



U3 ADMINISTRACIÓN DE SOFTWARE BASE

ACTIVIDAD PRÁCTICA 6: GESTIÓN DE PROCESOS

Sergio Cobo García
SISTEMAS INFORMÁTICOS

UNIDAD DIDÁCTICA 3 Administración de Software Base	ACTIVIDAD PRÁCTICA 6 Gestión de procesos
---	---

Tabla de contenido

Procesos en Windows.....	2
Procesos en Unix/Linux	6

Procesos en Windows

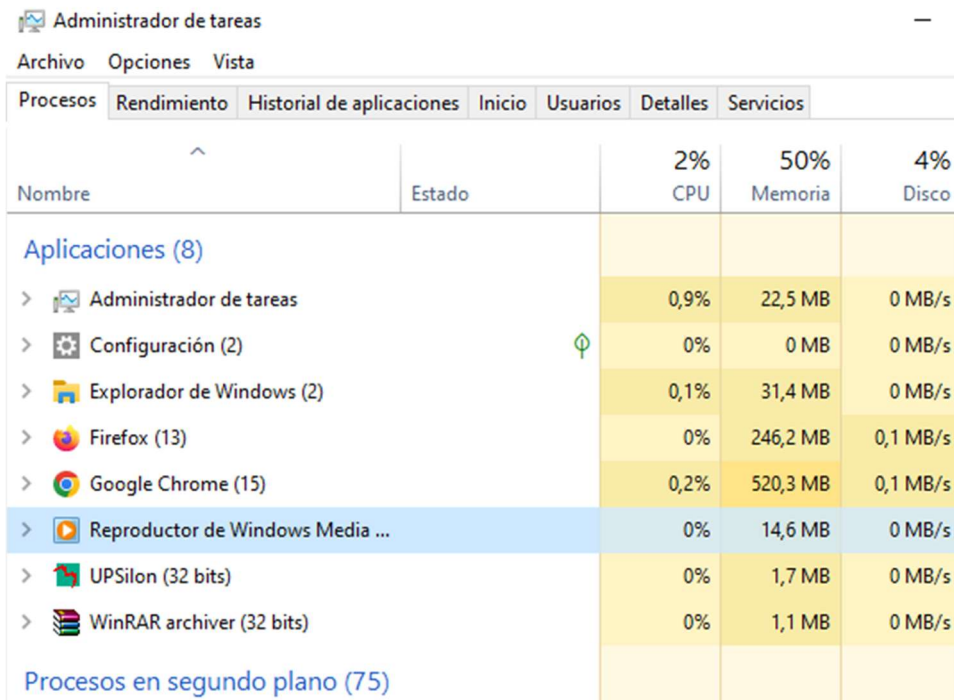
01	El administrador de tareas tiene diferentes puntos de entrada. Explicad todas las maneras posibles para acceder al administrador de tareas.
----	---

Las formas que hay para abrir el administrador de tareas son:

- Ctrl + Alt + Supr y seleccionamos Administrador de tareas.
- Click derecho encima de la barra de tareas y clicar en Administrador de tareas.
- Entrar en inicio y buscar Administrador de tareas.
- En la barra de búsqueda escribimos *taskmgr*.

02	Ejecutad un programa, como por ejemplo el Windows Media Player, e indicad el nombre del ejecutable, que consumo de CPU y que uso de memoria tiene.
----	--

