

10

Processes.

Modern operating systems are usually multitasking, meaning they create the illusion of doing more than one thing at once by rapidly switching from one executing program to another. The kernel manages this through the use of processes. Processes are how Linux organizes the different programs waiting for their turn at the CPU.

Los sistemas operativos modernos suelen ser multitarea, lo que significa que crean la ilusión de hacer más de una cosa a la vez al cambiar rápidamente de un programa en ejecución a otro. El kernel gestiona esto mediante el uso de procesos. Los procesos son la forma en que Linux organiza los diferentes programas que esperan su turno en la CPU.

Sometimes a computer will become sluggish or an application will stop responding. In this chapter, we will look at some of the tools available at the command line that let us examine what programs are doing and how to terminate processes that are misbehaving.

A veces, una computadora se vuelve lenta o una aplicación deja de responder. En este capítulo, veremos algunas de las herramientas disponibles en la línea de comandos que nos permiten examinar qué están haciendo los programas y cómo terminar los procesos que se están comportando mal.

This chapter will introduce the following commands:

Este capítulo presentará los siguientes comandos:

- `ps` -- Report a snapshot of current processes
`ps` -- informe de una instantánea de los procesos actuales
- `top` -- Display tasks
`top` - Mostrar tareas
- `jobs` -- List active jobs
`jobs` -- muestra los trabajos activos
- `bg` -- Place a job in the background
`bg` -- coloque un trabajo en segundo plano
- `fg` -- Place a job in the foreground
`fg` -- coloca un trabajo en primer plano
- `kill` -- Send a signal to a process
`kill` -- envía una señal a un proceso
- `killall` -- Kill processes by name
`killall` - Mata procesos por nombre

- shutdown -- Shut down or reboot the system
shutdown -- apaga o reinicia el sistema

How a Process Works

Cómo funciona un proceso

When a system starts up, the kernel initiates a few of its own activities as processes and launches a program called `init`. `init`, in turn, runs a series of shell scripts (located in `/etc`) called `init` scripts, which start all the system services. Many of these services are implemented as daemon programs, programs that just sit in the background and do their thing without having any user interface. So, even if we are not logged in, the system is at least a little busy performing routine stuff.

Cuando se inicia un sistema, el kernel inicia algunas de sus propias actividades como procesos y lanza un programa llamado `init`. `init`, a su vez, ejecuta una serie de scripts de shell (ubicados en `/etc`) llamados scripts de inicio, que inician todos los servicios del sistema. Muchos de estos servicios se implementan como programas demonio, programas que simplemente se sientan en segundo plano y hacen lo suyo sin tener ninguna interfaz de usuario. Entonces, incluso si no estamos conectados, el sistema está al menos un poco ocupado realizando tareas de rutina.

The fact that a program can launch other programs is expressed in the process scheme as a parent process producing a child process.

El hecho de que un programa pueda lanzar otros programas se expresa en el esquema del proceso como un proceso padre que produce un proceso hijo.

The kernel maintains information about each process to help keep things organized. For example, each process is assigned a number called a process ID (PID). PIDs are assigned in ascending order, with `init` always getting PID 1. The kernel also keeps track of the memory assigned to each process, as well as the processes' readiness to resume execution. Like files, processes also have owners and user IDs, effective user IDs, and so on.

El kernel mantiene información sobre cada proceso para ayudar a mantener las cosas organizadas. Por ejemplo, a cada proceso se le asigna un número llamado ID de proceso (PID). Los PID se asignan en orden ascendente, con `init` siempre obteniendo PID 1. El kernel también realiza un seguimiento de la memoria asignada a cada proceso, así como de la preparación de los procesos para reanudar la ejecución. Al igual que los archivos, los procesos también tienen propietarios e ID de usuario, ID de usuario efectivos, etc.

Viewing Processes

Procesos de visualización

The most commonly used command to view processes (there are several) is `ps`. The `ps` program has a lot of options, but in its simplest form it is used like this:

El comando más utilizado para ver procesos (hay varios) es `ps`. El programa `ps` tiene muchas opciones, pero en su forma más simple se usa así:

```
[me@linuxbox ~]$ ps
PID TTY      TIME CMD
```

```
5198 pts/1 00:00:00 bash
10129 pts/1 00:00:00 ps
```

The result in this example lists two processes, process 5198 and process 10129, which are bash and ps , respectively. As we can see, by default, ps doesn't show us very much, just the processes associated with the current terminal session. To see more, we need to add some options; however, before we do that, let's look at the other fields produced by ps . TTY is short for "teletype" and refers to the controlling terminal for the process. Unix is showing its age here.

El resultado de este ejemplo enumera dos procesos, el proceso 5198 y el proceso 10129, que son bash y ps, respectivamente. Como podemos ver, por defecto, ps no nos muestra mucho, solo los procesos asociados con la sesión de terminal actual. Para ver más, necesitamos agregar algunas opciones; sin embargo, antes de hacer eso, veamos los otros campos producidos por ps. TTY es la abreviatura de "teletipo" y se refiere a la terminal de control del proceso. Unix está mostrando su edad aquí.

