

# GESTIÓN DE PROCESOS CRÍTICOS (LINUX)

Cristóbal Suárez Abad

ADMINISTRACIÓN DE SISTEMAS OPERATIVOS - 2º ASIR

## Índice:

Introducción:.....	2
1) Identifica los procesos activos en el sistema y registra al menos cinco con su información más relevante. (systemd, bash, NetworkManager, sshd, udevd) .....	4
2) Filtra la lista de procesos para mostrar solo los pertenecientes a un usuario concreto, a un nombre determinado o a un PID específico. ....	6
3) Analiza qué procesos consumen más recursos y determina si pertenecen al sistema o al usuario. Registra los tres procesos más exigentes y describe su comportamiento. ....	8
4) Crea dos procesos que te permitan observar diferencias en el consumo de recursos:.....	9
5) Observa:.....	10
6) Documenta las fases del ciclo de vida de uno de los procesos. ....	11
7) Finaliza los procesos y documenta las fases que ha atravesado durante su ciclo de vida.12	
8) Clasifica los procesos observados en tres grupos: del sistema, de usuario y demonios. ....	13
9) Desde GNOME, abre el monitor del sistema. Captura una vista general de la pestaña de procesos e identifica visualmente los de mayor consumo. ....	14
10) Añade a GNOME la extensión System-monitor, busca el paquete, instalalo y muestra su funcionamiento. ....	15
11) Accede a la herramienta de administración web <b>Webmin (WebAdmin)</b> y localiza el apartado de gestión de procesos. ....	16
Observa cómo se listan los procesos activos y qué información adicional muestra respecto a la terminal. ....	18
Comprueba si puedes detener, reiniciar o cambiar la prioridad desde la interfaz. ...	20
Comenta las ventajas e inconvenientes de usar una herramienta web frente al uso directo de la línea de comandos. ....	23

## Introducción:

1. Identifica los procesos activos en el sistema y registra al menos cinco con su información más relevante. (systemd, bash, NetworkManager, sshd, udevd)
  - PID
  - Usuario propietario
  - Estado aproximado (en ejecución, en espera, detenido)
  - Porcentaje de CPU y memoria utilizada

verifica si son procesos del sistema o de usuario y describe brevemente su función principal.

2. Filtra la lista de procesos para mostrar solo los pertenecientes a un usuario concreto, a un nombre determinado o a un PID específico.
3. Analiza qué procesos consumen más recursos y determina si pertenecen al sistema o al usuario. Registra los tres procesos más exigentes y describe su comportamiento.
4. Crea dos procesos que te permitan observar diferencias en el consumo de recursos:
  - Proceso ligero: por ejemplo, abrir un editor de texto o ejecutar una orden simple.
  - Proceso intensivo: lanzar una tarea de cálculo o compresión que mantenga el procesador ocupado.
5. Observa:
  - Qué PID se les asigna.
  - Cómo varía el consumo de CPU y memoria entre ambos.
  - Cómo afecta el cambio de prioridad del proceso intensivo al rendimiento general del sistema.
6. Documenta las fases del ciclo de vida de uno de los procesos
7. Finaliza los procesos y documenta las fases que ha atravesado durante su ciclo de vida.
8. Clasifica los procesos observados en tres grupos: del sistema, de usuario y demonios.

9. Desde GNOME, abre el monitor del sistema. Captura una vista general de la pestaña de procesos e identifica visualmente los de mayor consumo.
10. Añade a GNOME la extensión System-monitor, busca el paquete, instálalo y muestra su funcionamiento.
11. Accede a la herramienta de administración web **Webmin (WebAdmin)** y localiza el apartado de gestión de procesos.
  - Observa cómo se listan los procesos activos y qué información adicional muestra respecto a la terminal.
  - Comprueba si puedes detener, reiniciar o cambiar la prioridad desde la interfaz.
  - Comenta las ventajas e inconvenientes de usar una herramienta web frente al uso directo de la línea de comandos.

1) Identifica los procesos activos en el sistema y registra al menos cinco con su información más relevante. (systemd, bash, NetworkManager, sshd, udevd)

- *PID*
- *Usuario propietario*
- *Estado aproximado (en ejecución, en espera, detenido)*
- *Porcentaje de CPU y memoria utilizada*

Usamos el comando “ps aux”. En este caso, como se pide los 5 primeros:

ps aux | head -n 6

```
crisobal@ubuntumysqlsuarez:~$ ps aux | head -n 6
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.6  0.3 22084 13016 ?        Ss   11:25   0:02 /sbin/init
root         2  0.0  0.0      0     0 ?        S    11:25   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    11:25   0:00 [pool_workqueue_release]
root         4  0.0  0.0      0     0 ?        I<   11:25   0:00 [kworker/R-rcu_g]
root         5  0.0  0.0      0     0 ?        I<   11:25   0:00 [kworker/R-rcu_p]
```

Los valores de “STAT” son<sup>1</sup>:

- *“Running or Runnable (R)”*
- *Uninterruptible Sleep (D)*
- *Interruptable Sleep (S)*
- *Stopped (T)*
- *Zombie (Z)*
- *Idle (I)”*

“aux” significa<sup>2</sup>:

*“a = show processes for all users*

*u = display the process's user/owner*

*x = also show processes not attached to a terminal”*

<sup>1</sup> <https://www.baeldung.com/linux/process-states>

<sup>2</sup> <https://unix.stackexchange.com/questions/106847/what-does-aux-mean-in-ps-aux>

Verifica si son procesos del sistema o de usuario y describe brevemente su función principal.