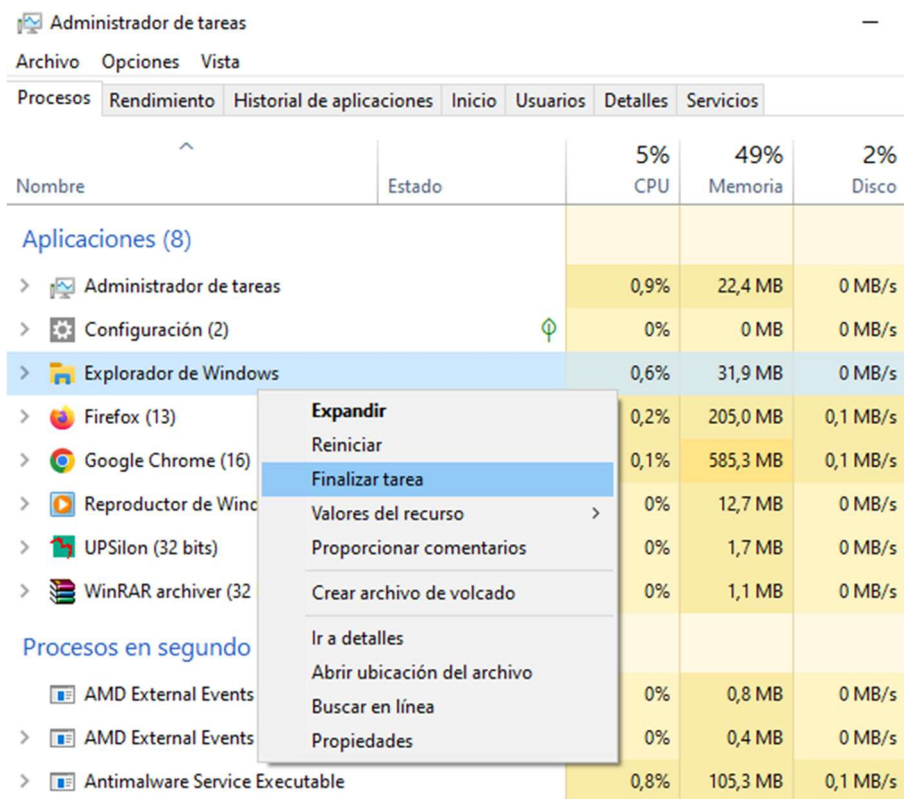
El Windows Media Player hace un consumo del 0% de la CPU y hace un uso de 14,6 MB de la memoria RAM.



Administrador de tareas				
Archivo Opciones Vista				
Procesos Rendimiento Historial de aplicaciones Inicio Usuarios Detalles Servicios				
Nombre	Estado	2% CPU	50% Memoria	4% Disco
Aplicaciones (8)				
> Administrador de tareas		0,9%	22,5 MB	0 MB/s
> Configuración (2)		0%	0 MB	0 MB/s
> Explorador de Windows (2)		0,1%	31,4 MB	0 MB/s
> Firefox (13)		0%	246,2 MB	0,1 MB/s
> Google Chrome (15)		0,2%	520,3 MB	0,1 MB/s
> Reproductor de Windows Media ...		0%	14,6 MB	0 MB/s
> UPSilon (32 bits)		0%	1,7 MB	0 MB/s
> WinRAR archiver (32 bits)		0%	1,1 MB	0 MB/s
Procesos en segundo plano (75)				

03	¿Como se elimina un proceso determinado? Elimina el proceso explorer y mirad que pasa. ¿Como podemos solucionar esta situación?
----	--

Para eliminar un proceso en el Administrador de tareas clicamos encima del proceso a eliminar y damos en Finalizar tarea.



04	¿Cuál es el proceso IDLE en Windows? ¿Cuál es su comportamiento? ¿Como podemos mostrar, mediante el comando <code>Get-Process</code> el proceso IDLE? Intentad eliminarlo, mediante el comando Stop-Process . ¿Qué pasa?
----	---

El proceso IDLE es un proceso especial ya que es el que se ejecuta cuando en el sistema no hay ningún otro proceso ejecutándose, sirve para representar el tiempo en el que en el sistema no hay ninguna tarea realizándose.

Para mostrar el proceso IDLE ejecutamos *Get-Process idle*.

```

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\alumne> Get-Process idle

Handles  NPM(K)  PM(K)  WS(K)  CPU(s)  Id  SI ProcessName
-----  -
0         0       60      8      0       0  0 Idle
  
```

Al intentar detener el proceso IDLE con *Stop-Process idle* nos mandará un mensaje de error, ya que no está permitida la detención ni eliminación del mismo al ser necesario para el correcto funcionamiento del sistema.

```
PS C:\Users\alumne> Stop-Process idle
Stop-Process : No se puede enlazar el parámetro 'InputObject'. No se puede convertir el
valor "idle" de tipo
"System.String" al tipo "System.Diagnostics.Process".
En línea: 1 Carácter: 14
+ Stop-Process idle
+ ~~~~~
+ CategoryInfo          : InvalidArgument: (:) [Stop-Process], ParameterBindingExcep
tion
+ FullyQualifiedErrorId : CannotConvertArgumentNoMessage,Microsoft.PowerShell.Comman
ds.StopProcessCommand
```

05

¿Como podemos mostrar, mediante el comando **Get-Process**, el usuario que ha ejecutado cada proceso y el pid de cada uno de los procesos?

El comando a usar sería el siguiente, aunque he probado otros comandos y me ha sido imposible mostrar el usuario que ha ejecutado cada proceso.

