

Actividad de Gestión de Procesos

NOMBRE:

FECHA:

Abre una terminal Linux y vé realizando los siguientes pasos. Responde a las preguntas indicadas.

1. Ejecuta el comando `ps`. ¿Cuántos procesos aparecen? ¿Cual es su PID? Seguramente tienes más procesos ejecutándose en tu ordenador, como el procesador de textos o firefox. Es curioso que esos no aparezcan con `ps`. Ahora veremos por qué.



```
Terminal 22 de feb 00:42 57 %  
Terminal  
> ps  
  PID TTY          TIME CMD  
 8846 pts/0    00:00:00 zsh  
 8850 pts/0    00:00:00 zsh  
 8851 pts/0    00:00:00 zsh  
 8868 pts/0    00:00:00 zsh  
 8869 pts/0    00:00:00 zsh  
 8871 pts/0    00:00:00 gitstatusd-linu  
 8888 pts/0    00:00:00 sleep  
 8891 pts/0    00:00:00 ps
```

Aparecen 8 procesos

Sus PID son: 8846, 8850, 8851, 8868, 8869, 8871, 8888 y 8891

2. Ejecuta el comando `firefox &`, el símbolo de ampersand `&` es importante ponerlo para que lo ejecute en segundo plano y te permita seguir usando la terminal. Deja firefox abierto. Vuelve a la terminal y ejecuta `ps`. ¿Qué proceso nuevo aparece? ¿Cual es su PID? Como habrás observado, el comando `ps` solo muestra los procesos que se están ejecutando en esa terminal (y solo en esa terminal). Cierra la ventana de firefox que abriste antes.



```
Terminal 22 de feb 00:43 58 %  
Terminal  
> ps  
  PID TTY          TIME CMD  
 8846 pts/0    00:00:00 zsh  
 8850 pts/0    00:00:00 zsh  
 8868 pts/0    00:00:00 zsh  
 8869 pts/0    00:00:00 zsh  
 8871 pts/0    00:00:00 gitstatusd-linu  
 9062 pts/0    00:00:05 firefox  
 9159 pts/0    00:00:00 Socket Process  
 9180 pts/0    00:00:00 Privileged Cont  
 9232 pts/0    00:00:02 WebExtensions  
 9281 pts/0    00:00:00 Web Content  
 9288 pts/0    00:00:00 Web Content  
 9307 pts/0    00:00:00 Web Content  
 9366 pts/0    00:00:00 ps
```

Aparecen 7 nuevos procesos pero creo que todos son por el firefox.
El PID de firefox es 9062.

3. Ejecuta el comando `ps aux`. Verás que aparecen muchísimos procesos, eso es porque `ps aux` muestra todos los procesos del sistema (los tuyos, los de otros usuarios y los del sistema operativo). ¿Qué otro usuario aparece como propietario de muchos procesos? ¿Qué usuario es ese y por qué tiene tantos procesos?

```

kike 4157 0.0 0.0 319204 14704 ? S1 00:07 0:00 /usr/lib/gvfsd-recent --spawner :1.17 /org/gtk/gvfs/exec_spaw/4
kike 4241 0.0 0.0 161752 7712 ? S1 00:08 0:00 /usr/lib/libreoffice/program/oosplash
kike 4279 3.7 2.2 9480552 465356 ? S1 00:08 1:38 /usr/lib/libreoffice/program/soffice.bin --splash-pipe=5
root 6236 0.0 0.0 0 0 ? I 00:23 0:00 [kworker/6:0-mm_percpu_wq]
root 6542 0.0 0.0 0 0 ? I 00:26 0:00 [kworker/u16:2-events_unbound]
root 6900 0.0 0.0 0 0 ? I 00:28 0:00 [kworker/4:0-cgroup_destroy]
root 7393 0.0 0.0 0 0 ? I 00:29 0:00 [kworker/u16:4-events_unbound]
root 7723 0.0 0.0 0 0 ? I 00:34 0:00 [kworker/0:0-mm_percpu_wq]
root 8170 0.0 0.0 0 0 ? I 00:36 0:00 [kworker/6:1-events]
root 8173 0.0 0.0 0 0 ? I 00:36 0:00 [kworker/3:1-events]
root 8780 0.0 0.0 0 0 ? I 00:39 0:00 [kworker/5:1-mm_percpu_wq]
root 8802 0.0 0.0 0 0 ? I 00:41 0:00 [kworker/2:0-events]
root 8803 0.0 0.0 0 0 ? I 00:41 0:00 [kworker/1:0-events]
root 8809 0.0 0.0 0 0 ? I 00:41 0:00 [kworker/7:0-cgroup_destroy]
kike 8824 0.2 0.3 645832 69056 ? Rsl 00:42 0:01 /usr/lib/gnome-terminal-server
kike 8846 0.1 0.0 15244 8892 pts/0 Ss 00:42 0:01 zsh
kike 8850 0.0 0.0 12388 4048 pts/0 S 00:42 0:00 zsh
kike 8868 0.0 0.0 14492 5104 pts/0 S 00:42 0:00 zsh
kike 8869 0.0 0.0 14476 3684 pts/0 S 00:42 0:00 zsh
kike 8871 0.0 0.0 4576 1644 pts/0 Sl 00:42 0:00 /home/kike/.cache/gitstatus/gitstatusd-linux-x86_64 -G v1.5.4 -s -1 -u -1 -
root 9384 0.0 0.0 0 0 ? I 00:43 0:00 [kworker/0:1]
root 9578 0.0 0.0 0 0 ? I 00:45 0:00 [kworker/4:1-mm_percpu_wq]
root 9593 0.0 0.0 0 0 ? I 00:47 0:00 [kworker/6:2]
root 9594 0.0 0.0 0 0 ? I 00:47 0:00 [kworker/7:1-events]
root 9596 0.0 0.0 0 0 ? I 00:47 0:00 [kworker/5:0-inet_frag_wq]
root 9600 0.0 0.0 0 0 ? I 00:47 0:00 [kworker/2:2]
root 9623 0.0 0.0 0 0 ? I 00:51 0:00 [kworker/u16:1-events_power_efficient]
kike 9624 0.0 0.0 12580 3396 pts/0 R+ 00:51 0:00 ps aux