Son procesos del sistema.

USER	PID	COMMAND	Tipo de Proceso	Función Principal
root	1	/sbin/init	Sistema	Es el primer proceso que se inicia (PID 1). Gestiona el arranque del sistema y es el <b>padre de todos los demás procesos</b> . Es crucial para la inicialización y gestión de servicios del sistema (sistema <b>init</b> o <b>systemd</b> ).
root	2	[kthreadd]	Sistema	Es el proceso del <b>kernel</b> responsable de la gestión y creación de otros <b>hilos del kernel</b> (kernel threads). Se ejecuta en el espacio del kernel.
root	3	[pool_workqueue_release]	Sistema	Un hilo del <b>kernel</b> asociado a la gestión de colas de trabajo (workqueues), que son mecanismos para ejecutar tareas en el contexto del kernel de forma asíncrona.
root	4	[kworker/R-rcu_g]	Sistema	Un hilo de <b>trabajo del kernel</b> (kworker) dedicado a tareas específicas, en este caso, relacionado con el mecanismo <b>RCU</b> (Read-Copy-Update), que es una técnica de sincronización del kernel.
root	5	[kworker/R-rcu_p]	Sistema	Otro hilo de <b>trabajo del kernel</b> (kworker) asociado al mecanismo <b>RCU</b> , posiblemente una parte del proceso de <i>poda</i> (RCU grace period).

- 2) Filtra la lista de procesos para mostrar solo los pertenecientes a un usuario concreto, a un nombre determinado o a un PID específico.

Se utilizan los comandos ps junto con grep o herramientas específicas como pgrep

**ps aux | grep [s]shd**

**ps aux | grep sshd**

```
crisobal@ubuntumysqlsuarez:~$ ps aux | grep [s]shd
root      881  0.0  0.2 12020 8064 ?        Ss   11:25   0:00 sshd: /usr/sbin/sshd -D [listener]
0 of 10-100 startups
root      1194  0.0  0.2 15084 10496 ?        Ss   11:27   0:00 sshd: crisobal [priv]
crisobal+ 1252  0.0  0.1 15084 6988 ?        S    11:27   0:00 sshd: crisobal@pts/0
crisobal@ubuntumysqlsuarez:~$ ps aux | grep sshd
root      881  0.0  0.2 12020 8064 ?        Ss   11:25   0:00 sshd: /usr/sbin/sshd -D [listener]
0 of 10-100 startups
root      1194  0.0  0.2 15084 10496 ?        Ss   11:27   0:00 sshd: crisobal [priv]
crisobal+ 1252  0.0  0.1 15084 6988 ?        S    11:27   0:00 sshd: crisobal@pts/0
crisobal+ 1425  0.0  0.0 6544 2304 pts/0    S+   11:34   0:00 grep --color=auto sshd
crisobal@ubuntumysqlsuarez:~$
```

Resultado: Muestra la línea completa del proceso sshd. (Se usan corchetes [s] para evitar que el grep se muestre a sí mismo).

Filtro por usuario:

**ps -u root**

**ps -u crisobal**

```
crisobal@ubuntumysqlsuarez:~$ ps -u crisobal
  PID TTY          TIME CMD
 1128 ?           00:00:00 systemd
 1129 ?           00:00:00 (sd-pam)
 1142 tty1       00:00:00 bash
 1252 ?           00:00:00 sshd
 1253 pts/0       00:00:00 bash
 1437 pts/0       00:00:00 ps
crisobal@ubuntumysqlsuarez:~$
```

Filtro por PID Específico (PID 1, que es systemd):

**ps -p 1**

```
crisobal@ubuntumysqlsuarez:~$ ps -p 1
  PID TTY          TIME CMD
    1 ?           00:00:02 systemd
crisobal@ubuntumysqlsuarez:~$ |
```

- 3) Analiza qué procesos consumen más recursos y determina si pertenecen al sistema o al usuario. Registra los tres procesos más exigentes y describe su comportamiento.

Usamos **htop** para monitorizar el uso.