```
PS C:\Users\alumne> Get-Process | Select-Object ProcessName, Id, User

ProcessName      Id User
-----
ApplicationFrameHost 8060
conhost          5216
csrss             420
csrss             504
ctfmon           4720
```

He probado los siguientes comandos:

```
PS C:\Users\alumne> Get-WmiObject Win32_Process | Select-Object ProcessName, ProcessId, User

ProcessName      ProcessId User
-----
System Idle Process      0
System                  4
Registry                92
smss.exe              328
csrss.exe             420
wininit.exe           496
csrss.exe             504
winlogon.exe          596
```

```
PS C:\Users\alumne> Get-Process | Select-Object ProcessName, Id, @{Name="User";Expression={$_.GetOwner().User}}

ProcessName      Id User
-----
ApplicationFrameHost 8060
conhost          5216
csrss             420
csrss             504
ctfmon           4720
```

06

Indicad el significado de las columnas cuando lanzamos el comando **Get-Process**.

```
PS C:\Users\alumne> Get-Process

Handles  NPM(K)  PM(K)  WS(K)  CPU(s)  Id  SI ProcessName
-----
331      20      8156  11040  0,05    8060 1 ApplicationFrameHost
214      12      2888  15228  0,08    4280 1 backgroundTaskHost
435      32      11924 33804  0,39    8028 1 backgroundTaskHost
```

UNIDAD DIDÁCTICA 3 Administración de Software Base	ACTIVIDAD PRÁCTICA 6 Gestión de procesos
---	---

Las columnas son:

- Handles: Se utilizan para acceder a archivos, procesos... Cada vez que un proceso necesita acceder a un objeto, utiliza un handle para hacerlo.
- NPM: Muestra el número de objetos de memoria privada que está utilizando cada proceso. La memoria privada es la memoria asignada a un único proceso.
- PM: Sirve para saber la cantidad de memoria privada que están utilizando cada uno de los procesos.
- WS: Muestra la cantidad de memoria privada y compartida que está utilizando el proceso.
- CPU: Se refiere a la cantidad de tiempo que cada proceso ha estado ejecutándose en el procesador.
- Id: Es el identificador único de cada proceso en ejecución del sistema, en otras palabras, el PID.
- SI: Es la suma de Identificadores. Muestra la cantidad de recursos del sistema que están siendo utilizados por un proceso.
- ProcessName: Se refiere al nombre del proceso.

07	¿Como podemos enseñar, mediante el comando Get-Process , la versión del proceso que estemos ejecutando?
----	--

Podemos enseñar la versión de un proceso usando el siguiente comando.