```

El usuario que más procesos tiene en ejecución es `root`, ya que ejecuta muchos procesos a bajo nivel como sistema operativo.

4. Ejecuta el comando `ps -u root`. Esto mostrará solo los procesos del sistema operativo. Ahora ejecuta el comando `ps -u`

```

> ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
kike    1198  0.0  0.0 160864  7880 tty2      Ssl+  feb21   0:00 /usr/lib/gdm-wayland-session /usr/bin/gnome-session
kike    1204  0.0  0.0 304720 20380 tty2      Sl+   feb21   0:00 /usr/lib/gnome-session-binary
kike    8846  0.1  0.0 15228  8872 pts/0    Ss   00:42   0:01 zsh
kike    8850  0.0  0.0 12388  4048 pts/0    S   00:42   0:00 zsh
kike    8868  0.0  0.0 14492  5104 pts/0    S   00:42   0:00 zsh
kike    8869  0.0  0.0 14476  3684 pts/0    S   00:42   0:00 zsh
kike    8871  0.0  0.0  4576  1644 pts/0    Sl   00:42   0:00 /home/kike/.cache/gitstatus/gitstatusd-linux-x86_64 -G v1.5.4 -s -1 -u -1 -
kike   10677  0.0  0.0 12580  3436 pts/0    R+   01:04   0:00 ps -u

```

El comando `ps -u` parece que me ha mostrado solo los procesos que usa mi usuario `kike`.

5. Ejecuta el comando `ps -u root | wc -l`. El símbolo `|` es una tubería, sirve para "pasarle" el resultado de un comando a otro comando distinto. En este caso estamos "pasándole" la salida de `ps -u root` al comando `wc -l`, que cuenta el número de líneas. ¿Cuántos procesos está ejecutando `root`?

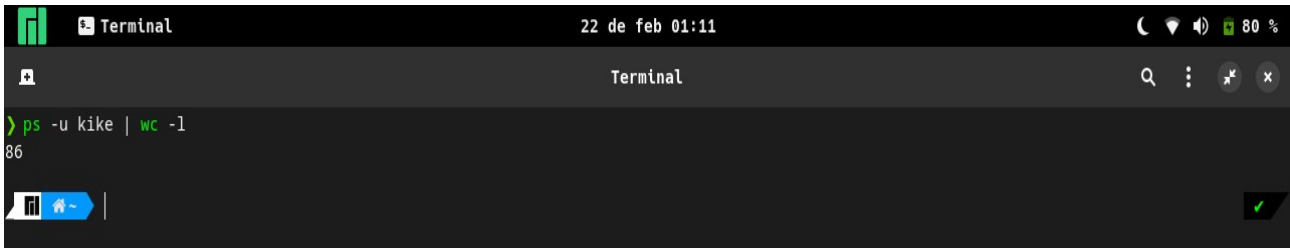
```

> ps -u root | wc -l
160

```

Está ejecutando 160 procesos.