The TIME field is the amount of CPU time consumed by the process. As we can see, neither process makes the computer work very hard.

El campo TIME es la cantidad de tiempo de CPU consumido por el proceso. Como podemos ver, ninguno de los procesos hace que la computadora trabaje mucho.

If we add an option, we can get a bigger picture of what the system is doing.

Si agregamos una opción, podemos obtener una imagen más amplia de lo que está haciendo el sistema.

```
[me@linuxbox ~]$ ps x
```

Adding the x option (note that there is no leading dash) tells ps to show all of our processes regardless of what terminal (if any) they are controlled by. The presence of a ? in the TTY column indicates no controlling terminal.

Agregar la opción x (tenga en cuenta que no hay un guión inicial) le dice a ps que muestre todos nuestros procesos independientemente de la terminal (si corresponde) que los controle. La presencia de un ? en la columna TTY indica que no hay terminal de control.

Using this option, we see a list of every process that we own.

Con esta opción, vemos una lista de todos los procesos que poseemos.

Because the system is running a lot of processes, ps produces a long list. It is often helpful to pipe the output from ps into less for easier viewing.

Debido a que el sistema ejecuta muchos procesos, ps produce una lista larga. A menudo es útil canalizar la salida de ps a menos para facilitar la visualización.

Some option combinations also produce long lines of output, so maximizing the terminal emulator window might be a good idea, too.

Algunas combinaciones de opciones también producen largas líneas de salida, por lo que maximizar la ventana del emulador de terminal también puede ser una buena idea.

A new column titled STAT has been added to the output. STAT is short for state and reveals the current status of the process, as shown in Table 10-1.

Se ha agregado una nueva columna titulada STAT a la salida. STAT es la abreviatura de estado y revela el estado actual del proceso, como se muestra en la Tabla 10-1.

Table 10-1: Process States

Tabla 10-1: Estados del proceso

State	Meaning
R	Running. This means the process is running or ready to run. <i>Corriendo. Esto significa que el proceso se está ejecutando o está listo para ejecutarse.</i>
S	Sleeping. The process is not running; rather, it is waiting for an event, such as a keystroke or network packet. <i>Durmiendo. El proceso no se está ejecutando; más bien, está esperando un evento, como una pulsación de tecla o un paquete de red.</i>
D	Uninterruptible sleep. The process is waiting for I/O such as a disk drive. <i>Sueño ininterrumpido. El proceso está esperando E / S, como una unidad de disco.</i>
T	Stopped. The process has been instructed to stop. You'll learn more about this later in the chapter. <i>Interrumpido. Se ha ordenado que el proceso se detenga. Aprenderá más sobre esto más adelante en el capítulo.</i>
Z	A defunct or "zombie" process. This is a child process that has terminated but has not been cleaned up by its parent. <i>Un proceso difunto o "zombi". Este es un proceso hijo que ha terminado, pero su padre no lo ha limpiado.</i>
<	A high-priority process. It's possible to grant more importance to a process, giving it more time on the CPU. This property of a process is called niceness. A process with high priority is said to be less nice because it's taking more of the CPU's time, which leaves less for everybody else. <i>Un proceso de alta prioridad. Es posible otorgar más importancia a un proceso, dándole más tiempo en la CPU. Esta propiedad de un proceso se llama bondad. Se dice que un proceso con alta prioridad es menos agradable porque consume más tiempo de la CPU, lo que deja menos para todos los demás.</i>
N	A low-priority process. A process with low priority (a nice process) will get processor time only after other processes with higher priority have been serviced. <i>Un proceso de baja prioridad. Un proceso con baja prioridad (un buen proceso) obtendrá tiempo de procesador solo después de que se hayan atendido otros procesos con mayor prioridad.</i>

The process state may be followed by other characters. These indicate various exotic process characteristics. See the `ps` man page for more details.

El estado del proceso puede ir seguido de otros caracteres. Estos indican varias características exóticas del proceso. Consulte la página de manual de `ps` para obtener más detalles.

Another popular set of options is `aux` (without a leading dash). This gives us even more information.
Otro conjunto popular de opciones es `aux` (sin un guión inicial). Esto nos da aún más información.

```
[me@linuxbox ~]$ ps aux
```

This set of options displays the processes belonging to every user. Using the options without the leading dash invokes the command with “BSD-style” behavior. The Linux version of `ps` can emulate the behavior of the `ps` program found in several different Unix implementations. With these options, we get the additional columns shown in Table 10-2.

Este conjunto de opciones muestra los procesos que pertenecen a cada usuario. El uso de las opciones sin el guión inicial invoca el comando con un comportamiento de "estilo BSD". La versión de Linux de `ps` puede emular el comportamiento del programa `ps` que se encuentra en varias implementaciones de Unix diferentes. Con estas opciones, obtenemos las columnas adicionales que se muestran en la Tabla 10-2.