```
PS C:\Users\alumne> Get-Process -Name CalculatorApp | Select-Object Version
Version
-----
```

En algunos casos como en el del ejemplo anterior no nos mostrará la versión del proceso. Puede darse el caso de que el proceso forme parte del mismo sistema operativo y no tenga una versión específica, como es el caso de la calculadora.

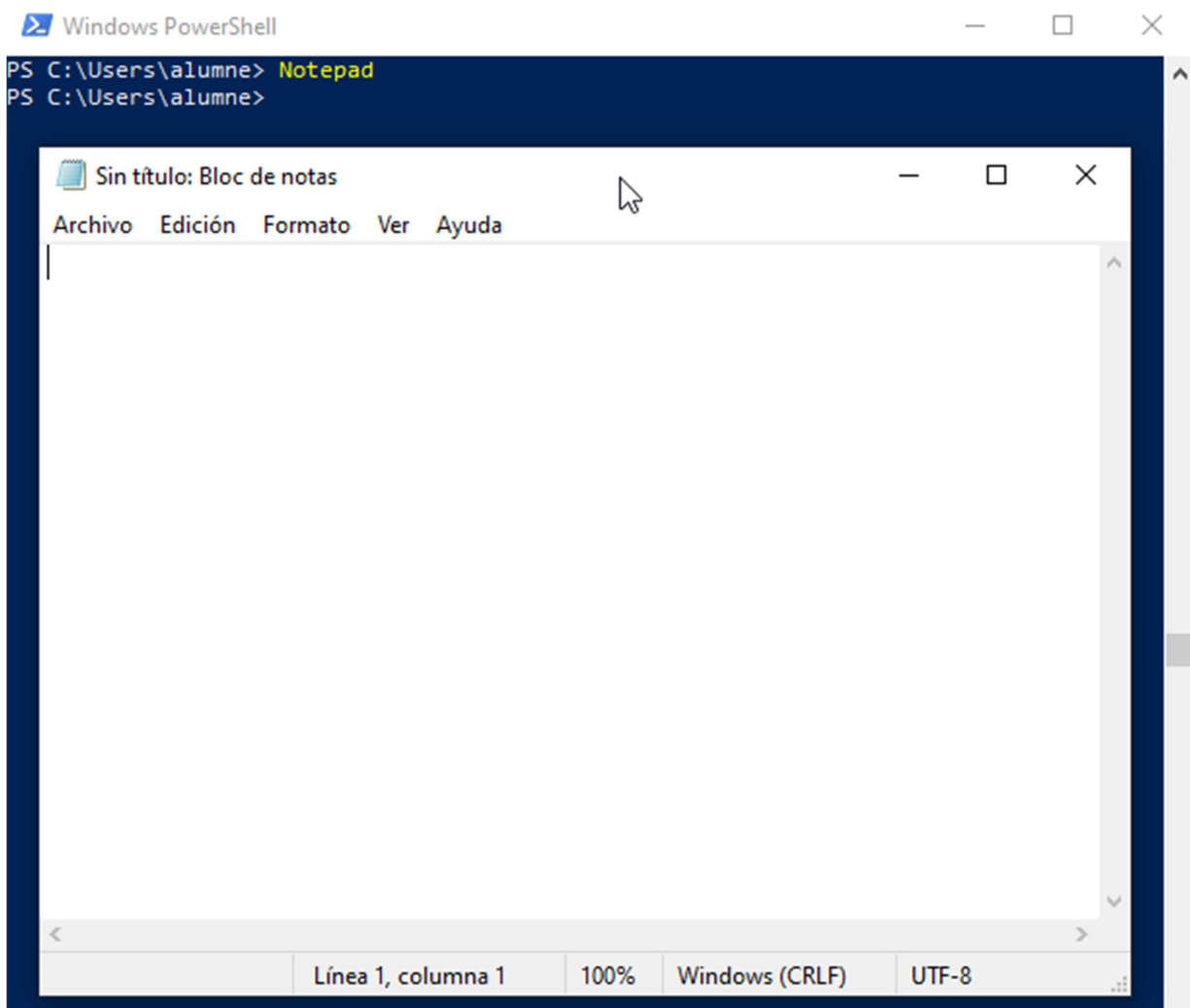
08	¿Como podemos mostrar, mediante el comando Get-process , todos los procesos que empiece su nombre con la cadena 'win' y 'ex'?
----	--

Podemos hacer la búsqueda de los procesos que empiecen por 'win' o 'ex' con el siguiente comando.

```
PS C:\Users\alumne> Get-Process -Name "win*", "ex*"
Handles  NPM(K)  PM(K)  WS(K)  CPU(s)  Id  SI ProcessName
-----
2016     79    51952   92716   10,34   4556 1 explorer
165      11     1360    1240    0.00    496 0 wininit
286      14     3056    4272    0.00    596 1 winlogon
```

09	Mediante la consola PowerShell, lanzad el programa 'Notepad'. A continuación, paradlo, también desde el PowerShell mediante el PID del programa.
----	---

Lanzamos el programa insertando *Notepad* o bien *notepad.exe*.



Para detener el Notepad usamos el comando *Stop-Process -Name notepad*.