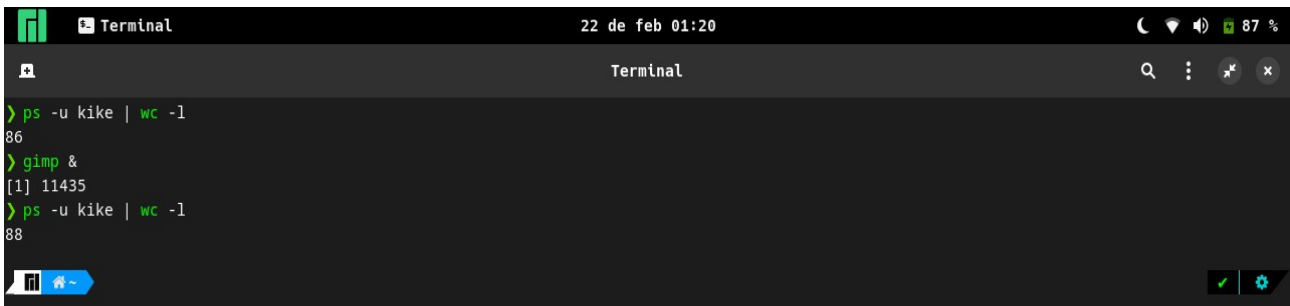
6. Averigua cuantos procesos estás ejecutando tú (tu usuario). ¿Qué comando has usado? ¿Qué valor que ha mostrado?



```
Terminal 22 de feb 01:11
Terminal
> ps -u kike | wc -l
86
```

He puesto `ps -u kike | wc -l` y me ha mostrado que hay 86 procesos ejecutándose como mi usuario.

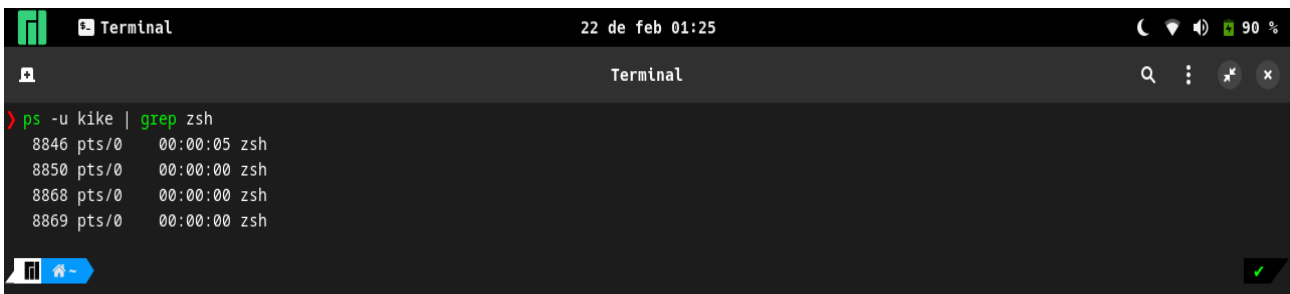
7. Ahora abre otro programa, como por ejemplo gimp, y vuelve a ejecutar el comando anterior. ¿Qué valor ha mostrado ahora?



```
Terminal 22 de feb 01:20
Terminal
> ps -u kike | wc -l
86
> gimp &
[1] 11435
> ps -u kike | wc -l
88
```

Ahora ha mostrado 88 procesos en ejecución.

8. Ejecuta el comando `ps -u <tuusuario> | grep bash`. El comando 'grep' funciona como filtro y muestra solo las líneas que contienen la palabra indicada. En este caso, al combinarlo con `ps` y la tubería, nos sirve para ver fácilmente qué procesos `bash` se están ejecutando. ¿Cuántos procesos `bash` aparecen? Abre otra terminal (sin cerrar la actual) y vuelve a ejecutar el comando. Debería aparecer otro proceso `bash`. ¿Es así?



```
Terminal 22 de feb 01:25
Terminal
> ps -u kike | grep zsh
8846 pts/0 00:00:05 zsh
8850 pts/0 00:00:00 zsh
8868 pts/0 00:00:00 zsh
8869 pts/0 00:00:00 zsh
```

Aparecen 4 procesos `bash`. Pongo `bash` pero en realidad son `zsh` ya que mi terminal no es `bash`.



```
Terminal 22 de feb 01:27
Terminal
> ps -u kike | grep zsh
8846 pts/0 00:00:05 zsh
8850 pts/0 00:00:00 zsh
8868 pts/0 00:00:00 zsh
8869 pts/0 00:00:00 zsh
11985 pts/1 00:00:00 zsh
11990 pts/1 00:00:00 zsh
12006 pts/1 00:00:00 zsh
12007 pts/1 00:00:00 zsh
```

En la segunda terminal que he abierto aparecen los procesos de la primera terminal más los procesos de la segunda, total 8 procesos `bash` en ejecución.

