.0   0:02.32 top                                             tty1
    1     0 root      20   0    8936    312    268 S   0.0   0.0   0:00.12 init                                            ?
    8     1 root      20   0    8936    296    252 S   0.0   0.0   0:00.01 init                                            tty1
    9     8 pr3vent+  20   0   18080   3588   3480 S   0.0   0.0   0:00.21 bash                                            tty1
```

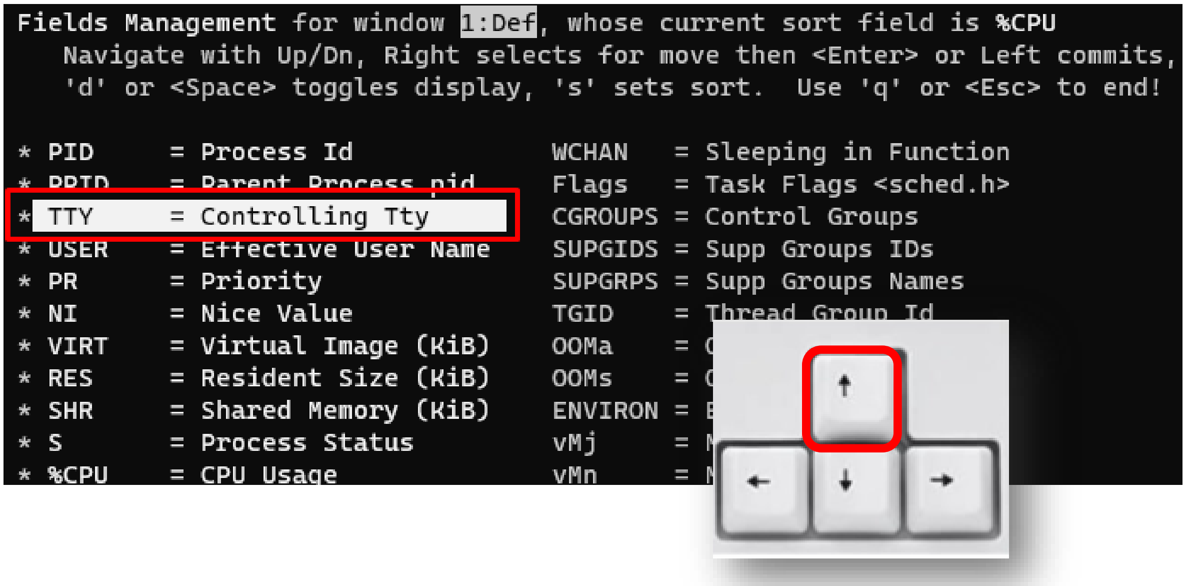
But now that data is seen as lonely. this can be fixed given the flexibility of the top options to customize it

We must press the f key again to return to the menu and we scroll to the TTY option, once there we press the direction key to the right to select the entire command



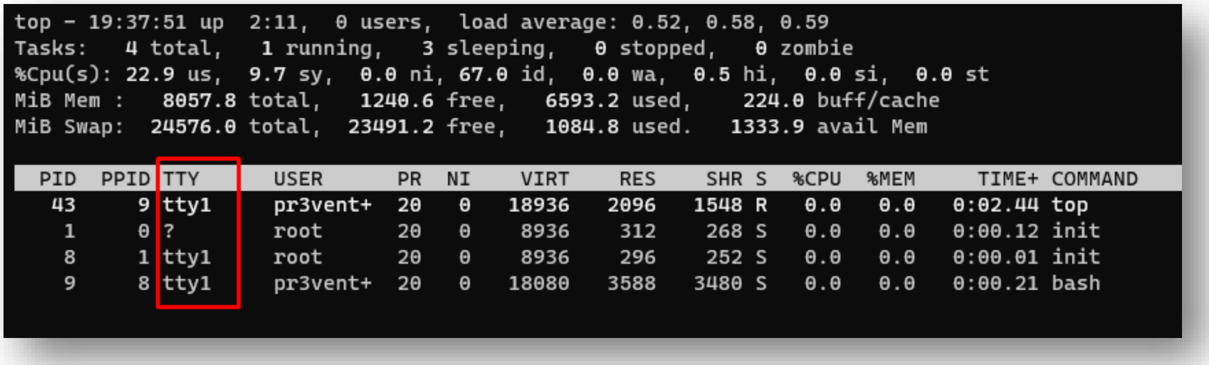
```
   RUID     = Real User Id             nsNET    = NET namespace Inode
   RUSER    = Real User Name           nsPID    = PID namespace Inode
   SUID     = Saved User Id            nsUSER   = USER namespace Inode
   SUSER    = Saved User Name          nsUTS    = UTS namespace Inode
   GID      = Group Id                 LXC      = LXC container name
   GROUP    = Group Name               RSan     = RES Anonymous (KiB)
   PGRP     = Process Group Id         RSfd     = RES File-based (KiB)
 * TTY      = Controlling Tty          RSlk     = RES Locked (KiB)
   TPGID    = Tty Process Grp Id       RSsh     = RES Shared (KiB)
   SID      = Session Id               CGNAME   = Control Group name
   nTH      = Number of Threads        NU       = Last Used NUMA node
   P        = Last Used Cpu (SMP)
   TIME     = CPU Time
   SWAP     = Swapped Size (KiB)
   CODE     = Code Size (KiB)
   DATA     = Data+Stack (KiB)
   nMaj     = Major Page Faults
   nMin     = Minor Page Faults
```

Now that the TTY command is selected, we must raise or lower it according to our visual comfort, in my case I will raise it and place it right next to the PPID