```
PS C:\Users\alumne> Stop-Process -Name notepad
```

Procesos en Unix/Linux

10	Indicad que hace cada uno de los siguientes modificadores del comando ps : -l, -f, -u, -p, -ef, -e, -C, -o, -aux, -eo
----	---

Modificadores:

- -l: Lista el formato largo de la información de los procesos
- -f: Muestra la información completa de los procesos.
- -u: Muestra los procesos de un usuario concreto.
- -p: Permite especificar uno o más IDs de procesos para mostrar su información.
- -ef: Muestra la información completa de todos los procesos que se están ejecutando en el sistema.
- -e: Permite visualizar la información de todos los procesos del sistema.
- -C: Muestra la información sobre todos los procesos que están ejecutando un programa concreto.

- -o: Permite especificar los campos de la información de los procesos que el usuario desea que le muestre.
- -aux: Muestra información detallada de cada proceso.
- -eo: Tiene el mismo propósito que la opción -o pero además muestra la información solicitada sobre todos los procesos en el sistema.

11	Escribid un comando que liste todos los procesos el usuario root.
----	---

El comando que lo realiza es el `ps -u root`.

```

alumne@alumne-VirtualBox: ~
alumne@alumne-VirtualBox:~$ ps -u root
  PID TTY          TIME CMD
    1 ?            00:00:02 systemd
    2 ?            00:00:00 kthreadd
    3 ?            00:00:00 rcu_gp
    4 ?            00:00:00 rcu_par_gp
    5 ?            00:00:00 netns
    7 ?            00:00:00 kworker/0:0H-events_highpri
    9 ?            00:00:04 kworker/0:1H-events_highpri
   10 ?            00:00:00 mm_percpu_wq
   11 ?            00:00:00 rcu_tasks_kthread
   12 ?            00:00:00 rcu_tasks_rude_kthread
   13 ?            00:00:00 rcu_tasks_trace_kthread
   14 ?            00:00:00 ksoftirqd/0
   15 ?            00:00:03 rcu_preempt
   16 ?            00:00:00 migration/0
   17 ?            00:00:00 idle_inject/0
   18 ?            00:00:03 kworker/0:1-events
   19 ?            00:00:00 cpuhp/0
   20 ?            00:00:00 cpuhp/1
   21 ?            00:00:00 idle_inject/1
   22 ?            00:00:00 migration/1
   23 ?            00:00:00 ksoftirqd/1
   25 ?            00:00:00 kworker/1:0H-events_highpri

```

12	Entrad en el manual de la máquina y pulsad Ctrl+Z. ¿Qué ha pasado? Haced lo necesario para continuar la consulta del manual e indicadlo.
----	--

Con Ctrl+Z se suspende el proceso actual.

```

alumne@alumne-VirtualBox:~$ man man
[1]+  Detenido                  man man

```

Con `ps` muestro los procesos que se están ejecutando.

```

alumne@alumne-VirtualBox:~$ ps
  PID TTY          TIME CMD
 3181 pts/0        00:00:00 bash
 3190 pts/0        00:00:00 man
 3200 pts/0        00:00:00 pager
 3208 pts/0        00:00:00 ps

```

Con `fg` se reanuda la ejecución del proceso suspendido y lo trae al primer plano. Si hay más de un proceso suspendido usamos `fg %n`, donde `n` es el número del proceso que quieres reanudar.

```

alumne@alumne-VirtualBox:~$ fg
man man

```


13	<p>Iniciad el proceso man find y suspendedlo con Ctrl+Z.</p> <p>Ejecutad xclock en background. Utilizad jobs para listar los procesos en segundo plano y los procesos parados.</p> <p>Utilizad el comando fg para mandar el man find a primer plano. Salid normalmente con q.</p> <p>Utilizad fg para mandar xclock al foreground y acabadlo con Ctrl+C.</p> <p>Ejecutad xclock otra vez, pero esta vez en primer plano (no podréis usar la shell). Suspendedlo con Ctrl+Z y describid que pasa.</p>
----	--

Ejecutamos *man find* y hacemos Ctrl+Z para suspenderlo.

```
alumne@alumne-VirtualBox:~$ man find
[1]+  Detenido          man find
```

Iniciamos el xclock en segundo plano poniendo *xclock &*.

```
alumne@alumne-VirtualBox:~$ xclock &
[2] 3272
```

El comando *jobs* se utiliza para listar los procesos tanto en segundo plano como parados.

```
alumne@alumne-VirtualBox:~$ jobs
[1]+  Detenido          man find
[2]-  Ejecutando       xclock &
```

La orden *jobs -s* lista los procesos que se encuentran detenidos.

```
alumne@alumne-VirtualBox:~$ jobs -s
[1]+  Detenido          man find
```

Para mandar el man find a primer plano ejecutamos *fg man find*.

