
Package Mangement

Gestor de paquetes

If we spend any time in the Linux community, we hear many opinions as to which of the many Linux distributions is "best." Often, these discussions get really silly, focusing on such things as the prettiness of the desktop background (some people won't use Ubuntu because of its default color scheme!) and other trivial matters.

Si pasamos algún tiempo en la comunidad de Linux, escuchamos muchas opiniones sobre cuál de las muchas distribuciones de Linux es "mejor". A menudo, estas discusiones se vuelven realmente tontas, centrándose en cosas como la belleza del fondo del escritorio (¡algunas personas no usarán Ubuntu debido a su combinación de colores predeterminada!) Y otros asuntos triviales.

The most important determinant of distribution quality is the packaging system and the vitality of the distribution's support community. As we spend more time with Linux, we will see that its software landscape is extremely dynamic. Things are constantly changing. Most of the top-tier Linux distributions release new versions every six months and many individual program updates every day. To keep up with this blizzard of software, we need good tools for package management.

El determinante más importante de la calidad de la distribución es el sistema de empaque y la vitalidad de la comunidad de apoyo de la distribución. A medida que pasemos más tiempo con Linux, veremos que su panorama de software es extremadamente dinámico. Las cosas cambian constantemente. La mayoría de las distribuciones de Linux de primer nivel lanzan nuevas versiones cada seis meses y muchas actualizaciones de programas individuales todos los días. Para seguir el ritmo de esta tormenta de software, necesitamos buenas herramientas para la gestión de paquetes.

Package management is a method of installing and maintaining software on the system. Today, most people can satisfy all of their software needs by installing packages from their Linux distributor. This contrasts with the early days of Linux, when one had to download and compile source code to install software. There isn't anything wrong with compiling source code; in fact, having access to source code is the great wonder of Linux. It gives us (and everybody else) the ability to examine and improve the system.

La gestión de paquetes es un método para instalar y mantener software en el sistema. Hoy en día, la mayoría de las personas pueden satisfacer todas sus necesidades de software instalando paquetes de su distribuidor de Linux. Esto contrasta con los primeros días de Linux, cuando era necesario descargar y compilar el código fuente para instalar el software. No hay nada de malo en compilar el código fuente; de hecho, tener acceso al código fuente es la gran maravilla de Linux. Nos da (y a todos los demás) la capacidad de examinar y mejorar el sistema.

It's just that having a precompiled package is faster and easier to deal with.

Es solo que tener un paquete precompilado es más rápido y más fácil de manejar.

In this chapter, we will look at some of the command line tools used for package management. While all the major distributions provide powerful and sophisticated graphical programs for maintaining the system, it is important to learn about the command line programs, too. They can perform many tasks that are difficult (or impossible) to do with their graphical counterparts.

En este capítulo, veremos algunas de las herramientas de línea de comandos que se utilizan para la gestión de paquetes. Si bien todas las distribuciones principales proporcionan programas gráficos potentes y sofisticados para el mantenimiento del sistema, también es importante aprender sobre los programas de línea de comandos. Pueden realizar muchas tareas que son difíciles (o imposibles) de hacer con sus contrapartes gráficas.

Packaging Systems

Sistemas de empaque

Different distributions use different packaging systems, and as a general rule, a package intended for one distribution is not compatible with another distribution. Most distributions fall into one of two camps of packaging technologies: the Debian .deb camp and the Red Hat .rpm camp. There are some important exceptions such as Gentoo, Slackware, and Arch, but most others use one of these two basic systems, as shown in Table 14-1.

Diferentes distribuciones utilizan diferentes sistemas de empaquetado y, como regla general, un paquete destinado a una distribución no es compatible con otra distribución. La mayoría de las distribuciones se encuadran en uno de los dos campos de tecnologías de empaquetado: el campo .deb de Debian y el campo .rpm de Red Hat. Existen algunas excepciones importantes, como Gentoo, Slackware y Arch, pero la mayoría de los demás utilizan uno de estos dos sistemas básicos, como se muestra en la Tabla 14-1.