```
Fields Management for window 1:Def, whose current sort field is %CPU
   Navigate with Up/Dn, Right selects for move then <Enter> or Left commits,
   'd' or <Space> toggles display, 's' sets sort.  Use 'q' or <Esc> to end!

* PID     = Process Id             WCHAN   = Sleeping in Function
* PPID    = Parent Process pid     Flags   = Task Flags <sched.h>
* TTY     = Controlling Tty        CGROUPS = Control Groups
* USER    = Effective User Name    SUPGIDS = Supp Groups IDs
* PR      = Priority               SUPGRPS = Supp Groups Names
* NI      = Nice Value             TGID    = Thread Group Id
* VIRT    = Virtual Image (KiB)    OOMa    = O
* RES     = Resident Size (KiB)    OOMs    = O
* SHR     = Shared Memory (KiB)    ENVIRON = E
* S       = Process Status         vMj     = M
* %CPU    = CPU Usage              vMn     = M
```

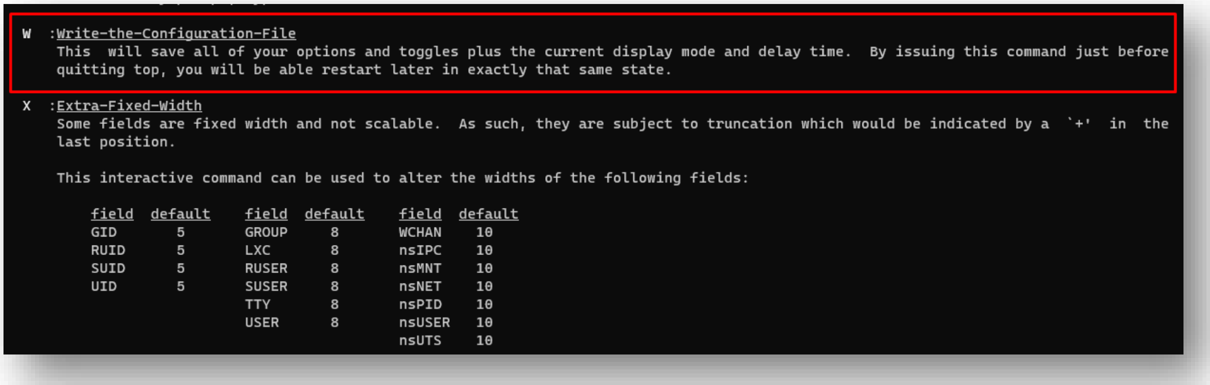
Let's take a look at the command output after we have customized it a bit.



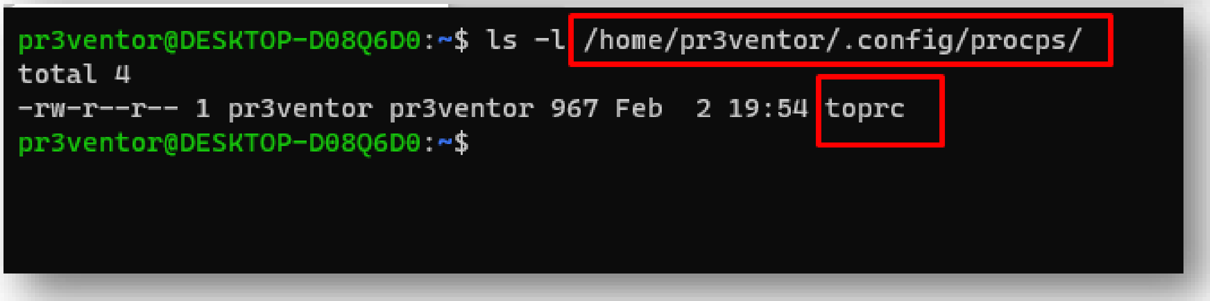
```
top - 19:37:51 up  2:11,  0 users,  load average: 0.52, 0.58, 0.59
Tasks:   4 total,   1 running,   3 sleeping,   0 stopped,   0 zombie
%Cpu(s): 22.9 us,  9.7 sy,  0.0 ni, 67.0 id,  0.0 wa,  0.5 hi,  0.0 si,  0.0 st
MiB Mem :  8057.8 total,  1240.6 free,  6593.2 used,   224.0 buff/cache
MiB Swap:  24576.0 total,  23491.2 free,  1084.8 used.  1333.9 avail Mem