Table 10-2: BSD Style `ps` Column Headers

Tabla 10-2: Encabezados de columna `ps` estilo BSD

Header	Meaning
USER	User ID. This is the owner of the process. ID de usuario. Este es el dueño del proceso.
%CPU	CPU usage in percent. Uso de CPU en porcentaje.
%MEM	Memory usage in percent. Uso de memoria en porcentaje.
VSZ	Virtual memory size. Tamaño de la memoria virtual.
RSS	Resident set size. This is the amount of physical memory (RAM) the process is using in kilobytes. Tamaño del conjunto residente. Esta es la cantidad de memoria física (RAM) que el proceso está usando en kilobytes.
START	Time when the process started. For values over 24 hours, a date is used. Hora en que comenzó el proceso. Para valores superiores a 24 horas, se utiliza una fecha.

Viewing Processes Dynamically with `top`

Visualización de procesos dinámicamente con `top`

While the `ps` command can reveal a lot about what the machine is doing, it provides only a snapshot of the machine’s state at the moment the `ps` command is executed. To see a more dynamic view of the machine’s activity, we use the `top` command.

Si bien el comando `ps` puede revelar mucho sobre lo que está haciendo la máquina, solo proporciona una instantánea del estado de la máquina en el momento en que se ejecuta el comando `ps`. Para ver una vista más dinámica de la actividad de la máquina, usamos el comando superior.

```
[me@linuxbox ~]$ top
```

The `top` program displays a continuously updating (by default, every three seconds) display of the system processes listed in order of process activity. The name `top` comes from the fact that the `top` program is used to see the “top” processes on the system. The `top` display consists of two parts: a system summary at

the top of the display, followed by a table of processes sorted by CPU activity.

El programa superior muestra una pantalla de actualización continua (por defecto, cada tres segundos) de los procesos del sistema enumerados en orden de actividad del proceso. El nombre top proviene del hecho de que el programa top se usa para ver los procesos “top” en el sistema. La pantalla superior consta de dos partes: un resumen del sistema en la parte superior de la pantalla, seguido de una tabla de procesos ordenados por actividad de la CPU.

The system summary contains a lot of good stuff. Table 10-3 provides a rundown.

El resumen del sistema contiene muchas cosas buenas. La Tabla 10-3 proporciona un resumen.

Table 10-3: top Information Fields

Tabla 10-3: Campos de información superiores

Row	Field	Meaning
1	top	This is the name of the program. Este es el nombre del programa.
	14:59:20	This is the current time of day. Esta es la hora actual del día.
	up 6:30	This is called uptime. It is the amount of time since the machine was last booted. In this example, the system has been up for six and a half hours. A esto se le llama tiempo de actividad. Es la cantidad de tiempo desde que se inició la máquina por última vez. En este ejemplo, el sistema ha estado funcionando durante seis horas y media.
	2 users	There are two users logged in. Hay dos usuarios conectados.
	load average:	Load average refers to the number of processes that are waiting to run; that is, the number of processes that are in a runnable state and are sharing the CPU. Three values are shown, each for a different period of time. The first is the average for the last 60 seconds, the next the previous 5 minutes, and finally the previous 15 minutes. Values less than 1.0 indicate that the machine is not busy. El promedio de carga se refiere al número de procesos que están esperando para ejecutarse; es decir, el número de procesos que están en un estado ejecutable y comparten la CPU. Se muestran tres valores, cada uno para un período de tiempo diferente. El primero es el promedio de los últimos 60 segundos, el siguiente los 5 minutos anteriores y finalmente los 15 minutos anteriores. Los valores inferiores a 1.0 indican que la máquina no está ocupada.
2	Tasks:	This summarizes the number of processes and their various process states. Esto resume el número de procesos y sus diversos estados de proceso.
3	Cpu(s):	This row describes the character of the activities that the CPU is performing. Esta fila describe el carácter de las actividades que realiza la CPU.

Row	Field	Meaning
	0.7%us	0.7 percent of the CPU is being used for user processes. This means processes outside the kernel. El 0,7 por ciento de la CPU se utiliza para procesos de usuario. Esto significa procesos fuera del kernel.
	1.0%sy	1.0 percent of the CPU is being used for system (kernel) processes. El 1.0 por ciento de la CPU se usa para procesos del sistema (kernel).
	0.0%ni	0.0 percent of the CPU is being used by "nice" (low-priority) processes. El 0,0 por ciento de la CPU está siendo utilizado por procesos "agradables" (de baja prioridad).
	98.3%id	98.3 percent of the CPU is idle. El 98.3 por ciento de la CPU está inactivo.
	0.0%wa	0.0 percent of the CPU is waiting for I/O. El 0,0 por ciento de la CPU está esperando E / S.
4	Mem	This shows how physical RAM is being used. Esto muestra cómo se utiliza la RAM física.
5	Swap	This shows how swap space (virtual memory) is being used. Esto muestra cómo se utiliza el espacio de intercambio (memoria virtual).

The top program accepts a number of keyboard commands. The two most interesting are `h`, which displays the program's help screen, and `q`, which quits top.

El programa superior acepta varios comandos de teclado. Los dos más interesantes son `h`, que muestra la pantalla de ayuda del programa, `yq`, que sale de la parte superior.

Both major desktop environments provide graphical applications that display information similar to top (in much the same way that Task Manager in Windows works), but top is better than the graphical versions because it is faster and it consumes far fewer system resources. After all, our system monitor program shouldn't be the source of the system slowdown that we are trying to track.