1.3% Load average: 0.00 0.02 0.04  
 Mem[|||||] 567M/3.82G Uptime: 00:13:59  
 Swp[ ] 0K/3.82G

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1458	cristobal	20	0	8504	4736	3584	R	1.3	0.1	0:01.47	htop
891	mysql	20	0	1744M	384M	36480	S	0.7	9.8	0:04.89	mysqld
1252	cristobal	20	0	15084	6988	5120	S	0.7	0.2	0:00.55	sshd: cristobal@pts/0
1	root	20	0	22084	13016	9304	S	0.0	0.3	0:02.22	init
310	root	19	-1	66760	16992	15968	S	0.0	0.4	0:00.38	systemd-journald
365	root	RT	0	282M	27264	8704	S	0.0	0.7	0:00.10	multipathd -d -s
379	root	20	0	282M	27264	8704	S	0.0	0.7	0:00.00	multipathd -d -s
380	root	RT	0	282M	27264	8704	S	0.0	0.7	0:00.00	multipathd -d -s
381	root	RT	0	282M	27264	8704	S	0.0	0.7	0:00.00	multipathd -d -s
382	root	RT	0	282M	27264	8704	S	0.0	0.7	0:00.00	multipathd -d -s
383	root	RT	0	282M	27264	8704	S	0.0	0.7	0:00.08	multipathd -d -s
384	root	RT	0	282M	27264	8704	S	0.0	0.7	0:00.00	multipathd -d -s
386	root	20	0	29188	7680	4992	S	0.0	0.2	0:00.23	systemd-udevd
538	systemd-ne	20	0	19008	9472	8320	S	0.0	0.2	0:00.06	systemd-networkd
550	systemd-re	20	0	21588	12800	10624	S	0.0	0.3	0:00.15	systemd-resolved
555	systemd-ti	20	0	91024	7680	6784	S	0.0	0.2	0:00.06	systemd-timesyncd
589	systemd-ti	20	0	91024	7680	6784	S	0.0	0.2	0:00.00	systemd-timesyncd
674	messagebus	20	0	9796	5376	4608	S	0.0	0.1	0:00.17	@dbus-daemon --system --address=

Pertenecen al usuario “cristobal”, “mysql” y “root” y a “systemd”.

Los tres procesos que más consumen son:

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1458	cristobal	20	0	8504	4736	3584	R	1.3	0.1	0:01.47	htop
891	mysql	20	0	1744M	384M	36480	S	0.7	9.8	0:04.89	mysqld
1252	cristobal	20	0	15084	6988	5120	S	0.7	0.2	0:00.55	sshd: cristobal@pts/0

El propio “htop”, el gestor de bases de datos “mysqld”, y la conexión “ssh” con la que estamos accediendo a la máquina. En el tiempo que los hemos observado, se han mantenido estables y con un consumo de recursos bajo.



## 5) Observa:

- *Qué PID se les asigna.*

Al bloc de notas: **1625**.

```
cristobal@ubuntumysqlsuarez:~$ nano ligero.txt &
[1] 1625
```

A la compresión: **1675**.

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1675	cristobal	20	0	3556	2048	1408	R	100.3	0.1	0:07.85	gzip -9 debian-13.1.0-amd64-netinst

- *Cómo varía el consumo de CPU y memoria entre ambos.*

El de bloc de notas es minúsculo, mientras que la compresión aprovecha toda la capacidad de la CPU.

- *Cómo afecta el cambio de prioridad del proceso intensivo al rendimiento general del sistema.*

Para cambiar la prioridad, usamos<sup>3</sup>:

**sudo renice 19 -p 1675**

El resultado: en este caso apenas se nota la diferencia. Puede ser porque en el sistema no se estaba ejecutando ningún otro proceso que demandase mucho.

<sup>3</sup> <https://didweb.gitbooks.io/comandos-linux/content/chapter1/procesos/nice-y-renice.html>

## 6) Documenta las fases del ciclo de vida de uno de los procesos.

### Ciclo de Vida del Proceso y Finalización

Documentaremos el ciclo de vida del proceso intensivo (gzip, PID 1675).

Fases del Ciclo de Vida:

**Creación:** El shell (bash) crea un nuevo proceso (gzip) y el sistema le asigna el PID 1675. (Estado: R - Running).

**Ejecución:** El proceso está utilizando la CPU para comprimir el archivo. (Estado: R en top, %CPU ~100%).

**Espera:** Si la compresión tuviera que esperar a que el disco duro leyera o escribiera datos, pasaría brevemente a este estado. (Estado: D - Disk Sleep).

**Finalización:** La compresión finaliza por sí misma. El proceso sale de ejecución.

**Muerte:** Una vez finalizado, el proceso queda en un estado Z (Zombie) hasta que su proceso padre (bash en este caso) lee su código de salida. Una vez leído, el kernel elimina por completo el proceso.

7) Finaliza los procesos y documenta las fases que ha atravesado durante su ciclo de vida.