  PID  PPID TTY      USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   43     9 tty1     pr3vent+  20   0   18936   2096   1548 R   0.0   0.0   0:02.44 top
    1     0 ?        root      20   0    8936    312    268 S   0.0   0.0   0:00.12 init
    8     1 tty1     root      20   0    8936    296    252 S   0.0   0.0   0:00.01 init
    9     8 tty1     pr3vent+  20   0   18080   3588   3480 S   0.0   0.0   0:00.21 bash
```

Great, just what we were looking for. This was only a test but the possibilities are as many as we need them. We have a disadvantage but it comes with a solution. When we exit the command, this customization will be lost. But there is a way to make it permanent.

If we check the internal manual of the program, there is an interesting option which is "W". With this option, the top configuration file will be created to be able to save that customization



```
 W  :Write-the-Configuration-File
     This  will save all of your options and toggles plus the current display mode and delay time.  By issuing this command just before
     quitting top, you will be able restart later in exactly that same state.

 X  :Extra-Fixed-Width
     Some fields are fixed width and not scalable.  As such, they are subject to truncation which would be indicated by a  `+'  in  the
     last position.

     This interactive command can be used to alter the widths of the following fields:

         field  default    field  default    field  default
         GID       5       GROUP     8       WCHAN    10
         RUID      5       LXC       8       nsIPC    10
         SUID      5       RUSER     8       nsMNT    10
         UID       5       SUSER     8       nsNET    10
                           TTY       8       nsPID    10
                           USER      8       nsUSER   10
                                             nsUTS    10
```



```
pr3ventor@DESKTOP-D08Q6D0:~$ ls -l /home/pr3ventor/.config/procps/
total 4
-rw-r--r-- 1 pr3ventor pr3ventor 967 Feb  2 19:54 toprc
pr3ventor@DESKTOP-D08Q6D0:~$
```

And we have better news, this file can be copied to different computers, in order to have a standard if we need it. In other words, we do not need to configure dozens of times, but to configure only one and then export to as many computers as we need. Hope you learned something new today.