Los dos entornos de escritorio principales proporcionan aplicaciones gráficas que muestran información similar a top (de la misma manera que funciona el Administrador de tareas en Windows), pero top es mejor que las versiones gráficas porque es más rápido y consume muchos menos recursos del sistema. Después de todo, nuestro programa de monitorización del sistema no debería ser la fuente de la ralentización del sistema que estamos intentando rastrear.

Controlling Processes

Procesos de control

Now that we can see and monitor processes, let's gain some control over them. For our experiments, we're going to use a little program called `xlogo` as our guinea pig. The `xlogo` program is a sample program supplied with the X Window System (the underlying engine that makes the graphics on our display go), which simply displays a resizable window containing the X logo. First, we'll get to know our test subject. Ahora que podemos ver y monitorear los procesos, obtengamos cierto control sobre ellos. Para nuestros

experimentos, usaremos un pequeño programa llamado xlogo como nuestro conejillo de indias. El programa xlogo es un programa de muestra suministrado con X Window System (el motor subyacente que hace que los gráficos en nuestra pantalla funcionen), que simplemente muestra una ventana redimensionable que contiene el logo X. Primero, conoceremos a nuestro sujeto de prueba.

```
[me@linuxbox ~]$ xlogo
```

After entering the command, a small window containing the logo should appear somewhere on the screen. On some systems, xlogo might print a warning message, but it can be safely ignored.

Después de ingresar el comando, debería aparecer una pequeña ventana que contiene el logotipo en algún lugar de la pantalla. En algunos sistemas, xlogo puede imprimir un mensaje de advertencia, pero se puede ignorar con seguridad.

NOTE

Nota

If your system does not include the xlogo program, try using gedit or kwrite instead.

Si su sistema no incluye el programa xlogo, intente usar gedit o kwrite en su lugar.

We can verify that xlogo is running by resizing its window. If the logo is redrawn in the new size, the program is running.

Podemos verificar que xlogo se está ejecutando cambiando el tamaño de su ventana. Si el logotipo se vuelve a dibujar en el nuevo tamaño, el programa se está ejecutando.

Notice how our shell prompt has not returned? This is because the shell is waiting for the program to finish, just like all the other programs we have used so far. If we close the xlogo window (see Figure 10-1), the prompt returns.

¿Observa cómo nuestro indicador de shell no ha regresado? Esto se debe a que el shell está esperando a que finalice el programa, al igual que todos los demás programas que hemos utilizado hasta ahora. Si cerramos la ventana de xlogo (vea la Figura 10-1), el indicador regresa.



Interrupting a Process

Interrupción de un proceso

Let's observe what happens when we run xlogo again. First, enter the xlogo command and verify that the program is running. Next, return to the terminal window and press ctrl-C.

Observemos lo que sucede cuando ejecutamos xlogo nuevamente. Primero, ingrese el comando xlogo y verifique que el programa se esté ejecutando. Luego, regrese a la ventana de terminal y presione ctrl-C.

```
[me@linuxbox ~]$ xlogo  
[me@linuxbox ~]$
```

In a terminal, pressing ctrl -C interrupt a program. This means we are politely asking the program to terminate. After we pressed ctrl-C, the xlogo window closed, and the shell prompt returned.

En una terminal, presionar ctrl -C interrumpe un programa. Esto significa que estamos pidiendo amablemente al programa que termine. Después de presionar ctrl-C, la ventana de xlogo se cerró y el indicador de shell regresó.

Many (but not all) command line programs can be interrupted by using this technique.

Muchos (pero no todos) los programas de línea de comandos se pueden interrumpir utilizando esta técnica.

Putting a Process in the Background

Poner un proceso en segundo plano

Let's say we wanted to get the shell prompt back without terminating the xlogo program. We can do this by placing the program in the background. Think of the terminal as having a foreground (with stuff visible on the surface like the shell prompt) and a background (with stuff hidden behind the surface). To launch a program so that it is immediately placed in the background, we follow the command with an ampersand (&) character.

Digamos que queremos recuperar el indicador de shell sin terminar el programa xlogo. Podemos hacer esto colocando el programa en segundo plano. Piense en la terminal como si tuviera un primer plano (con cosas visibles en la superficie como el indicador de shell) y un fondo (con cosas ocultas detrás de la superficie). Para lanzar un programa para que se coloque inmediatamente en segundo plano, seguimos el comando con un carácter comercial (&).

```
[me@linuxbox ~]$ xlogo &  
[1] 28236  
[me@linuxbox ~]$
```

After entering the command, the xlogo window appeared, and the shell prompt returned, but some funny numbers were printed too. This message is part of a shell feature called job control. With this message, the shell is telling us that we have started job number 1 ([1]) and that it has PID 28236. If we run ps , we can see our process.

