



## Control de procesos

### Los procesos

Se llaman procesos a todos los programas que se están ejecutando en el sistema. Son muchos más de los que podría parecer, puesto que no sólo hay que tener en cuenta los programas que ejecuta cada usuario, sino los que son necesarios para el correcto funcionamiento del sistema. Hay que considerar, por ejemplo, los *daemons*, que permanecen a la espera de ciertas señales para responder y desencadenar otros procesos.

Cada proceso tiene asignado un número que lo identifica, llamado sencillamente "PID". Mediante este número, es posible enviar **señales** a los procesos.

### Listado de procesos

El programa *ps* lista los procesos en ejecución. Si se ejecuta sin parámetros, sólo dirá los procesos asociados al usuario que invoca al programa. Pero ejecutado con las opciones *auxw* (cada letra es una opción), listará todos. A continuación se muestra un ejemplo (ligeramente retocado) de la salida de *ps auxw*:

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	COMMAND
root	1	0.0	0.2	1020	464	?	init
root	152	0.0	0.3	1352	640	?	/sbin/syslogd
root	154	0.0	0.5	1428	848	?	/sbin/klogd
root	168	0.0	0.2	1048	444	?	/usr/sbin/gpm -m /dev/psaux -t ps2
root	173	0.0	0.3	1300	548	?	/usr/sbin/inetd
root	179	0.0	0.3	1352	564	?	/usr/sbin/lpd
postgres	207	0.0	0.7	5752	1160	?	/usr/lib/postgresql/bin/postmaster
root	213	0.0	0.5	2088	960	?	proftpd (accepting connections)
daemon	230	0.0	0.3	1140	556	?	/usr/sbin/atd
root	233	0.0	0.3	1168	620	?	/usr/sbin/cron
root	238	0.0	1.0	3320	1700	?	/usr/sbin/apache
root	246	0.0	0.3	1116	600	?	/sbin/cardmgr
curso	258	0.0	0.8	2076	1328	tty1	-bash
pedro	259	0.0	1.0	2424	1680	tty2	-bash
root	260	0.0	0.2	1004	456	tty3	/sbin/getty 38400 tty3
root	261	0.0	0.2	1004	456	tty4	/sbin/getty 38400 tty4
www-data	268	0.0	1.0	3344	1708	?	/usr/sbin/apache
root	358	0.0	0.7	2220	1176	?	nmbd -a
curso	607	0.0	0.5	1748	844	tty1	sh /usr/bin/X11/startx
curso	615	0.0	0.4	2172	776	tty1	xinit /home/curso/.xinitrc
root	616	3.1	2.6	8644	4280	?	X :0 -auth /home/curso/.Xauthority
curso	620	2.7	2.8	7348	4524	tty1	panel
curso	622	0.8	1.4	3988	2264	tty1	sawmill
curso	629	1.3	2.2	6560	3564	?	asclock_applet
curso	631	1.3	2.2	6616	3648	?	multiloader_applet
curso	639	1.6	2.3	6652	3768	?	gnomepager_applet
curso	644	1.7	2.2	6216	3696	tty1	gnome-terminal
curso	647	0.0	0.3	1144	580	tty1	gnome-ptty-helper
curso	648	0.0	0.7	1992	1208	pts/0	bash
curso	650	0.0	0.7	2868	1180	pts/0	ps aux

### Eliminar un proceso

Cada usuario puede eliminar (se podría decir *matar*) un proceso que le pertenezca. Será necesario hacerlo si se ha perdido el control del programa. Para ello se le manda la señal *kill* (la número 9), con el programa *kill*. Por ejemplo, para matar el proceso *gnome-terminal*, que tiene PID 644, se tecla la orden `kill -9 644`

### Ver la evolución

Con el programa *top* se puede ver cómo van evolucionando los distintos procesos, además de una estadística de la carga del sistema. En la ilustración se ve un ejemplo.