9. Ejecuta el comando `ps -u <tuusuario> | grep firefox` para averiguar cuántos procesos firefox se están ejecutando. Si no tienes firefox abierto, ábrelo y vuelve a probar. ¿Cuántos procesos son?

```
Terminal
22 de feb 01:35
Terminal
> ps -u kike | grep firefox
12255 pts/0    00:00:09 firefox
```

En este caso me sale solo 1 proceso firefox.

10. Averigua cuántos procesos 'chrome' se están ejecutando. Si no tienes chrome o chromium abierto, ejecútalo y abre varias pestañas de Internet. ¿Cuántos procesos son? ¿Qué comando has utilizado?

```
Terminal
22 de feb 01:40
Terminal
> ps -u kike | grep chrome
12811 ?          00:00:13 chrome
12822 ?          00:00:00 chrome_crashpad
12824 ?          00:00:00 chrome_crashpad
12830 ?          00:00:00 chrome
12831 ?          00:00:00 chrome
12835 ?          00:00:00 chrome
12859 ?          00:00:04 chrome
12860 ?          00:00:02 chrome
12871 ?          00:00:00 chrome
12999 ?          00:00:04 chrome
13080 ?          00:00:02 chrome
13138 ?          00:00:02 chrome
13222 ?          00:00:00 chrome
```

Son 13 procesos. He usado el comando `ps -u kike | grep chrome` para ver si veía algo más de información pero podría haber usado `ps -u kike | grep chrome | wc -l` para no tener que contar.

11. Averigua cuántos procesos 'office' se están ejecutando. ¿Cuántos procesos son? ¿Qué comando has utilizado?

```
Terminal
22 de feb 02:09
Terminal
> ps -u kike | grep office
4279 ?          00:06:21 soffice.bin
```

Es solo un proceso y he usado el comando `ps -u kike | grep office`.

12. Ejecuta el comando `top`. Observa la información que se muestra e indica:

```
Terminal
22 de feb 02:17
Terminal
top - 02:17:42 up 3:03, 1 user, load average: 0,22, 0,41, 0,43
Tasks: 249 total, 1 running, 248 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1,1 us, 0,4 sy, 0,0 ni, 98,2 id, 0,0 wa, 0,2 hi, 0,1 si, 0,0 st
MiB Mem : 19916,5 total, 14977,2 free, 1628,2 used, 3311,1 buff/cache
MiB Swap: 0,0 total, 0,0 free, 0,0 used. 17764,7 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 1288 kike       20   0 5306472 292936 135468 S   7,9   1,4   13:53.33 gnome-shell
 8824 kike       20   0 648032 72036 49280 S   4,3   0,4    0:31.49 gnome-terminal-
 1183 kike       20   0 413408 26556 12568 S   0,7   0,1    1:27.59 openrazer-daemo
  186 root        0 -20    0      0      0 I   0,3   0,0    0:01.41 kworker/5:1H-events_highpri
  806 root       20   0 247528 12896 8372 S   0,3   0,1    0:02.03 upowerd
 4279 kike       20   0 9618824 601668 219344 S   0,3   3,0    6:42.49 soffice.bin
12631 root       20   0      0      0      0 I   0,3   0,0    0:00.78 kworker/5:2-events
15336 kike       20   0 13292 4120 3384 R   0,3   0,0    0:00.07 top
   1 root       20   0 169208 13828 10104 S   0,0   0,1    0:03.27 systemd
   2 root       20   0      0      0      0 S   0,0   0,0    0:00.01 kthreadd
   3 root       0 -20    0      0      0 I   0,0   0,0    0:00.00 rcu_gp
```

a. Cuantas tareas en total se están ejecutando: 249, cuantas están corriendo (running): 1, y cuantas están dormidas (sleeping): 248.

b. Cuanta memoria RAM total hay instalada en el sistema: 20,88 GB, cuanta está siendo utilizada: 5,179 GB, y cuanta está libre (en GB): 15,7 GB.

c. Cual es el proceso que utiliza más CPU (esto puede variar cada segundo y depende de qué programa estés usando en cada momento): gnome-shell.

d. Cual es el proceso que utiliza más memoria RAM: soffice.bin.

Cuando quieras puedes parar el comando top con CTRL+C.