Table 14-1: Major Packaging System Families
Tabla 14-1: Principales familias de sistemas de embalaje

| Packaging system | Distributions (partial listing) |
|----------------------|--|
| Debian-style (.deb) | Debian, Ubuntu, Linux Mint, Raspbian |
| Red Hat-style (.rpm) | Fedora, CentOS, Red Hat Enterprise Linux, OpenSUSE |

How a Package System Works

Cómo funciona un sistema de paquetes

The method of software distribution found in the proprietary software industry usually entails buying a piece of installation media such as an “install disk” or visiting each vendor’s website and downloading the product and then running an “installation wizard” to install a new application on the system.

El método de distribución de software que se encuentra en la industria del software propietario generalmente implica comprar un medio de instalación como un "disco de instalación" o visitar el sitio web de cada proveedor y descargar el producto y luego ejecutar un "asistente de instalación" para instalar una nueva aplicación en el sistema.

Linux doesn’t work that way. Virtually all software for a Linux system will be found on the Internet. Most of it will be provided by the distribution vendor in the form of package files, and the rest will be available in source code form that can be installed manually. We’ll talk about how to install software by compiling source code in Chapter 23.

Linux no funciona de esa manera. Prácticamente todo el software para un sistema Linux se encontrará en Internet. La mayor parte será proporcionada por el proveedor de distribución en forma de archivos de paquete, y el resto estará disponible en forma de código fuente que se puede instalar manualmente. Hablaremos sobre cómo instalar software compilando el código fuente en el Capítulo 23.

Package Files

Archivos de paquete

The basic unit of software in a packaging system is the package file. A package file is a compressed collection of files that comprise the software package.

La unidad básica de software en un sistema de empaquetado es el archivo de paquete. Un archivo de paquete es una colección comprimida de archivos que componen el paquete de software.

A package may consist of numerous programs and data files that support the programs. In addition to the files to be installed, the package file also includes metadata about the package, such as a text description of the package and its contents. Additionally, many packages contain pre- and post-installation scripts that perform configuration tasks before and after the package installation.

Un paquete puede constar de numerosos programas y archivos de datos compatibles con los programas. Además de los archivos que se instalarán, el archivo del paquete también incluye metadatos sobre el paquete, como una descripción de texto

del paquete y su contenido. Además, muchos paquetes contienen scripts previos y posteriores a la instalación que realizan tareas de configuración antes y después de la instalación del paquete.

Package files are created by a person known as a package maintainer, often (but not always) an employee of the distribution vendor. The package maintainer gets the software in source code form from the upstream provider (the author of the program), compiles it, and creates the package metadata and any necessary installation scripts. Often, the package maintainer will apply modifications to the original source code to improve the program's integration with the other parts of the Linux distribution.

Los archivos de paquetes son creados por una persona conocida como mantenedor de paquetes, a menudo (pero no siempre) un empleado del proveedor de distribución. El encargado del paquete obtiene el software en forma de código fuente del proveedor original (el autor del programa), lo compila y crea los metadatos del paquete y los scripts de instalación necesarios. A menudo, el mantenedor del paquete aplicará modificaciones al código fuente original para mejorar la integración del programa con las otras partes de la distribución de Linux.

Repositories

Repositorios

While some software projects choose to perform their own packaging and distribution, most packages today are created by the distribution vendors and interested third parties. Packages are made available to the users of a distribution in central repositories that may contain many thousands of packages, each specially built and maintained for the distribution.

Si bien algunos proyectos de software optan por realizar su propio empaquetado y distribución, la mayoría de los paquetes actuales son creados por los proveedores de distribución y terceros interesados. Los paquetes se ponen a disposición de los usuarios de una distribución en repositorios centrales que pueden contener muchos miles de paquetes, cada uno especialmente construido y mantenido para la distribución.

A distribution may maintain several different repositories for different stages of the software development life cycle. For example, there will usually be a "testing" repository that contains packages that have just been built and are intended for use by brave souls who are looking for bugs before the packages are released for general distribution. A distribution will often have a "development" repository where work-in-progress packages destined for inclusion in the distribution's next major release are kept.