Después de ingresar el comando, apareció la ventana xlogo y el indicador de shell regresó, pero también se imprimieron algunos números divertidos. Este mensaje es parte de una función de shell llamada control de trabajo. Con este mensaje, el shell nos dice que hemos iniciado el trabajo número 1 ([1]) y que tiene el PID 28236. Si ejecutamos ps, podemos ver nuestro proceso.

```
[me@EliteDesk:~]$ ps
  PID TTY          TIME CMD
 29015 pts/3        00:00:00 bash
 29024 pts/3        00:00:00 xlogo
 29026 pts/3        00:00:00 ps
```

The shell's job control facility also gives us a way to list the jobs that have been launched from our terminal. Using the jobs command, we can see this list:

La función de control de trabajos del shell también nos brinda una forma de enumerar los trabajos que se han lanzado desde nuestra terminal. Usando el comando de trabajos, podemos ver esta lista:

```
hernani@EliteDesk:~$ jobs
[1]+  Ejecutando          xlogo &
```

The results show that we have one job, numbered 1; that it is running; and that the command was xlogo &. Los resultados muestran que tenemos un trabajo, el número 1; que está funcionando; y que el comando era xlogo &.

Returning a Process to the Foreground

Devolver un proceso al primer plano

A process in the background is immune from terminal keyboard input, including any attempt to interrupt it with ctrl-C. To return a process to the foreground, use the fg command in this way:

Un proceso en segundo plano es inmune a la entrada del teclado del terminal, incluido cualquier intento de interrumpirlo con ctrl-C. Para devolver un proceso al primer plano, use el comando fg de esta manera:

```
me@EliteDesk:~$ jobs
[1]+  Ejecutando          xlogo &
me@EliteDesk:~$ fg %1
xlogo
```

Stopping (Pausing) a Process

Detener (pausar) un proceso

Sometimes we'll want to stop a process without terminating it. This is often done to allow a foreground process to be moved to the background. To stop a foreground process and place it in the background, press ctrl-Z. Let's try it. At the command prompt, type xlogo , press enter , and then press ctrl-Z.

A veces, queremos detener un proceso sin terminarlo. Esto se hace a menudo para permitir que un proceso en primer plano se mueva al fondo. Para detener un proceso en primer plano y colocarlo en segundo plano, presione ctrl-Z. Vamos a intentarlo. En el símbolo del sistema, escriba xlogo, presione enter y luego presione ctrl-Z.

```
me@EliteDesk:~$ fg %1
xlogo
^Z
[1]+  Detenido                  xlogo
me@EliteDesk:~$
```

After stopping xlogo , we can verify that the program has stopped by attempting to resize the xlogo window. We will see that it appears quite dead. We can either continue the program's execution in the foreground, using the fg command, or resume the program's execution in the background with the bg command.

Después de detener xlogo, podemos verificar que el programa se ha detenido intentando cambiar el tamaño de la ventana de xlogo. Veremos que parece bastante muerto. Podemos continuar la ejecución del programa en primer plano, usando el comando fg, o reanudar la ejecución del programa en segundo plano con el comando bg.

```
hernani@EliteDesk:~$ bg %1
[1]+ xlogo &
hernani@EliteDesk:~$
```

As with the fg command, the jobspec is optional if there is only one job. Moving a process from the foreground to the background is handy if we launch a graphical program from the command line but forget to place it in the background by appending the trailing &.

Al igual que con el comando fg, la especificación de trabajo es opcional si solo hay un trabajo. Mover un proceso del primer plano al fondo es útil si lanzamos un programa gráfico desde la línea de comandos pero nos olvidamos de colocarlo en segundo plano agregando el símbolo & final.

Why would we want to launch a graphical program from the command line? There are two reasons.

¿Por qué querríamos lanzar un programa gráfico desde la línea de comandos? Hay dos razones

- The program you want to run might not be listed on the window manager's menus (such as xlogo). Es posible que el programa que desea ejecutar no aparezca en los menús del administrador de ventanas (como xlogo).
- By launching a program from the command line, you might be able to see error messages that would otherwise be invisible if the program were launched graphically. Sometimes, a program will fail to start up when launched from the graphical menu. By launching it from the command line instead, we may see an error message that will reveal the problem. Also, some graphical programs have interesting and useful command line options.

Al iniciar un programa desde la línea de comandos, es posible que pueda ver mensajes de error que de otro modo serían invisibles si el programa se iniciara gráficamente. A veces, un programa no puede iniciarse cuando se inicia desde el menú gráfico. Al iniciarlo desde la línea de comandos, es posible que veamos un mensaje de error que revelará el problema. Además, algunos programas gráficos tienen opciones de línea de comandos interesantes y útiles.

Signals

Señales

The kill command is used to “kill” processes. This allows us to terminate programs that need killing (that is, some kind of pausing or termination).

El comando kill se usa para "matar" procesos. Esto nos permite terminar programas que necesitan ser eliminados (es decir, algún tipo de pausa o finalización).

Here's an example:

Aquí tienes un ejemplo:

```
[me@EliteDesk:~]$ xlogo &
[2] 29651
[me@EliteDesk:~]$ kill 29651
[me@EliteDesk:~]$
[2]+ Terminado xlogo
[me@EliteDesk:~]$
```

We first launch xlogo in the background. The shell prints the jobspec and the PID of the background process. Next, we use the kill command and specify the PID of the process we want to terminate. We could have also specified the process using a jobspec (for example, %1) instead of a PID.