13. Utilizando top, indica cuanta CPU y RAM está utilizando firefox (si es necesario, ábrelo).

Ahora, en esa ventana de firefox, abre varias pestañas con varios videos de YouTube reproduciéndose a la vez (dos o tres debería bastar). Vuelve a ejecutar el comando top e indica cuánta CPU y RAM está utilizando ahora (debería ser más que antes). No cierres firefox todavía.

```
Firefox 22 de feb 02:56
Terminal
top - 02:56:22 up 3:41, 1 user, load average: 1,04, 1,20, 0,83
Tasks: 261 total, 1 running, 260 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,4 us, 0,2 sy, 0,0 ni, 99,2 id, 0,0 wa, 0,1 hi, 0,1 si, 0,0 st
MiB Mem : 19916,5 total, 14325,1 free, 1943,1 used, 3648,3 buff/cache
MiB Swap: 0,0 total, 0,0 free, 0,0 used, 17271,1 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 19978 kike       20   0 3284420 411104 251884 S   1,3   2,0   0:07.00 firefox
 1183 kike       20   0 413408   26556 12568 S   1,0   0,1   1:46.85 openrazer-daemo
 1288 kike       20   0 5349676 297296 136636 S   1,0   1,5 16:17.99 gnome-shell
19976 kike       20   0  13292   4016   3284 R   0,7   0,0   0:00.18 top
 1743 kike       20   0 852916   86728 49272 S   0,3   0,4   6:56.82 Xwayland
 4279 kike       20   0 9618828 598456 216076 S   0,3   2,9   8:23.15 soffice.bin
17170 root        20   0      0      0      0 I   0,3   0,0   0:01.14 kworker/4:1-mm_percpu_wq
19557 kike       20   0 646772   67824 49324 S   0,3   0,3   0:01.70 gnome-terminal-
    1 root        20   0 169208   13836 10104 S   0,0   0,1   0:03.51 systemd
```

En la primera está usando 1,3% de CPU y 2.0% de RAM

```
Firefox 22 de feb 02:53
Terminal
top - 02:53:03 up 3:38, 1 user, load average: 1,55, 1,30, 0,76
Tasks: 267 total, 1 running, 266 sleeping, 0 stopped, 0 zombie
%Cpu(s): 14,7 us, 3,7 sy, 0,0 ni, 80,7 id, 0,0 wa, 0,7 hi, 0,2 si, 0,0 st
MiB Mem : 19916,5 total, 13860,2 free, 2366,1 used, 3690,2 buff/cache
MiB Swap: 0,0 total, 0,0 free, 0,0 used, 16795,8 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
18198 kike       20   0 3888152 469208 284708 S  58,9   2,3   0:48.77 firefox
18556 kike       20   0 2995860 345048 145544 S  52,3   1,7   0:36.21 Isolated Web Co
18432 kike       20   0 3012560 321260 142716 S  14,9   1,6   0:23.24 Isolated Web Co
18611 kike       20   0 545804   94772 80708 S   7,0   0,5   0:06.91 RDD Process
18613 kike       20   0 485264   89188 74892 S   3,6   0,4   0:03.57 Utility Process
 1743 kike       20   0 852852   86728 49272 S   2,3   0,4   6:51.29 Xwayland
18196 kike       20   0  13292   4088   3352 R   1,0   0,0   0:00.56 top
 1422 kike       9 -11 2081656 33096 23872 S   0,7   0,2   0:03.60 pulseaudio
18436 kike       20   0 2486852 115628 95528 S   0,7   0,6   0:00.47 Isolated Web Co
18828 kike       20   0 2463664   90120 77436 S   0,7   0,4   0:00.18 Web Content
```

Y en la segunda está usando 58,9% de CPU y 2,3% de RAM

14. ¿Cual es el PID del proceso firefox del paso anterior?
PID 18198

15. Mata el proceso firefox del paso anterior con kill <PID> y espera a que se cierre (si por ejemplo el PID es 3799 sería con kill 3799). Si no funciona, envía la señal de apagado forzoso con kill -9 <PID> o kill -KILL <PID> (ambos hacen lo mismo). Si sigue sin funcionar, es posible que tengas que utilizar killall firefox. ¿Qué comandos has probado? ¿Qué ha sucedido funcionando? ¿Cual ha funcionado?

Con kill 18198 me ha matado el proceso.

Lo mejor es cerrar los programas de forma 'normal'. Pero cuando un proceso se queda 'colgado' y no responde, puedes utilizar kill o killall, como último recurso, para matarlo y que el sistema vuelva a funcionar correctamente. Hay que tener en cuenta que con kill o killall puedes perder la información del programa que no hayas guardado.