Una distribución puede mantener varios repositorios diferentes para diferentes etapas del ciclo de vida del desarrollo de software. Por ejemplo, normalmente habrá un repositorio de "prueba" que contiene paquetes que se acaban de construir y que están destinados a ser usados por almas valientes que buscan errores antes de que los paquetes se publiquen para distribución general. Una distribución a menudo tendrá un repositorio de "desarrollo" donde se guardan los paquetes de trabajo en progreso destinados a ser incluidos en la próxima versión principal de la distribución.

A distribution may also have related third-party repositories. These are often needed to supply software that, for legal reasons such as patents or DRM anti-circumvention issues, cannot be included with the distribution.

Una distribución también puede tener repositorios de terceros relacionados. A menudo, estos son necesarios para suministrar software que, por razones legales como patentes o problemas de prevención de elusión de DRM, no se pueden incluir en la distribución.

Perhaps the best known case is that of encrypted DVD support, which is not legal in the United States. The third-party repositories operate in countries where software patents and anti-circumvention laws do not apply. These repositories are usually wholly independent of the distribution they support, and to use them, one must know about them and manually include them in the configuration files for the package management system.

Quizás el caso más conocido sea el de la compatibilidad con DVD cifrados, que no es legal en los Estados Unidos. Los repositorios de terceros operan en países donde las patentes de software y las leyes contra la elusión no se aplican. Estos repositorios suelen ser totalmente independientes de la distribución a la que dan soporte y, para utilizarlos, es necesario conocerlos e incluirlos manualmente en los archivos de configuración del sistema de gestión de paquetes.

Dependencies

Dependencias

Programs are seldom “stand alone”; rather, they rely on the presence of other software components to get their work done. Common activities, such as input/output, for example, are handled by routines shared by many programs. These routines are stored in shared libraries, which provide essential services to more than one program. If a package requires a shared resource such as a shared library, it is said to have a dependency.

Los programas rara vez son "independientes"; más bien, dependen de la presencia de otros componentes de software para realizar su trabajo. Las actividades comunes, como la entrada / salida, por ejemplo, son manejadas por rutinas compartidas por muchos programas. Estas rutinas se almacenan en bibliotecas compartidas, que brindan servicios esenciales a más de un programa. Si un paquete requiere un recurso compartido, como una biblioteca compartida, se dice que tiene una dependencia.

Modern package management systems all provide some method of dependency resolution to ensure that when a package is installed, all of its dependencies are installed, too.

Todos los sistemas modernos de administración de paquetes proporcionan algún método de resolución de dependencias para garantizar que cuando se instala un paquete, también se instalan todas sus dependencias.

High- and Low-Level Package Tools

Herramientas de paquetes de alto y bajo nivel

Package management systems usually consist of two types of tools.

Los sistemas de gestión de paquetes suelen constar de dos tipos de herramientas.

- Low-level tools that handle tasks such as installing and removing package files
Herramientas de bajo nivel que manejan tareas como instalar y eliminar archivos de paquetes.
- High-level tools that perform metadata searching and dependency resolution
Herramientas de alto nivel que realizan búsquedas de metadatos y resolución de dependencias

In this chapter, we will look at the tools supplied with Debian-style systems (such as Ubuntu and many others) and those used by Red Hat products. While all Red Hat–style distributions rely on the same low-level program (rpm), they use different high-level tools. For our discussion, we will cover the high-level program yum , used by Red Hat Enterprise Linux and CentOS. Other Red Hat–style distributions provide high-level tools with comparable features (see Table 14-2).

En este capítulo, veremos las herramientas proporcionadas con los sistemas de estilo Debian (como Ubuntu y muchos otros) y las que utilizan los productos Red Hat. Si bien todas las distribuciones de estilo Red Hat se basan en el mismo programa de bajo nivel (rpm), utilizan diferentes herramientas de alto nivel. Para nuestra discusión, cubriremos el programa de alto nivel yum, utilizado por Red Hat Enterprise Linux y CentOS. Otras distribuciones de estilo Red Hat proporcionan herramientas de alto nivel con características comparables (consulte la Tabla 14-2).