```
alumne@alumne-VirtualBox:~$ fg man find
man find
```

Mando el xclock a primer plano con *fg xclock*.

```
alumne@alumne-VirtualBox:~$ fg xclock
xclock
```

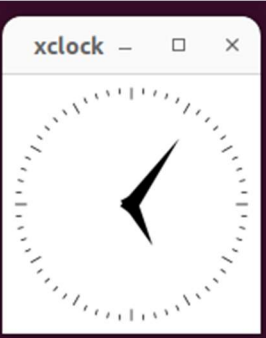


Seguidamente lo he parado con Ctrl+C y al hacer ps ya no aparece el proceso.

```
^C
alumne@alumne-VirtualBox:~$ ps
  PID TTY          TIME CMD
 2376 pts/0    00:00:00 bash
 2609 pts/0    00:00:00 ps
alumne@alumne-VirtualBox:~$
```

Vuelvo a abrir *xclock* y lo suspendo con Ctrl+Z. Me ha dejado suspenderlo sin problema.

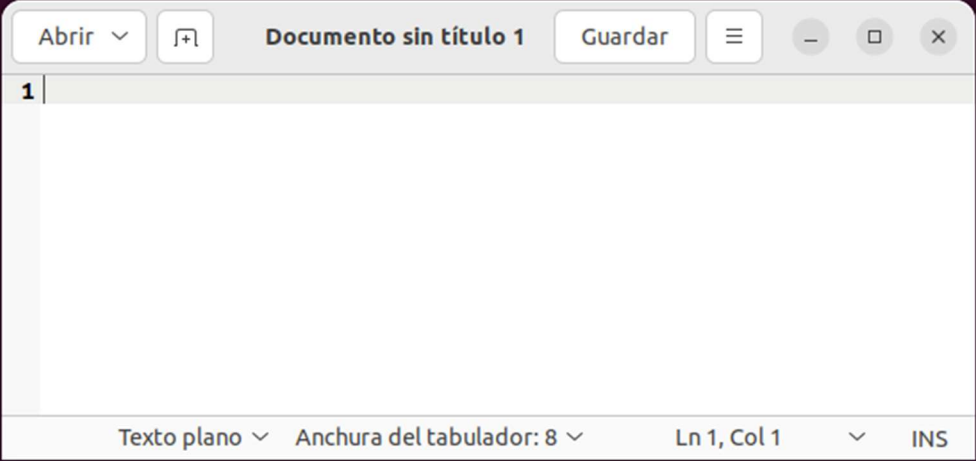
```
alumno@alumno-VirtualBox:~$ xclock
^Z
[1]+  Detenido                  xclock
alumno@alumno-VirtualBox:~$
```



- | | |
|----|---|
| 14 | Abrid un terminal y ejecutad el comando gedit . Desde otra terminal ejecutad ps y mirad quien es el proceso padre de gedit . Matad el padre de gedit . ¿Qué pasa? Volved a hacer lo mismo, pero en este caso ejecutad gedit en segundo plano. Matad el padre. ¿Qué pasa ahora? |
|----|---|

Ejecuto el comando *gedit* para abrir un nuevo documento.

```
alumno@alumno-VirtualBox:~$ gedit
```



Lanzo el comando *ps*, en él se muestra que el proceso padre de *gedit* es *systemd*.



Para matar el proceso padre del proceso gedit introducimos *kill systemd*. No nos permite eliminarlo porque systemd es un proceso esencial para el sistema operativo para que el sistema funcione correctamente.

```

alumne@alumne-VirtualBox:~$ kill systemd
bash: kill: systemd: los argumentos deben ser IDs de procesos o trabajos

```

Si lo volvemos a intentar a hacer, pero esta vez iniciando gedit en segundo plano seguirá sin dejarnos, ya que como lo he dicho anteriormente el proceso systemd es esencial para el sistema.

```

alumne@alumne-VirtualBox:~$ kill systemd
bash: kill: systemd: los argumentos deben ser IDs de procesos o trabajos

```

15	¿Como podríamos matar un proceso a partir de su número de tarea/job?
----	--

Para matar un proceso con su job usamos el comando *kill %NumeroJob* (por ejemplo, *kill %1*).