Para finalizar los procesos usamos "kill -9 1675".

```
root@ubuntumysqlsuarez:/home/cristobal# kill -9 1675
```

## 8) Clasifica los procesos observados en tres grupos: del sistema, de usuario y demonios.

### - **Del Sistema:**

Esenciales para el kernel y la inicialización.

Creados por root o el kernel. systemd (PID 1).

### - **De Usuario:**

Ejecutados bajo la cuenta del usuario para tareas interactivas o específicas.

bash, nano (ligero), gzip (intensivo), etc.

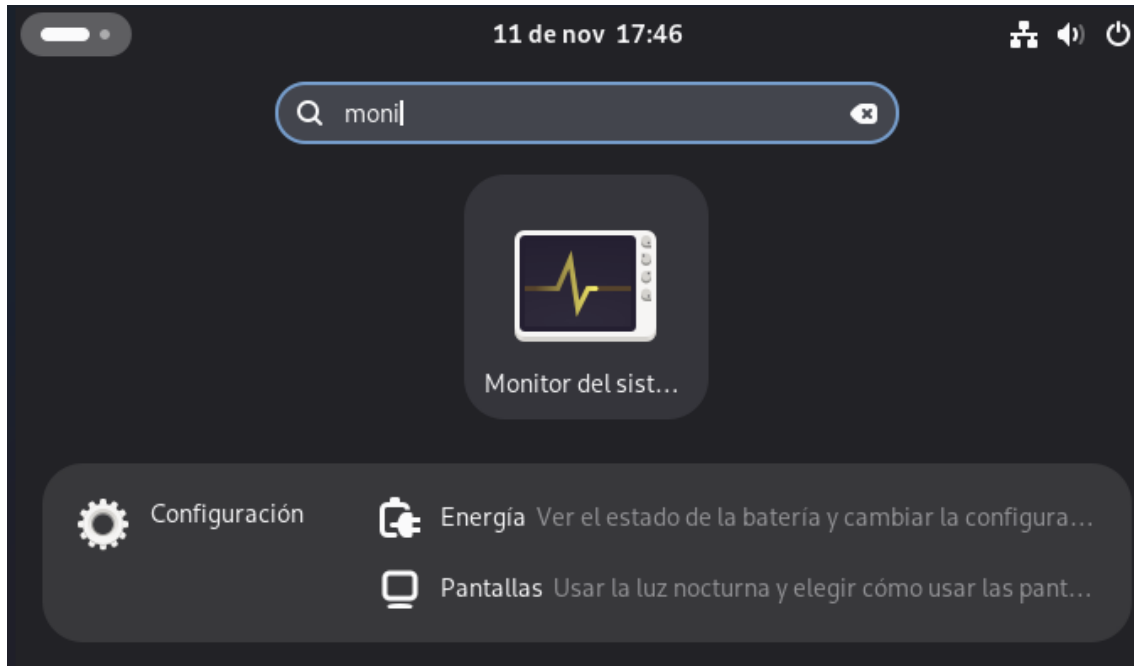
### - **Demonios (Daemons):**

Procesos del sistema que se ejecutan en segundo plano, sin interfaz interactiva, y ofrecen servicios.

sshd, udevd.

- 9) Desde GNOME, abre el monitor del sistema. Captura una vista general de la pestaña de procesos e identifica visualmente los de mayor consumo.

Buscamos el “**Monitor del Sistema**”.



Los que más consumen son el propio GNOME. Porque el sistema ahora mismo no está haciendo nada.

Buscar procesos y usuarios						
Nombre	ID	CPU	Memoria	Lectura de disco	Escritura en disco	
gnome-shell	3271	10,7 %	146,7 MB	—	—	
gnome-system-monitor	4128	9,0 %	127,3 MB	—	—	
ibus-engine-simple	3628	0,0 %	749,6 kB	—	—	
localsearch-extractor-3	4299	0,0 %	7,8 MB	—	—	
evolution-addressbook-factory	3993	0,0 %	4,5 MB	—	—	
evolution-calendar-factory	3963	0,0 %	3,9 MB	—	—	
dconf-service	3919	0,0 %	598,0 kB	—	—	

- 10) Añade a GNOME la extensión System-monitor, busca el paquete, instalalo y muestra su funcionamiento.

En nuestro caso ya estaba instalado, pero si hiciera falta<sup>4</sup>:

**sudo apt-get update**

**sudo apt-get install gnome-system-monitor**

Para lanzarlo:

**gnome-system-monitor**

Para desinstalarlo:

**sudo apt-get remove gnome-system-monitor**

---

<sup>4</sup> <https://vitux.com/how-to-install-and-use-gnome-system-monitor-and-task-manager-in-debian-10/>

- 11) Accede a la herramienta de administración web **Webmin (WebAdmin)** y localiza el apartado de gestión de procesos.