Table 14-2: Packaging System Tools

Tabla 14-2: Herramientas del sistema de embalaje

| Distributions | Low-level tools | High-level tools |
|--|-----------------|------------------------|
| Debian-style | dpkg | apt-get, apt, aptitude |
| Fedora, Red Hat Enterprise Linux, CentOS | rpm | yum, dnf |

Common Package Management Tasks

Tareas comunes de administración de paquetes

Many operations can be performed with the command line package management tools. We will look at the most common. Be aware that the lowlevel tools also support the creation of package files, an activity outside the scope of this book.

Se pueden realizar muchas operaciones con las herramientas de administración de paquetes de la línea de comandos. Veremos

los más comunes. Tenga en cuenta que las herramientas de bajo nivel también admiten la creación de archivos de paquetes, una actividad fuera del alcance de este libro.

In the discussion that follows, the term `package_name` refers to the actual name of a package rather than the term `package_file` , which is the name of the file that contains the package.

En la discusión que sigue, el término `nombre_paquete` se refiere al nombre real de un paquete en lugar del término `archivo_paquete`, que es el nombre del archivo que contiene el paquete.

Finding a Package in a Repository

Encontrar un paquete en un repositorio

Using the high-level tools to search repository metadata, a package can be located based on its name or description (see Table 14-3).

Utilizando las herramientas de alto nivel para buscar metadatos del repositorio, un paquete puede ubicarse según su nombre o descripción (consulte la Tabla 14-3).

Table 14-3: Package Search Commands
Tabla 14-3: Comandos de búsqueda de paquetes

| Style | Command(s) |
|---------|--|
| Debian | <code>apt-get update</code> <code>apt-cache search search_string</code> |
| Red Hat | <code>yum search search_string</code> |

For example, to search a yum repository for the emacs text editor, we can use this command:
Por ejemplo, para buscar en un repositorio de yum el editor de texto emacs, podemos usar este comando:

```
yum search emacs
```

Installing a Package from a Repository

Instalación de un paquete desde un repositorio

High-level tools permit a package to be downloaded from a repository and installed with full dependency resolution (see Table 14-4).

Las herramientas de alto nivel permiten descargar un paquete de un repositorio e instalarlo con una resolución de dependencia completa (consulte la Tabla 14-4).

Table 14-4: Package Installation Commands
Tabla 14-4: Comandos de instalación de paquetes

| Style | Command(s) |
|---------|--|
| Debian | <code>apt-get update</code> <code>apt-get install package_name</code> |
| Red Hat | <code>yum install package_name</code> |

For example, to install the emacs text editor from an apt repository on a Debian system, we can use this command:
Por ejemplo, para instalar el editor de texto emacs desde un repositorio apt en un sistema Debian, podemos usar este comando:

```
apt-get update; apt-get install emacs
```

Installing a Package from a Package File

Instalación de un paquete desde un archivo de paquete

If a package file has been downloaded from a source other than a repository, it can be installed directly (though without dependency resolution) using a low-level tool (see Table 14-5).
Si un archivo de paquete se ha descargado de una fuente que no sea un repositorio, se puede instalar directamente (aunque sin resolución de dependencias) utilizando una herramienta de bajo nivel (consulte la Tabla 14-5).

Table 14-5: Low-Level Package Installation Commands
Tabla 14-5: Comandos de instalación de paquetes de bajo nivel

| Style | Command(s) |
|---------|----------------------|
| Debian | dpkg -i package_file |
| Red Hat | rpm -i package_file |

For example, if the emacs-22.1-7.fc7-i386.rpm package file had been downloaded from a non-repository site, it would be installed this way:
Por ejemplo, si el archivo del paquete emacs-22.1-7.fc7-i386.rpm se hubiera descargado de un sitio que no sea un repositorio, se instalaría de esta manera:

```
rpm -i emacs-22.1-7.fc7-i386.rpm
```

Note

Nota

Because this technique uses the low-level rpm program to perform the installation, no dependency resolution is performed. If rpm discovers a missing dependency, rpm will exit with an error.
Dado que esta técnica utiliza el programa rpm de bajo nivel para realizar la instalación, no se realiza ninguna resolución de dependencia. Si rpm descubre una dependencia faltante, rpm saldrá con un error.

Removing a Package

Eliminar un paquete

Packages can be uninstalled using either the high-level or low-level tools.
Los paquetes se pueden desinstalar utilizando herramientas de alto o bajo nivel.