```

alumne@alumne-VirtualBox:~$ jobs
[1]+  Detenido          gedit
alumne@alumne-VirtualBox:~$ kill %1

[1]+  Detenido          gedit
alumne@alumne-VirtualBox:~$ 

```

16	Aplicad los parámetros necesarios, de forma interactiva, al comando <i>top</i> para que se muestren únicamente los procesos del usuario root, los ordene descendentemente por ocupación de memoria y actualice los datos cada 0.1 segundos.
----	---

El comando necesario para realizar esa búsqueda es *top -u root -o RES -d 0.1*.

```
alumno@alumno-VirtualBox:~$ top -u root -o RES -d 0.1

top - 23:17:05 up 2:48, 1 user, load average: 0,33, 0,14, 0,04
Tareas: 187 total, 1 ejecutar, 185 hibernar, 1 detener, 0 zombie
%Cpu(s): 0,0 us, 5,6 sy, 0,0 ni, 94,4 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 3923,5 total, 2063,5 libre, 832,5 usado, 1027,5 búfer/caché
MiB Intercambio: 4867,0 total, 4867,0 libre, 0,0 usado. 2862,0 dispon

  PID  USUARIO  PR  NI   VIRT   RES   SHR  S  %CPU  %MEM   HORA+  ORDEN
1616 root      20   0 385140 108168 63100 S   9,1   2,7   1:04.03 Xorg
2737 root      20   0 437360 63784 24104 S   0,0   1,6   0:01.65 fwupd
 695 root      20   0 874692 39552 18992 S   0,0   1,0   0:01.93 snapd
 728 root      20   0 126088 23196 14512 S   0,0   0,6   0:00.11 unatten+
1304 root      20   0 307552 20440 17524 S   0,0   0,5   0:00.20 package+
 687 root      20   0 269804 19272 16176 S   0,0   0,5   0:00.94 Network+
```

17 ¿Qué prioridad asigna el comando **nice** cuando no indicamos este valor de prioridad? Cread un proceso con la máxima prioridad posible y uno con la mínima posible utilizando este comando. ¿Como podemos consultar la prioridad de este proceso?

La prioridad por defecto es 0.

He creado un proceso con **nice -n -20 top**, pero al haberme asignado una prioridad de 0 en vez de -20 he hecho un **renice** para cambiar la prioridad de 0 a -20.

```
alumno@alumno-VirtualBox:~$ sudo renice -20 3786
3786 (process ID) prioridad anterior 0, nueva prioridad -20
```

```
alumno@alumno-VirtualBox:~$ ps -o pid,nice,comm -p 3786
  PID  NI  COMMAND
 3786 -20 top
```

He creado un nuevo proceso con la prioridad 19, en este caso me ha asignado la prioridad que le he puesto.

```
alumno@alumno-VirtualBox:~$ nice -n 19 top
```

```
  PID  USUARIO  PR  NI   VIRT   RES   SHR  S  %CPU  %MEM   HORA+  ORDEN
1616 root      20   0 397988 117616 63904 S   1,0   2,9   2:23.85 Xorg
3112 root      20   0      0      0      0 I   0,7   0,0   0:03.47 kworker+
 635 systemd+  20   0  16004   7216   6336 S   0,3   0,2   0:21.41 systemd+
 991 root      20   0 370684   3164   2712 S   0,3   0,1   0:04.45 VBoxSer+
1771 alumno    20   0 162060   2580   2192 S   0,3   0,1   1:26.97 VBoxCli+
1849 alumno    20   0 3873280 382576 144480 S   0,3   9,5   1:30.62 gnome-s+
3716 alumno    39  19  21764   4208   3364 R   0,3   0,1   0:00.29 top
    1 root      20   0 169344  13836  8672 S   0,0   0,3   0:02.42 systemd
```

18 Cread un proceso con una prioridad muy baja y modificadla utilizando el comando **renice**. ¿Como haríais para modificar la prioridad de todos vuestros procesos? ¿Qué opción del comando **top** os permite llevar a cabo esta misma acción?

Creo un proceso con prioridad 17.