Instalamos Webmin<sup>5</sup>:

```
curl -o webmin-setup-repo.sh
```

```
https://raw.githubusercontent.com/webmin/webmin/master/webmin-setup-repo.sh
```

```
sudo sh webmin-setup-repo.sh
```

```
sudo apt-get install webmin --install-recommends
```

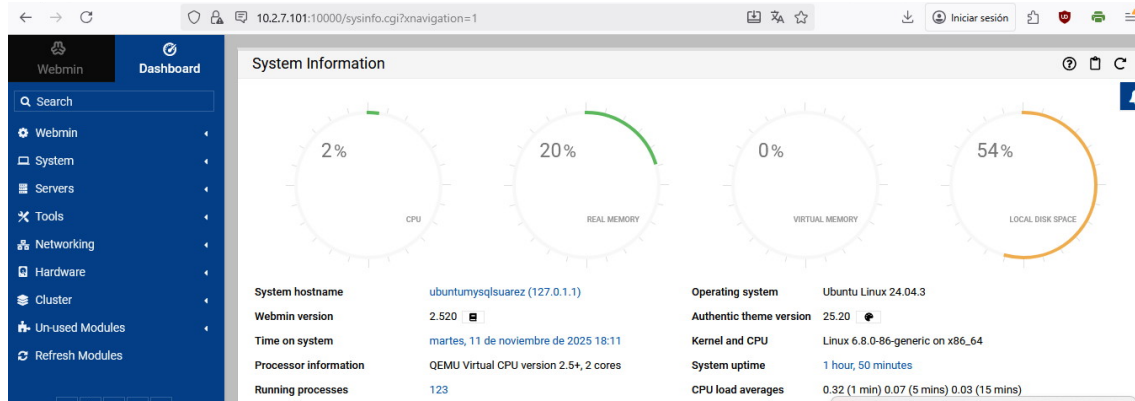
A veces tienes que hacer un “**apt update**”.

```
cristobal@ubuntumysqlsuarez:~$ curl -o webmin-setup-repo.sh https://raw.githubusercontent.com/webmin/webmin/master/webmin-setup-repo.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 17604  100 17604    0     0  62335      0 --:--:-- --:--:-- --:--:-- 62647
cristobal@ubuntumysqlsuarez:~$ sudo sh webmin-setup-repo.sh
[sudo] password for cristobal:
Setup Webmin releases repository? (y/N) y
  Downloading Webmin developers key ..
  .. done
  Installing Webmin developers key ..
  .. done
  Setting up Webmin releases repository ..
  .. done
  Cleaning repository metadata ..
  .. done
  Downloading repository metadata ..
  .. done
Webmin and Usermin can be installed with:
  apt-get install --install-recommends webmin usermin
cristobal@ubuntumysqlsuarez:~$ sudo apt-get install webmin --install-recommends
```

<sup>5</sup> <https://webmin.com/download/>

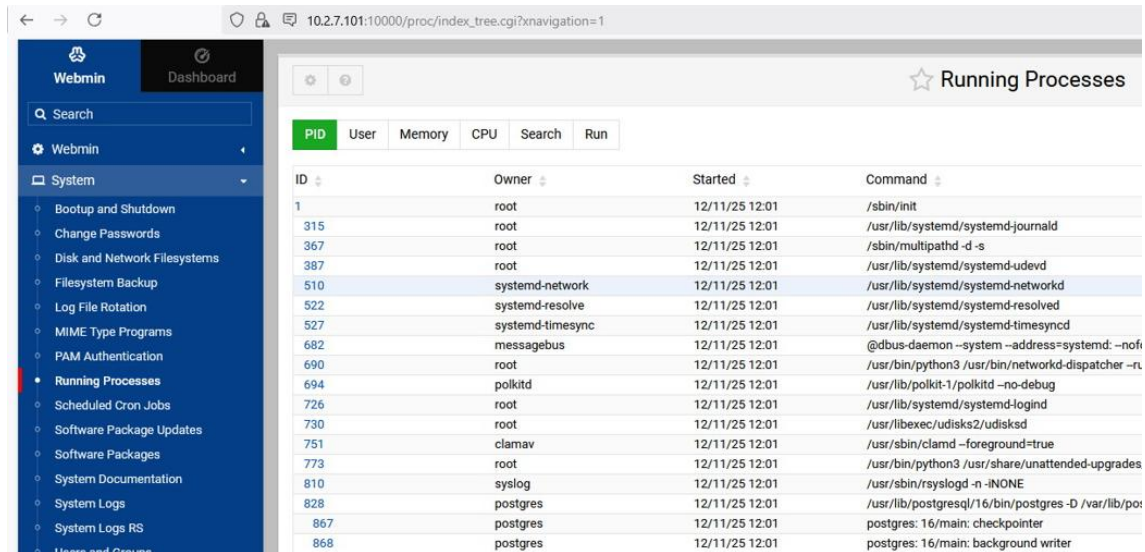
Una vez instalado, ponemos en el buscador:

<https://IP de la maquina:10000>



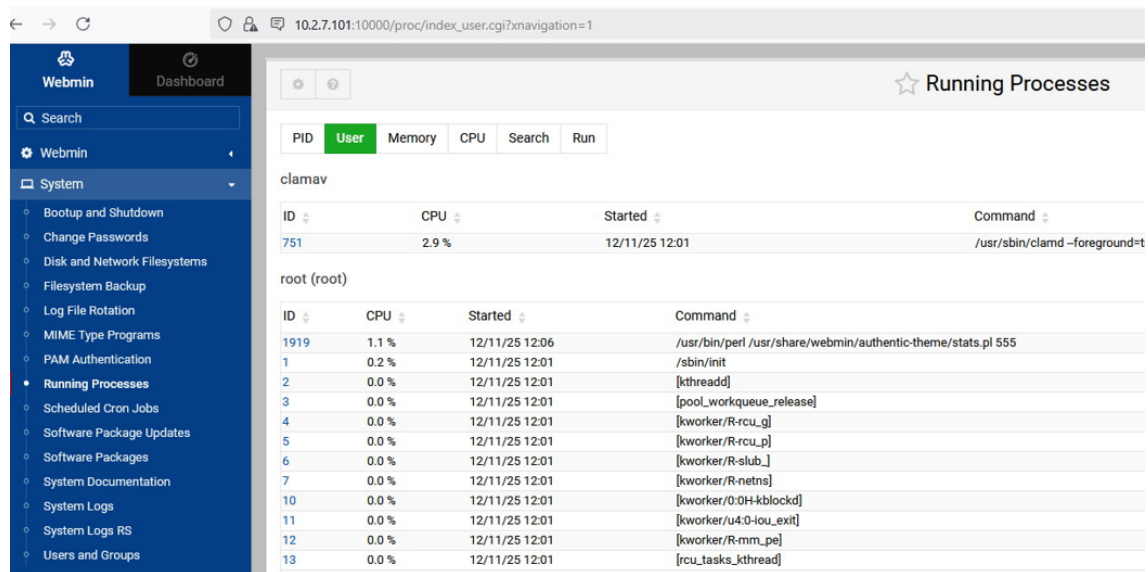
Observa cómo se listan los procesos activos y qué información adicional muestra respecto a la terminal.

Información general.



ID	User	Memory	CPU	Search	Run
1	root				/sbin/init
315	root				/usr/lib/systemd/systemd-journald
367	root				/sbin/multipathd -d -s
387	root				/usr/lib/systemd/systemd-udevd
510	systemd-network				/usr/lib/systemd/systemd-networkd
522	systemd-resolve				/usr/lib/systemd/systemd-resolved
527	systemd-timesync				/usr/lib/systemd/systemd-timesyncd
682	messagebus				@dbus-daemon --system --address=systemd: --nof
690	root				/usr/bin/python3 /usr/bin/networkd-dispatcher --r
694	polkitd				/usr/lib/polkit-1/polkitd --no-debug
726	root				/usr/lib/systemd/systemd-logind
730	root				/usr/libexec/udisks2/udisksd
751	clamav				/usr/sbin/clamd --foreground=true
773	root				/usr/bin/python3 /usr/share/unattended-upgrades
810	syslog				/usr/sbin/rsyslogd -n -iNONE
828	postgres				/usr/lib/postgresql/16/bin/postgres -D /var/lib/po
867	postgres				postgres: 16/main: checkpoint
868	postgres				postgres: 16/main: background writer

Por usuario.



ID	CPU	Started	Command
751	2.9 %	12/11/25 12:01	/usr/sbin/clamd --foreground=t

ID	CPU	Started	Command
1919	1.1 %	12/11/25 12:06	/usr/bin/perl /usr/share/webmin/authentic-theme/stats.pl 555
1	0.2 %	12/11/25 12:01	/sbin/init
2	0.0 %	12/11/25 12:01	[kthreadd]
3	0.0 %	12/11/25 12:01	[pool_workqueue_release]
4	0.0 %	12/11/25 12:01	[kworker/R-rcu_g]
5	0.0 %	12/11/25 12:01	[kworker/R-rcu_p]
6	0.0 %	12/11/25 12:01	[kworker/R-slub_]
7	0.0 %	12/11/25 12:01	[kworker/R-netns]
10	0.0 %	12/11/25 12:01	[kworker/0:0H-kblockd]
11	0.0 %	12/11/25 12:01	[kworker/u4:0-iou_exit]
12	0.0 %	12/11/25 12:01	[kworker/R-mm_pe]
13	0.0 %	12/11/25 12:01	[rcu_tasks_kthread]

RAM utilizada:

The screenshot shows the Webmin interface with the 'Running Processes' tab selected. The left sidebar shows the 'System' menu with 'Running Processes' highlighted. The main content area displays memory statistics and a table of running processes.