Table 14-6 lists the high-level tools.
La Tabla 14-6 enumera las herramientas de alto nivel.

Table 14-6: Package Removal Commands
Tabla 14-6: Comandos de eliminación de paquetes

| Style | Command(s) |
|---------|-----------------------------|
| Debian | apt-get remove package_name |
| Red Hat | yum erase package_name |

For example, to uninstall the emacs package from a Debian-style system, we can use this command:
Por ejemplo, para desinstalar el paquete emacs de un sistema estilo Debian, podemos usar este comando:

```
apt-get remove emacs
```

Updating Packages from a Repository

Actualización de paquetes desde un repositorio

The most common package management task is keeping the system up-to-date with the latest versions of packages. The high-level tools can perform this vital task in a single step (see Table 14-7).
La tarea de administración de paquetes más común es mantener el sistema actualizado con las últimas versiones de paquetes. Las herramientas de alto nivel pueden realizar esta tarea vital en un solo paso (consulte la Tabla 14-7).

Table 14-7: Package Update Commands
Tabla 14-7: Comandos de actualización de paquetes

| Style | Command(s) |
|---------|---------------------------------|
| Debian | apt-get update; apt-get upgrade |
| Red Hat | yum update |

For example, to apply all available updates to the installed packages on a Debian-style system, we can use this command:
Por ejemplo, para aplicar todas las actualizaciones disponibles a los paquetes instalados en un sistema estilo Debian, podemos usar este comando:

```
apt-get update; apt-get upgrade
```

Upgrading a Package from a Package File

Actualización de un paquete desde un archivo de paquete

If an updated version of a package has been downloaded from a non repository source, it can be installed, replacing the previous version (see Table 14-8).
Si se descargó una versión actualizada de un paquete de una fuente que no es un repositorio, se puede instalar, reemplazando la versión anterior (consulte la Tabla 14-8).

Table 14-8: Low-Level Package Upgrade Commands
Tabla 14-8: Comandos de actualización de paquetes de bajo nivel

| Style | Command(s) |
|---------|-----------------------------------|
| Debian | <code>dpkg -i package_file</code> |
| Red Hat | <code>rpm -U package_file</code> |

For example, to update an existing installation of emacs to the version contained in the package file emacs-22.1-7.fc7-i386.rpm on a Red Hat system, we can use this command:

Por ejemplo, para actualizar una instalación existente de emacs a la versión contenida en el archivo de paquete emacs-22.1-7.fc7-i386.rpm en un sistema Red Hat, podemos usar este comando:

```
rpm -U emacs-22.1-7.fc7-i386.rpm
```

Note

Nota

dpkg does not have a specific option for upgrading a package versus installing one as rpm does.
dpkg no tiene una opción específica para actualizar un paquete en lugar de instalar uno como lo hace rpm.

Listing Installed Packages

Listado de paquetes instalados

Table 14-9 lists the commands we can use to display a list of all the packages installed on the system.
La Tabla 14-9 enumera los comandos que podemos usar para mostrar una lista de todos los paquetes instalados en el sistema.

Table 14-9: Package Listing Commands
Tabla 14-9: Comandos de listado de paquetes

| Style | Command(s) |
|---------|----------------------|
| Debian | <code>dpkg -l</code> |
| Red Hat | <code>rpm -qa</code> |

Determining Whether a Package Is Installed

Establecer dónde un paquete está instalado

Table 14-10 lists the low-level tools we can use to display whether a specified package is installed.
La Tabla 14-10 enumera las herramientas de bajo nivel que podemos usar para mostrar si un paquete específico está instalado.

Table 14-10: Package Status Commands
Tabla 14-10: Comandos de estado del paquete

| Style | Command(s) |
|-------|------------|
|-------|------------|

| Style | Command(s) |
|---------|-----------------------------------|
| Debian | <code>dpkg -s package_name</code> |
| Red Hat | <code>rpm -q package_name</code> |

For example, to determine whether the emacs package is installed on a Debian-style system, we can use this:
Por ejemplo, para determinar si el paquete emacs está instalado en un sistema estilo Debian, podemos usar esto:

```
dpkg -s emacs
```

Displaying Information About an Installed Package

Mostrar la Informacion sobre un paquete instalado

If the name of an installed package is known, we can use the commands in Table 14-11 to display a description of the package.
Si se conoce el nombre de un paquete instalado, podemos usar los comandos de la Tabla 14-11 para mostrar una descripción del paquete.