```
alumno@alumno-VirtualBox:~$ nice -n 17 top
```


PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
1771	alumne	20	0	162060	2580	2192	S	0,7	0,1	1:33.22	VBoxCli+
1849	alumne	20	0	3873280	382528	144480	S	0,7	9,5	1:36.75	gnome-s+
9	root	0	-20	0	0	0	I	0,3	0,0	0:03.11	kworker+
1616	root	20	0	397988	117616	63904	S	0,3	2,9	2:34.29	Xorg
1622	alumne	20	0	323540	8188	7168	S	0,3	0,2	0:02.66	gvfs-af+
2355	alumne	20	0	635312	57600	42284	S	0,3	1,4	0:24.55	gnome-t+
3112	root	20	0	0	0	0	I	0,3	0,0	0:04.40	kworker+
3803	alumne	37	17	21756	4200	3368	R	0,3	0,1	0:00.01	top
1	root	20	0	169344	13836	8672	S	0,0	0,3	0:02.43	systemd

Utilizo *renice -n +10 PID* para modificar la prioridad de 17 a 10.

```
alumne@alumne-VirtualBox:~$ sudo renice -n +10 3803
3803 (process ID) prioridad anterior 17, nueva prioridad 10
```

Para entrar en el modo interactivo introducimos el comando *top*.

Dentro del comando *top* solo está permitido cambiar la prioridad de un único proceso a la vez pulsando *r* e introduciendo el PID.

```
alumne@alumne-VirtualBox:~$ top

top - 00:54:46 up 4:26, 5 users, load average: 0,00, 0,05, 0,03
Tareas: 185 total, 1 ejecutar, 182 hibernar, 2 detener, 0 zombie
%Cpu(s): 1,0 us, 4,4 sy, 0,0 ni, 94,6 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 3923,5 total, 2165,8 libre, 723,2 usado, 1034,5 búfer/caché
MiB Intercambio: 4867,0 total, 4867,0 libre, 0,0 usado, 2974,7 dispon
PID para reiniciar [pid por defecto = 4367] 5269
```

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
635	systemd+	20	0	16176	7212	6336	S	0,3	0,2	0:24.61	systemd+
4532	alumne	20	0	162060	2516	2124	S	0,3	0,1	0:01.92	VBoxCli+
5269	alumne	20	0	21760	4200	3368	R	0,3	0,1	0:00.01	top
1	root	20	0	169344	13836	8672	S	0,0	0,3	0:03.01	systemd

Renice PID 5269 al valor 1

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
635	systemd+	20	0	16176	7212	6336	S	0,3	0,2	0:24.61	systemd+
4532	alumne	20	0	162060	2516	2124	S	0,3	0,1	0:01.92	VBoxCli+
5269	alumne	20	0	21760	4200	3368	R	0,3	0,1	0:00.01	top
1	root	20	0	169344	13836	8672	S	0,0	0,3	0:03.01	systemd

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
4367	root	20	0	371196	94880	61060	S	3,3	2,4	0:06.25	Xorg
4612	alumne	20	0	3770284	344516	141728	S	1,0	8,6	0:06.26	gnome-s+
5097	alumne	20	0	563000	53596	41196	S	1,0	1,3	0:01.19	gnome-t+
4532	alumne	20	0	162060	2516	2124	S	0,7	0,1	0:02.35	VBoxCli+
4823	alumne	20	0	678340	30056	21872	S	0,3	0,7	0:00.13	gsd-med+
5269	alumne	21	1	21760	4200	3368	R	0,3	0,1	0:00.02	top

19	¿Cuál es la función de los comandos <i>pidof</i> , <i>pgrep</i> y <i>pkill</i> ?
----	--

El comando *pidof* se usa para obtener el PID de un proceso en ejecución en el sistema.

```
alumne@alumne-VirtualBox:~$ pidof top
5279
```

El comando *pgrep* sirve para buscar procesos que coincidan con un patrón determinado y mostrar su número de proceso.

```
alumne@alumne-VirtualBox:~$ pgrep top
4664
4737
4741
4745
4999
5279
```

El comando *pgrep* se usa para matar procesos en un sistema Linux.

Ejecutamos Firefox por ejemplo manualmente y si introducimos en la terminal el comando *top* veremos que Firefox ya se está ejecutando.

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
4367	root	20	0	384580	106664	69584	S	20,9	2,7	0:31.83	Xorg
8376	alumne	20	0	2878732	278684	147124	S	9,6	6,9	0:05.29	firefox

El comando *kill* *firefox* lo elimina.

```
alumne@alumne-VirtualBox:~$ kill firefox
```