**Real memory:** 3.82 GiB total / 1.85 GiB free / 675.28 MiB cached

**Swap space:** 3.82 GiB total / 3.82 GiB free

PID	User	Memory	CPU	Search	Run
ID	Owner	Size	Command		
751	clamav	1.31 GiB	/usr/sbin/clamd --foreground=true		
846	mysql	384.73 MiB	/usr/sbin/mysqld		
1919	root	38.84 MiB	/usr/bin/perl /usr/share/webmin/authentic-theme/stats.pl 555		
1945	root	35.46 MiB	/usr/bin/perl /usr/share/webmin/miniserv.pl /etc/webmin/miniserv.conf		
1360	root	32.12 MiB	/usr/bin/perl /usr/share/webmin/miniserv.pl /etc/webmin/miniserv.conf		
828	postgres	32 MiB	/usr/lib/postgresql/16/bin/postgres -D /var/lib/postgresql/16/main -c config_...		
1361	root	29.12 MiB	/usr/bin/perl /usr/share/webmin/miniserv.pl /etc/webmin/miniserv.conf		
367	root	26.5 MiB	/sbin/multipathd -d -s		
773	root	22.37 MiB	/usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown -wa ...		
690	root	20.12 MiB	/usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers		
315	root	16.58 MiB	/usr/lib/systemd/systemd-journald		
730	root	13.25 MiB	/usr/libexec/udisks2/udisksd		
961	www-data	13.23 MiB	/usr/sbin/apache2 -k start		

CPU utilizada:

The screenshot shows the Webmin interface with the 'Running Processes' tab selected. The left sidebar shows the 'System' menu with 'Running Processes' highlighted. The main content area displays CPU load averages and a table of running processes.

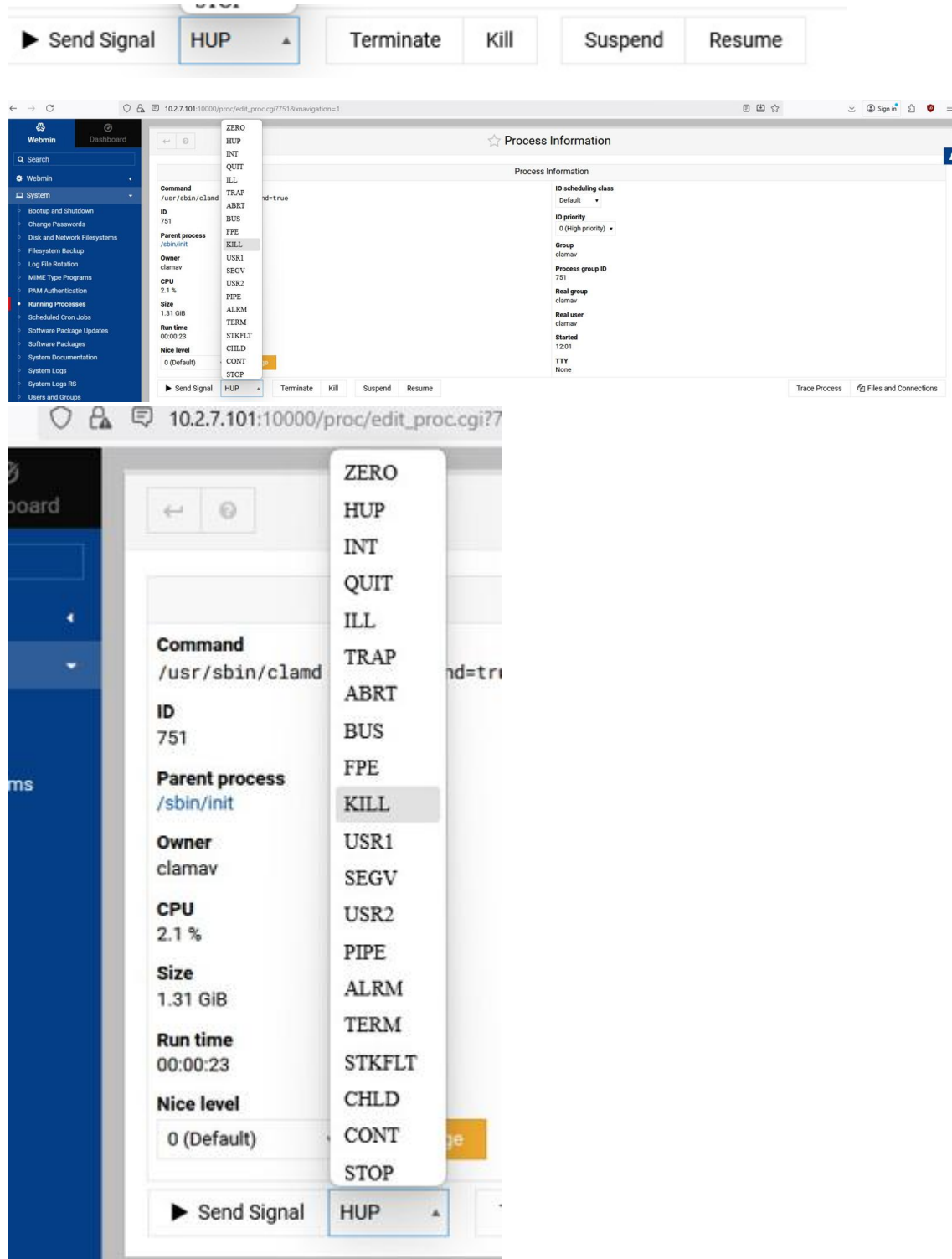
**CPU load averages:** 0.00 (1 mins), 0.02 (5 mins), 0.01 (15 mins)