Table 14-11: Package Information Commands
Tabla 14-11: Comandos de información del paquete

| Style | Command(s) |
|---------|--|
| Debian | <code>apt-cache show package_name</code> |
| Red Hat | <code>yum info package_name</code> |

For example, to see a description of the emacs package on a Debian-style system, we can use the following:
Por ejemplo, para ver una descripción del paquete emacs en un sistema estilo Debian, podemos usar lo siguiente:

```
apt-cache show emacs
```

Finding Which Package Installed a File

Encontrar qué paquete instaló un archivo

To determine what package is responsible for the installation of a particular file, we can use the commands in Table 14-12.

Table 14-12: Package File Identification Commands
Para determinar qué paquete es responsable de la instalación de un archivo en particular, podemos usar los comandos de la Tabla 14-12.

| Style | Command(s) |
|---------|--------------------------------|
| Debian | <code>dpkg -S file_name</code> |
| Red Hat | <code>rpm -qf file_name</code> |

To see what package installed the `/usr/bin/vim` file on a Red Hat system, we can use the following:

Para ver qué paquete instaló el archivo `/usr/bin/vim` en un sistema Red Hat, podemos usar lo siguiente:

```
rpm -qf /usr/bin/vim
```

Summing Up

Resumen

In the chapters that follow, we will explore many different programs covering a wide range of application areas. While most of these programs are commonly installed by default, we may need to install additional packages if the necessary programs are not already installed on our system. With our newfound knowledge (and appreciation) of package management, we should have no problem installing and managing the programs we need.

En los capítulos que siguen, exploraremos muchos programas diferentes que cubren una amplia gama de áreas de aplicación. Si bien la mayoría de estos programas se instalan comúnmente de forma predeterminada, es posible que necesitemos instalar paquetes adicionales si los programas necesarios aún no están instalados en nuestro sistema. Con nuestro nuevo conocimiento (y apreciación) de la administración de paquetes, no deberíamos tener problemas para instalar y administrar los programas que necesitamos.

The Linux Software Installation Myth

El mito de la instalación del software Linux

People migrating from other platforms sometimes fall victim to the myth that software is somehow difficult to install under Linux and that the variety of packaging schemes used by different distributions is a hindrance. Well, it is a hindrance, but only to proprietary software vendors that want to distribute binary-only versions of their secret software.

Las personas que migran desde otras plataformas a veces son víctimas del mito de que el software es de alguna manera difícil de instalar en Linux y que la variedad de esquemas de empaquetado utilizados por diferentes distribuciones es un obstáculo. Bueno, es un obstáculo, pero solo para los proveedores de software propietario que quieran distribuir versiones binarias de su software secreto.

The Linux software ecosystem is based on the idea of open source code. If a program developer releases source code for a program, it is likely that a person associated with a distribution will package the program and include it in their repository. This method ensures that the program is well integrated into

El ecosistema de software de Linux se basa en la idea del código fuente abierto. Si un desarrollador de programas publica el código fuente de un programa, es probable que una persona asociada con una distribución empaque el programa y lo incluya en su repositorio. Este método asegura que el programa esté bien integrado en

the distribution, and the user is given the convenience of “one-stop shopping” for software, rather than having to search for each program’s website. Recently, major proprietary platform vendors have begun building application stores that mimic this idea.

distribución, y el usuario tiene la conveniencia de “comprar en un solo lugar” el software, en lugar de tener que buscar el sitio web de cada programa. Recientemente, los principales proveedores de plataformas patentadas han comenzado a crear tiendas de aplicaciones que imitan esta idea.

Device drivers are handled in much the same way, except that instead of being separate items in a distribution’s repository, they become part of the Linux kernel. Generally speaking, there is no such thing as a “driver disk” in Linux. Either the kernel supports a device or it doesn’t, and the Linux kernel supports a lot of devices—many more, in fact, than Windows does. Of course, this is of no consolation if the particular device you need is not supported. When that happens, you need to look at the cause. A lack of driver support is usually caused by one of three things.