Primero lanzamos xlogo en segundo plano. El shell imprime la especificación de trabajo y el PID del proceso en segundo plano. A continuación, usamos el comando kill y especificamos el PID del proceso que queremos terminar. También podríamos haber especificado el proceso usando una especificación de trabajo (por ejemplo, % 1) en lugar de un PID.

While this is all very straightforward, there is more to it than that. The kill command doesn't exactly “kill” processes; rather, it sends them signals. Signals are one of several ways that the operating system communicates with programs. We have already seen signals in action with the use of ctrl-C and ctrl-Z. When the terminal receives one of these keystrokes, it sends a signal to the program in the foreground. In the case of ctrl-C, a signal called INT (interrupt) is sent; with ctrl-Z, a signal called TSTP (terminal stop) is sent. Programs, in turn, “listen” for signals and may act upon them as they are received. The fact that a program can listen and act upon signals allows a program to do things such as save work in progress when it is sent a termination signal.

Si bien todo esto es muy sencillo, hay más que eso. El comando kill no "mata" exactamente los procesos; más bien, les envía señales. Las señales son una de las diversas formas en que el sistema operativo se comunica con los programas. Ya hemos visto señales en acción con el uso de ctrl-C y ctrl-Z. Cuando el terminal recibe una de estas pulsaciones de tecla, envía una señal al programa en primer plano. En el caso de ctrl-C, se envía una señal llamada INT (interrupción); con ctrl-Z, se envía una señal llamada TSTP (parada de terminal). Los programas, a su vez, "escuchan" las señales y pueden actuar sobre ellas a medida que las reciben. El hecho de que un programa pueda escuchar y actuar sobre las señales permite que un programa haga cosas como guardar el trabajo en curso cuando se le envía una señal de terminación.

Sending Signals to Processes with kill

Envío de señales a procesos con kill

The kill command is used to send signals to programs. Its most common syntax looks like this:

El comando kill se usa para enviar señales a programas. Su sintaxis más común se ve así:

```
kill -signal PID...
```

If no signal is specified on the command line, then the TERM (terminate) signal is sent by default. The kill command is most often used to send the signals listed in Table 10-4.

Si no se especifica ninguna señal en la línea de comando, la señal TERM (terminar) se envía por defecto. El comando kill se usa con mayor frecuencia para enviar las señales enumeradas en la Tabla 10-4.

Table 10-4: Common Signals

Tabla 10-4: Señales comunes

Number	Name	Meaning
1	HUP	<p>Hang up. This is a vestige of the good old days when terminals were attached to remote computers with phone lines and modems. The signal is used to indicate to programs that the controlling terminal has “hung up.” The effect of this signal can be demonstrated by closing a terminal session. The foreground program running on the terminal will be sent the signal and will terminate.</p> <p>Hang up. This is a vestige of the good old days when terminals were attached to remote computers with phone lines and modems. The signal is used to indicate to programs that the controlling terminal has “hung up.” The effect of this signal can be demonstrated by closing a terminal session. The foreground program running on the terminal will be sent the signal and will terminate.</p> <p>This signal is also used by many daemon programs to cause a reinitialization. This means that when a daemon is sent this signal, it will restart and reread its configuration file. The Apache web server is an example of a daemon that uses the HUP signal in this way.</p> <p>Esta señal también es utilizada por muchos programas demonio para provocar una reinicialización. Esto significa que cuando un demonio recibe esta señal, se reiniciará y volverá a leer su archivo de configuración. El servidor web Apache es un ejemplo de un demonio que usa la señal HUP de esta manera.</p>
2	INT	<p>Interrupt. This performs the same function as ctrl-C sent from the terminal. It will usually terminate a program.</p> <p>Interrumpir. Esto realiza la misma función que ctrl-C enviado desde el terminal. Por lo general, terminará un programa.</p>

Number	Name	Meaning
9	KILL	<p>Kill. This signal is special. Whereas programs may choose to handle signals sent to them in different ways, including ignoring them all together, the KILL signal is never actually sent to the target program. Rather, the kernel immediately terminates the process. When a process is terminated in this manner, it is given no opportunity to "clean up" after itself or save its work. For this reason, the KILL signal should be used only as a last resort when other termination signals fail.</p> <p>Matar. Esta señal es especial. Mientras que los programas pueden optar por manejar las señales que se les envían de diferentes maneras, incluso ignorarlas todas juntas, la señal KILL nunca se envía realmente al programa de destino. Más bien, el kernel termina inmediatamente el proceso. Cuando un proceso se termina de esta manera, no se le da la oportunidad de "limpiar" después de sí mismo o guardar su trabajo. Por esta razón, la señal KILL debe usarse solo como último recurso cuando fallan otras señales de terminación.</p>
15	TERM	<p>Terminate. This is the default signal sent by the kill command. If a program is still "alive" enough to receive signals, it will terminate.</p> <p>Terminar. Esta es la señal predeterminada enviada por el comando kill. Si un programa todavía está lo suficientemente "vivo" para recibir señales, terminará.</p>
18	CONT	<p>Continue. This will restore a process after a STOP or TSTP signal. This signal is sent by the bg and fg commands.</p> <p>Continuar. Esto restaurará un proceso después de una señal de PARADA o TSTP. Esta señal es enviada por los comandos bg y fg.</p>
19	STOP	<p>Stop. This signal causes a process to pause without terminating. Like the KILL signal, it is not sent to the target process, and thus it cannot be ignored.</p> <p>Detener. Esta señal hace que un proceso se detenga sin terminar. Como la señal KILL, no se envía al proceso objetivo y, por lo tanto, no se puede ignorar.</p>
20	TSTP	<p>Terminal stop. This is the signal sent by the terminal when ctrl-Z is pressed. Unlike the STOP signal, the TSTP signal is received by the program, but the program may choose to ignore it.</p> <p>Parada terminal. Esta es la señal enviada por el terminal cuando se presiona ctrl-Z. A diferencia de la señal STOP, el programa recibe la señal TSTP, pero el programa puede optar por ignorarla.</p>