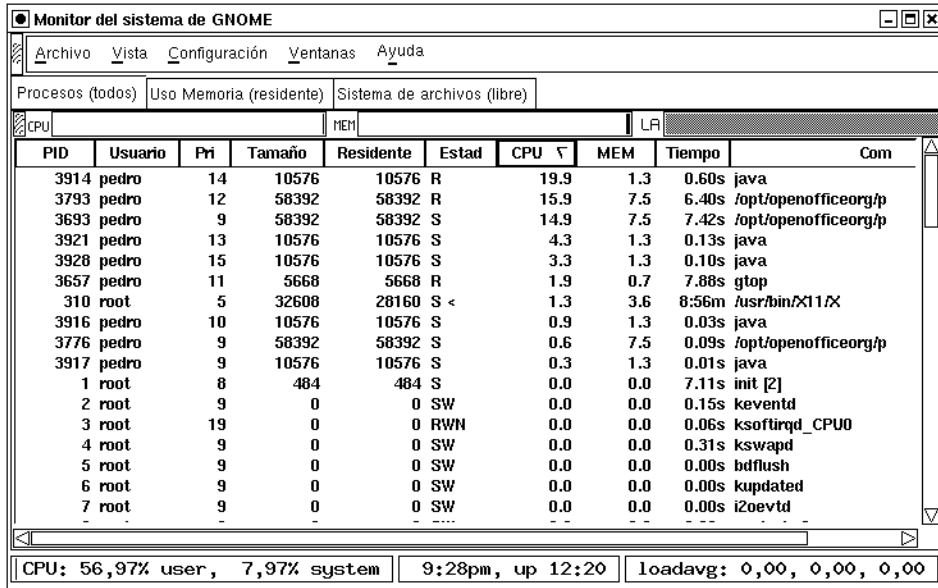
```

Terminal
11:19pm up 5:15, 4 users, load average: 0.12, 0.04, 0.01
65 processes: 62 sleeping, 3 running, 0 zombie, 0 stopped
CPU states: 34.6% user, 5.3% system, 0.0% nice, 60.0% idle
Mem: 160696K av, 155500K used, 5196K free, 66864K shrd, 75212K buff
Swap: 64256K av, 360K used, 63896K free

  PID USER   PRI  NI  SIZE  RSS SHARE STAT  LIB %CPU %MEM  TIME COMMAND
  678 root    14   0  7772 5528 1984 R    0 10.0  3.4  0:04 XF86_SVGA
  728 curso   10   0  6380 6376 5492 R    0  7.4  3.9  0:01 ksnapshot
  714 curso   3   0  1308 1308  704 R    0  1.5  0.8  0:03 top
  736 curso   5   0  4824 4824 4584 S    0  1.1  3.0  0:00 kdeinit
  734 curso   2   0  4772 4772 4540 S    0  0.5  2.9  0:00 kdeinit
  701 curso   0   0  3752 3752 2784 S    0  0.3  2.3  0:00 gnomepage
    
```

## Monitor del sistema de GNOME

Es la versión GNOME de *top*, aunque incorpora muchas mejoras. Se pueden ver los procesos en ejecución, colocarlos por orden, mandarles señales (con el menú de contexto), etc. Se arranca eligiendo en el **menú GNOME**, sección **Sistema**, la entrada **Monitor del sistema**; o por su nombre de programa, *gtop*. Este es un ejemplo:



The screenshot shows the 'Monitor del sistema de GNOME' window. It has a menu bar with 'Archivo', 'Vista', 'Configuración', 'Ventanas', and 'Ayuda'. Below the menu bar are three tabs: 'Procesos (todos)', 'Uso Memoria (residente)', and 'Sistema de archivos (libre)'. The main area displays a table of processes with columns for PID, Usuario, Pri, Tamaño, Residente, Estad, CPU, MEM, Tiempo, and Com. The status bar at the bottom shows 'CPU: 56,97% user, 7,97% system', '9:28pm, up 12:20', and 'loadavg: 0,00, 0,00, 0,00'.

PID	Usuario	Pri	Tamaño	Residente	Estad	CPU	MEM	Tiempo	Com
3914	pedro	14	10576	10576	R	19.9	1.3	0.60s	java
3793	pedro	12	58392	58392	R	15.9	7.5	6.40s	/opt/openofficeorg/p
3693	pedro	9	58392	58392	S	14.9	7.5	7.42s	/opt/openofficeorg/p
3921	pedro	13	10576	10576	S	4.3	1.3	0.13s	java
3928	pedro	15	10576	10576	S	3.3	1.3	0.10s	java
3657	pedro	11	5668	5668	R	1.9	0.7	7.88s	gtop
310	root	5	32608	28160	S <	1.3	3.6	8:56m	/usr/bin/X11/X
3916	pedro	10	10576	10576	S	0.9	1.3	0.03s	java
3776	pedro	9	58392	58392	S	0.6	7.5	0.09s	/opt/openofficeorg/p
3917	pedro	9	10576	10576	S	0.3	1.3	0.01s	java
1	root	8	484	484	S	0.0	0.0	7.11s	init [2]
2	root	9	0	0	SW	0.0	0.0	0.15s	keventd
3	root	19	0	0	RWN	0.0	0.0	0.06s	ksoftirqd_CPU0
4	root	9	0	0	SW	0.0	0.0	0.31s	kswapd
5	root	9	0	0	SW	0.0	0.0	0.00s	bdflush
6	root	9	0	0	SW	0.0	0.0	0.00s	kupdated
7	root	9	0	0	SW	0.0	0.0	0.00s	iZoevtd
-	-	-	-	-	----	--	--	---	-