Los controladores de dispositivos se manejan de la misma manera, excepto que en lugar de ser elementos separados en el repositorio de una distribución, se convierten en parte del kernel de Linux. En términos generales, no existe un “disco de controlador” en Linux. O el kernel admite un dispositivo o no, y el kernel de Linux es compatible con muchos dispositivos; de hecho, muchos más que Windows. Por supuesto, esto no sirve de consuelo si el dispositivo en particular

que necesita no es compatible. Cuando eso sucede, debe buscar la causa. La falta de asistencia del conductor suele deberse a una de estas tres cosas.

- The device is too new. Since many hardware vendors don't actively support Linux development, it falls upon a member of the Linux community to write the kernel driver code. This takes time.

El dispositivo es demasiado nuevo. Dado que muchos proveedores de hardware no apoyan activamente el desarrollo de Linux, le corresponde a un miembro de la comunidad de Linux escribir el código del controlador del kernel. Esto lleva tiempo.

- The device is too exotic. Not all distributions include every possible device driver. Each distribution builds its own kernels, and since kernels are configurable (which is what makes it possible to run Linux on everything from wristwatches to mainframes), they may have overlooked a particular device. By locating and downloading the source code for the driver, it is possible for you (yes, you) to compile and install the driver yourself. This process is not overly difficult, but it is rather involved. We'll talk about compiling software in a later chapter.

El dispositivo es demasiado exótico. No todas las distribuciones incluyen todos los controladores de dispositivo posibles. Cada distribución construye sus propios núcleos, y dado que los núcleos son configurables (que es lo que hace posible ejecutar Linux en todo, desde relojes de pulsera hasta mainframes), es posible que hayan pasado por alto un dispositivo en particular. Al localizar y descargar el código fuente del controlador, es posible que usted (sí, usted) compile e instale el controlador usted mismo. Este proceso no es demasiado difícil, pero sí complicado. Hablaremos sobre la compilación de software en un capítulo posterior.

- The hardware vendor is hiding something. It has neither released source code for a Linux driver nor has it released the technical documentation for somebody to create one for them. This means the hardware vendor is trying to keep the programming interfaces to the device a secret. Because we don't want secret devices in our computers, it is best that you avoid such products.

El proveedor de hardware está ocultando algo. No ha publicado el código fuente para un controlador de Linux ni ha publicado la documentación técnica para que alguien cree uno para ellos. Esto significa que el proveedor de hardware está tratando de mantener en secreto las interfaces de programación del dispositivo. Debido a que no queremos dispositivos secretos en nuestras computadoras, es mejor que evite dichos productos.

```
paquetes: dpkg apt

# comando de busqueda paquetes
apt-get update
apt-cache search search_string

# Instalar un paquete desde un repositorio
apt-get update
apt-get install package_name

# ejemplo instalar el editor de texto emacs
apt-get update; apt-get install emacs

# Instalar paquete desde linea de comandos
dpkg -i package_file
rpm -i package_file
# por ejemplo el package emacs-22.7.fc7-i386.rpm que ha sido descargado de un repositorio se instalará:
rpm -i emacs-22.7.fc7-i386.rpm

# Actualizando los paquetes (packages) del repositorio:

apt-get update; apt-get upgrade

# Actualizando un paquete desde un fichero de paquete:

dpkg -i package_file
rpm -U package_file

ejemplo: rpm -U emacs-22.7.fc7-i386.rpm

# Listar paquetes instalados:

dpkg -l
```

```
rpm -qa
```

```
# Determinar donde un paquete está instalado:
```

```
dpkg -s package_name  
rpm -q package_name
```

```
# ejemplo: dpkg -s emacs
```

```
# Mostrando información sobre un paquete instalado:
```

```
apt-cache show package_name
```

```
# ejemplo: apt-cache show emacs
```

```
# Encontrar cual paquete que archivo:
```

```
dpkg -S file_name  
rpm -qf file_name
```

```
# ejemplo: rpm -qf /usr/bin/vim
```