Let's try the kill command.

Probemos con el comando de matar.

```
[me@EliteDesk:~]$ xlogo &
[1] 30035
[me@EliteDesk:~]$ ps
  PID TTY          TIME CMD
 29015 pts/3        00:00:00 bash
  30035 pts/3        00:00:00 xlogo
  30039 pts/3        00:00:00 ps
[me@EliteDesk:~]$ kill -1 30035
```

```
[me@EliteDesk:~]$
[1]+  Colgar (hangup)          xlogo
[me@EliteDesk:~]$
```

In this example, we start the xlogo program in the background and then send it a HUP signal with kill . The xlogo program terminates, and the shell indicates that the background process has received a hang-up signal. We may need to press enter a couple of times before the message appears. Note that signals may be specified either by number or by name, including the name prefixed with the letters SIG.

En este ejemplo, iniciamos el programa xlogo en segundo plano y luego le enviamos una señal HUP con kill. El programa xlogo termina y el shell indica que el proceso en segundo plano ha recibido una señal de colgar. Es posible que debamos presionar Intro un par de veces antes de que aparezca el mensaje. Tenga en cuenta que las señales se pueden especificar por número o por nombre, incluido el nombre con el prefijo de las letras SIG.

```
[me@EliteDesk:~]$ xlogo &
[1] 30301
[me@EliteDesk:~]$ kill -INT 30301
[me@EliteDesk:~]$
[1]+  Interrupción            xlogo
[me@EliteDesk:~]$ xlogo &
[1] 30315
[me@EliteDesk:~]$ kill -SIGINT 30315
[me@EliteDesk:~]$
[1]+  Interrupción            xlogo
[me@EliteDesk:~]$
```

Repeat the preceding example and try the other signals. Remember, we can also use jobspecs in place of PIDs.

Repita el ejemplo anterior y pruebe las otras señales. Recuerde, también podemos usar especificaciones de trabajo en lugar de PID.

Processes, like files, have owners, and you must be the owner of a process (or the superuser) to send it signals with kill.

Los procesos, como los archivos, tienen propietarios, y usted debe ser el propietario de un proceso (o el superusuario) para enviarle señales con kill.

In addition to the list of signals covered so far, which are most often used with kill , there are other signals frequently used by the system, as listed in Table 10-5.

Además de la lista de señales cubiertas hasta ahora, que se utilizan con mayor frecuencia con kill, hay otras señales que el sistema utiliza con frecuencia, como se enumera en la tabla 10-5.

Table 10-5: Other Common Signals
Tabla 10-5: Otras señales comunes

Number	Name	Meaning
3	QUIT	Quit. quitar

Number	Name	Meaning
11	SEGV	Segmentation violation. This signal is sent if a program makes illegal use of memory; that is, if it tried to write somewhere it was not allowed to write. Violación de segmentación. Esta señal se envía si un programa hace un uso ilegal de la memoria; es decir, si intentó escribir en algún lugar, no se le permitió escribir.
28	WINCH	Window change. This is the signal sent by the system when a window changes size. Some programs, such as top and less , will respond to this signal by redrawing themselves to fit the new window dimensions. Cambio de ventana. Esta es la señal que envía el sistema cuando una ventana cambia de tamaño. Algunos programas, como top y less, responderán a esta señal redibujándose para adaptarse a las nuevas dimensiones de la ventana.

For the curious, you can display a complete list of signals with the following command:

Para los curiosos, puede mostrar una lista completa de señales con el siguiente comando:

```
[me@linuxbox ~]$ kill -l
```

Sending Signals to Multiple Processes with killall

Envío de señales a múltiples procesos con killall

It's also possible to send signals to multiple processes matching a specified program or username by using the killall command. Here is the syntax:

También es posible enviar señales a varios procesos que coincidan con un programa o nombre de usuario específico mediante el comando killall. Aquí está la sintaxis:

```
killall [-u user] [-signal] name...
```

To demonstrate, we will start a couple of instances of the xlogo program and then terminate them.