**CPU type:** QEMU Virtual CPU version 2.5+ (2294 MHz), 2 cores

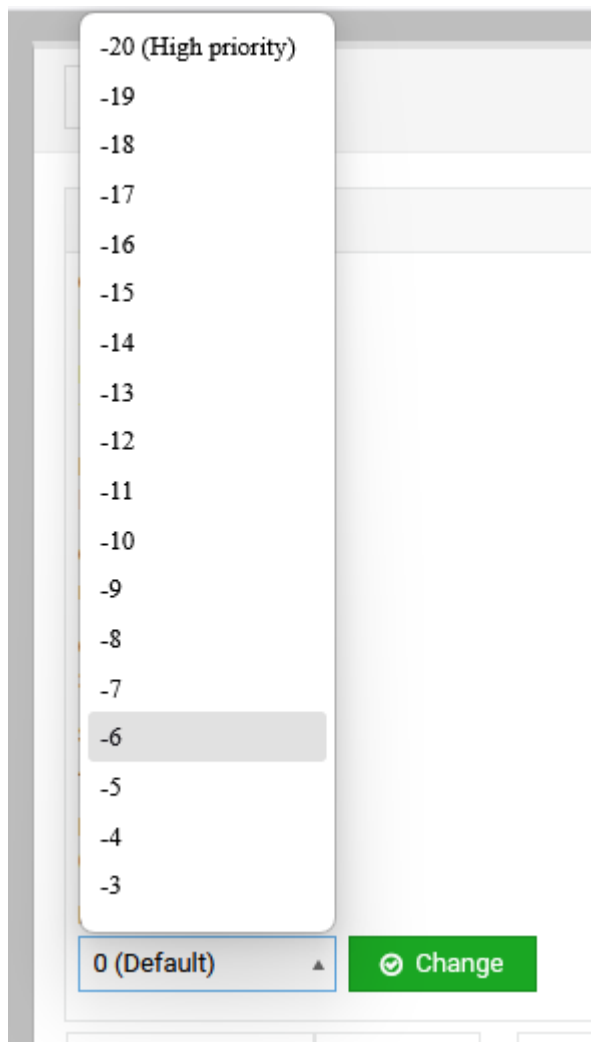
PID	User	Memory	CPU	Search	Run
ID	Owner	CPU	Command		
751	clamav	2.8 %	/usr/sbin/clamd --foreground=true		
1919	root	1.1 %	/usr/bin/perl /usr/share/webmin/authentic-theme/stats.pl 555		
846	mysql	1.0 %	/usr/sbin/mysqld		
1	root	0.2 %	/sbin/init		
2	root	0.0 %	[kthread]		
3	root	0.0 %	[pool_workqueue_release]		
4	root	0.0 %	[kworker/R-rcu_g]		
5	root	0.0 %	[kworker/R-rcu_p]		
6	root	0.0 %	[kworker/R-slab_]		
7	root	0.0 %	[kworker/R-netns]		
10	root	0.0 %	[kworker/0:0H-kblockd]		
11	root	0.0 %	[kworker/u4:0-iou_exit]		
12	root	0.0 %	[kworker/R-mm_pe]		
13	root	0.0 %	[kworker/R-mm_pe]		

Comprueba si puedes detener, reiniciar o cambiar la prioridad desde la interfaz.

Si se puede. Pinchamos en PID del proceso y nos llevará a una ventana donde podremos dar una serie de órdenes al proceso:



La prioridad también se puede modificar: tanto para CPU



Como para procesos Input/Output.

Process Information

IO scheduling class

Default ▾

IO priority

0 (High priority) ▲

0 (High priority)

1

2

3

4

5

6

7 (Low priority)

10188

Group

Comenta las ventajas e inconvenientes de usar una herramienta web frente al uso directo de la línea de comandos.

Ventajas:

- La herramienta web ofrece más información.
- No necesitas saberte los comandos.

Inconvenientes:

- Si cae la conexión te quedas no tienes acceso a la máquina.
- Hay siempre una pequeña "tardanza" a la hora de ejecutar los comandos.