

GNU/Linux para programadores:

KILL & KILLALL: Asesinato en Primer Grado

Cuando recién te pasas a GNU/Linux uno de los problemas más frecuentes con el que te encuentras es que tarde o temprano, algún programa se “cuelga” y comienzas a experimentar el síndrome del Ctr + Alt + Supr heredado de tu anterior sistema privativo. Pero no sufras: existe una cura.

Escrito por: **Eugenia Bahit** (GLAMP Hacker & eXtreme Programmer)



Eugenia es **Arquitecta de Software**, docente e instructora de tecnologías **GLAMP** (GNU/Linux, Apache, MySQL, Python y PHP) y **Agile coach** (UTN) especializada en Scrum y **eXtreme Programming**. Miembro de la **Free Software Foundation**, **The Linux Foundation** y **Debian Hackers**. Creadora de **python-printr**, **Europio Engine** y colaboradora de **Vim**.

Webs:

Cursos de programación: www.cursosdeprogramacionadistancia.com

Web personal: www.eugeniabahit.com

Redes sociales:

Twitter / Identi.ca: [@eugeniabahit](https://twitter.com/eugeniabahit)

Según la *distro* que utilices, los programas que instales, las fuentes desde las cuales los instales, los recursos de tu ordenador y la forma en la que “maltrates” a tu Sistema Operativo, los programas -en modo gráfico generalmente- experimentarán “cuelgues” que, si recién te inicias en GNU/Linux, provocarán la reinstalación de tu sistema al grito de diversos improperios. Pero... tranquilo: no es grave y no necesitas reinstalar el sistema cuando algo no resulta como lo esperabas. Por el contrario, cerrar “por la fuerza” un programa “colgado” es mucho más simple de lo que te imaginas.

Ahora, antes de continuar, te haré la misma recomendación que hago constantemente a mis alumnos: **siempre, deja una terminal abierta y maximizada**.

La diferencia entre un Sistema Operativo privativo popular y uno libre como el Sistema

Operativo GNU con Linux, es que este último te permite mantener un control absoluto sobre él mismo, haciéndolo independiente de su interfaz gráfica. Y no. Que no hayas pagado por tu *distro*, no significa que la diferencia sea de precio. De hecho, cuando hablamos de libre y privativo, simplemente hablamos de tu libertad para controlar lo que tienes y no del precio de lo que adquieres. Y en los Sistemas Operativos GNU, dicho control, lo tienes por intermedio de la terminal en el 99,9% de los casos.

Entonces ¿para qué cambiarte de Sistema Operativo si no vas a hacer uso de sus ventajas? Si intentas utilizar GNU/Linux como tu antiguo Sistema Operativo privativo, fracasarás en la experiencia. Así que lo mejor que puedes hacer, es iniciar una amistad con la terminal y dejarla siempre abierta y maximizada.

Tener la terminal abierta te ayudará a muchísimas cosas, pero por sobre todo, te facilitará la vida cuando un programa, eventualmente, se quede congelado.

¿Se colgó? Entonces ¡mátalo!

El tema es que se congeló tu programa ¿entonces...? Entonces, es hora de asesinar a un proceso. Para ello, lo primero que necesitas es conocer el nombre del proceso. Éste, generalmente, es similar al nombre del programa. Por ejemplo, el proceso correspondiente al navegador *Firefox*, se llama *firefox*.

Pero no siempre sabrás el nombre con exactitud y para ello, hay una solución: los procesos que están corriendo en tu sistema, puedes obtenerlos con el comando `ps` con los siguientes argumentos:

```
user@host:~$ ps aux
```

El comando `ps` arrojará una lista similar a la siguiente:

```
root      2073  0.0  0.0  15124  1428 ?        Ss   Apr04  0:02 sendmail: MTA: accepting connections
eugenia   2075  0.0  0.3  76264  7672 ?        Sl   Apr04  0:00 telepathy-indicator
eugenia   2079  0.0  0.3  42152  6112 ?        Sll  Apr04  0:01 /usr/lib/telepathy/mission-control-5
eugenia   2084  0.0  0.2  91208  5284 ?        Sl   Apr04  0:00 /usr/lib/gnome-online-accounts/goa-daemon
root      2147  0.0  0.2  37764  5392 ?        Ss   Apr04  0:02 /usr/sbin/apache2 -k start
root      2188  0.0  0.0   4632   812 tty1    Ss+  Apr04  0:00 /sbin/getty -8 38400 tty1
www-data  2190  0.0  0.3  39108  5948 ?        S    Apr04  0:00 /usr/sbin/apache2 -k start
www-data  2191  0.0  0.3  39108  5948 ?        S    Apr04  0:00 /usr/sbin/apache2 -k start
www-data  2192  0.0  0.3  39108  5948 ?        S    Apr04  0:00 /usr/sbin/apache2 -k start
www-data  2193  0.0  0.3  39108  5948 ?        S    Apr04  0:00 /usr/sbin/apache2 -k start
www-data  2194  0.0  0.3  39108  5948 ?        S    Apr04  0:00 /usr/sbin/apache2 -k start
eugenia   2195  0.0  0.3  55688  6528 ?        Sl   Apr04  0:14 zeitgeist-datahub
eugenia   2201  0.0  0.2  44324  4380 ?        Sl   Apr04  0:01 /usr/bin/zeitgeist-daemon
eugenia   2207  0.0  0.2  52072  5124 ?        Sl   Apr04  0:01 /usr/lib/zeitgeist/zeitgeist-fts
eugenia   2215  0.0  0.0   5448   252 ?        S    Apr04  0:00 /bin/cat
eugenia   2216  0.0  0.5  73068  10056 ?       Sl   Apr04  0:03 gnome-screensaver
```

Solo que la lista, será muchísimo más extensa. Así que un método corto para obtener el programa que buscas, será aplicar un filtro sobre esa salida con el comando `grep` y utilizando como patrón de búsqueda el nombre del programa o parte de él:

```
user@host:~$ ps aux | grep gedit
```

Obtendrás algo como lo siguiente:

```
eugenia 18519 16.5 1.9 121956 38600 ? Sl 15:51 0:01 gedit
eugenia 18542 0.0 0.0 5628 836 pts/0 S+ 15:51 0:00 grep --color=auto gedit
```

El dato que te interesa, será generalmente el del primer resultado que aparezca en la lista:

```
eugenia 18519 16.5 1.9 121956 38600 ? Sl 15:51 0:01 gedit
```

De ese registro, dos columnas serán las que deberás observar: la de ID del proceso (PID):

```
eugenia 18519 16.5 1.9 121956 38600 ? Sl 15:51 0:01 gedit
```

Y la del nombre del comando (COMMAND):

```
eugenia 18519 16.5 1.9 121956 38600 ? Sl 15:51 0:01 gedit
```

Con esa información ya puedes “asesinar” al proceso, ya sea con el comando `killall` más el nombre del proceso:

```
killall COMMAND
```

O con el comando `kill` más la ID del proceso:

```
kill PID
```

Siguiendo el ejemplo anterior, para matar el proceso por nombre ejecutaríamos:

```
killall gedit
```

Y para matarlo por su ID:

```
kill 18519
```

Para verificar que un proceso haya terminado efectivamente, puedes ejecutar `killall` con el argumento `-w`:

```
killall -w COMMAND
```

Si el proceso finalizó, verás un mensaje de error como el siguiente:

```
killall -w gedit  
gedit: proceso no encontrado
```

Es muy importante que para esta verificación utilices `killall` y no `kill`, ya que podría suceder que tras matar un proceso se reinicie de forma automática y para entonces, su PID habrá cambiado.