Para demostrarlo, iniciaremos un par de instancias del programa xlogo y luego las terminaremos.

```
[me@EliteDesk:~]$ xlogo &
[1] 30487
[me@EliteDesk:~]$ xlogo &
[2] 30496
[me@EliteDesk:~]$ killall xlogo
[me@EliteDesk:~]$
[1]- Terminado          xlogo
[2]+ Terminado          xlogo
[me@EliteDesk:~]$
```

Remember, as with kill , you must have superuser privileges to send signals to processes that do not belong to you.

Recuerde, al igual que con kill, debe tener privilegios de superusuario para enviar señales a procesos que no le pertenecen.

Shutting Down the System

Apagar el sistema

The process of shutting down the system involves the orderly termination of all the processes on the system, as well as performing some vital house-keeping chores (such as syncing all of the mounted file systems) before the system powers off. Four commands can perform this function:

El proceso de apagado del sistema implica la terminación ordenada de todos los procesos en el sistema, así como la realización de algunas tareas domésticas vitales (como sincronizar todos los sistemas de archivos montados) antes de que el sistema se apague. Cuatro comandos pueden realizar esta función:

- halt
detener
- poweroff
apagado
- reboot
reiniciar
- shutdown
apagar

The first three are pretty self-explanatory and are generally used without any command line options. Here's an example:

Los primeros tres se explican por sí mismos y generalmente se usan sin ninguna opción de línea de comando. Aquí tienes un ejemplo:

```
[me@linuxbox ~]$ sudo reboot
```

The shutdown command is a bit more interesting. With it, we can specify which of the actions to perform (halt, power down, or reboot) and provide a time delay to the shutdown event. Most often it is used like this to halt the system:

El comando de apagado es un poco más interesante. Con él, podemos especificar cuál de las acciones realizar (detener, apagar o reiniciar) y proporcionar un retraso de tiempo para el evento de apagado. La mayoría de las veces se usa así para detener el sistema:

```
[me@linuxbox ~]$ sudo shutdown -h now
```

or like this to reboot the system:

o así para reiniciar el sistema:

```
[me@linuxbox ~]$ sudo shutdown -r now
```

The delay can be specified in a variety of ways. See the shutdown man page for details. Once the shutdown command is executed, a message is "broadcast" to all logged-in users warning them of the impending event.

El retraso se puede especificar de varias formas. Consulte la página del manual de apagado para obtener más detalles. Una vez que se ejecuta el comando de apagado, se "transmite" un mensaje a todos los usuarios registrados advirtiéndoles del evento inminente.

More Process-Related Commands

Más comandos relacionados con el proceso

Because monitoring processes is an important system administration task, there are a lot of commands for it. Table 10-6 lists some to play with.

Debido a que el monitoreo de procesos es una tarea importante de administración del sistema, existen muchos comandos para ello. La tabla 10-6 enumera algunos para jugar.

Table 10-6: Other Process-Related Commands

Tabla 10-6: Otros comandos relacionados con el proceso

Command	Description
pstree	Outputs a process list arranged in a tree-like pattern showing the parent-child relationships between processes. Genera una lista de procesos organizada en un patrón en forma de árbol que muestra las relaciones padre-hijo entre procesos.
vmstat	Outputs a snapshot of system resource usage including memory, swap, and disk I/O. To see a continuous display, follow the command with a time delay (in seconds) for updates. Here's an example: <code>vmstat 5</code> . Terminate the output with <code>ctrl-C</code> . Genera una instantánea del uso de recursos del sistema, incluida la memoria, el intercambio y la E / S de disco. Para ver una pantalla continua, siga el comando con un retraso de tiempo (en segundos) para las actualizaciones. Aquí tienes un ejemplo: <code>vmstat 5</code> . Termine la salida con <code>ctrl-C</code> .
xload	A graphical program that draws a graph showing system load over time. Un programa gráfico que dibuja un gráfico que muestra la carga del sistema a lo largo del tiempo.
tload	Similar to the xload program but draws the graph in the terminal. Terminate the output with <code>ctrl-C</code> . Similar al programa xload pero dibuja el gráfico en la terminal. Termine la salida con <code>ctrl-C</code> .

Summing Up

Resuman

Most modern systems feature a mechanism for managing multiple processes.

La mayoría de los sistemas modernos cuentan con un mecanismo para administrar múltiples procesos.

Linux provides a rich set of tools for this purpose. Given that Linux is the world's most deployed server operating system, this makes a lot of sense.

Linux proporciona un amplio conjunto de herramientas para este propósito. Dado que Linux es el sistema operativo de servidor más implementado del mundo, esto tiene mucho sentido.

Unlike some other systems, however, Linux relies primarily on command line tools for process management. Though there are graphical process tools for Linux, the command line tools are greatly preferred because of their speed and light footprint. While the GUI tools may look pretty, they often create a lot of system load themselves, which somewhat defeats the purpose.

Sin embargo, a diferencia de otros sistemas, Linux se basa principalmente en herramientas de línea de comandos para la gestión de procesos. Aunque existen herramientas de proceso gráfico para Linux, las herramientas de línea de comandos son las preferidas debido a su velocidad y tamaño reducido. Si bien las herramientas de la GUI pueden verse bonitas, a menudo crean una gran cantidad de carga del sistema, lo que en cierto modo frustra el propósito.