

Administración y configuración de GNU/Linux – Parte II

1. Instalación y gestión de paquetes	1
Paquetes deb	1
APT (<i>Advanced Packaging Tool</i>)	2
dpkg (<i>Debian Package Management System</i>)	3
Instalación de diferentes entornos gráficos	3
2. Otros tipos de paquetes	4
Paquetes tgz	4
Comandos para comprimir y empaquetar	4
Paquetes rpm	7

1. Instalación y gestión de paquetes

Una vez instalado el sistema operativo, muchas veces necesitaremos instalar software o actualizar el sistema operativo. Esto se hace mediante los **paquetes**. Un paquete es un **conjunto de ficheros que se utilizan para instalar una aplicación**. En los paquetes encontramos los ficheros ejecutables, archivos de configuración, ayuda sobre el paquete y su instalación, y también puede contener el código fuente de la aplicación.

A los paquetes que contienen **código máquina y no código fuente** se les llama **paquetes binarios**. En Linux, existen muchos tipos de paquetes binarios dependiendo de la distribución que tengamos instalada.

- **Paquetes RPM.** Los usan las distribuciones RedHat, SUSE, OpenSUSE, Mageia (Mandriva) o Fedora. Se gestionan mediante la **aplicación rpm**.
- **Paquetes deb.** Usados por la distribuciones Debian y derivadas, como **Ubuntu**. Se gestionan mediante las **aplicaciones apt, aptitude y dpkg**.
- **Paquetes tgz.** Usados por la distribución Slackware.

Paquetes deb

Los paquetes de tipo deb, usados sobre todo en Ubuntu, están mantenidos por una gran comunidad de desarrolladores y el proyecto está patrocinado por la empresa Canonical Ltd., que ofrece soporte y servicios para Ubuntu. Los paquetes deb tienen el siguiente formato:

nombre-del-paquete_version(n.n.nn-n).deb

El fichero **/etc/apt/sources.list** es necesario para la instalación de paquetes, ya que guarda las direcciones donde se descargan los paquetes. Se pueden descargar desde varios sitios, como un fichero, un CD o DVD, vía Internet por http, por ftp, etc.

APT (*Advanced Packaging Tool*)

Esta utilidad ofrece varios comandos, como **apt-get**, **apt-cache**, **apt-cdrom**. El comando **aptitude**, que es una versión mejorada de **apt**, se puede utilizar con las mismas opciones que este. Existe la versión **apt-rpm**, que es una versión para trabajar con tipos de paquetes RPM.

En el directorio **/etc/apt/apt.conf.d** podremos encontrar archivos de configuración de APT.

apt-get

El comando **apt-get** se utiliza para la manipulación de paquetes deb. Permite la instalación de paquetes, borrado, etc.

Opciones:

- i | --install** → instala un paquete.
- d | --download-only** → descarga paquetes deb en **/var/cache/apt/archives/** sin instalar o desinstalar.
- f | --fix-broken** → arregla los problemas de dependencias que haya.
- s | --simulate** → simula, sin llevar a cabo la acción.
- b | --build** → compila el paquete de código fuente.

apt-get install paquete/s → instala paquetes por la red.

apt-get install --reinstall paquete → reinstala un paquete.

apt-get remove paquete/s → desinstala paquetes.

apt-get source paquete/s → descarga el código fuente de los paquetes.

apt-get update → actualiza la lista de paquetes disponibles desde los repositorios del fichero **sources.list**.

apt-get upgrade → instala las nuevas versiones de los diferentes paquetes disponibles.

apt-get dist-upgrade → actualiza todos los paquetes resolviendo las dependencias.

apt-get purge paquete/s → elimina el paquete y sus archivos de configuración.

apt-get remove --purge paquete/s → desinstala un paquete y sus archivos de configuración.

apt-get build-dep paquete/s → instala los paquetes necesarios para la compilación del código fuente.

apt-get autoclean → elimina los archivos descargados antiguos.

apt-get check → verifica que no haya dependencias incumplidas.

apt-get clean → elimina los archivos descargados.

Elimina los ficheros que se encuentran en **/var/cache/apt/archives** y **/var/cache/apt/archives/partial**. Ahí se encuentran los paquetes deb que hemos descargado para instalar.

apt-cache

El comando **apt-cache** trabaja con la caché de los paquetes de APT buscando información. Este comando **no manipula el estado del sistema, sino que nos muestra información sobre los paquetes**, así que lo pueden usar usuarios normales. Algunas de las opciones de este comando son:

apt-cache show paquete/s → muestra la cabecera (información) de los paquetes.

apt-cache search texto → muestra una lista de todos los paquetes que concuerden con un texto.

apt-cache depends paquete → muestra las dependencias del paquete.

apt-cache stats → muestra las estadísticas del caché.

apt-cache showpkg paquete → muestra su relación con otros paquetes.

apt-cache rdepend paquete → muestra las dependencias inversas del paquete.

apt-cdrom

El comando **apt-cdrom** **permite añadir un CD o DVD al fichero /etc/apt/sources.list**. Mantiene una lista de todos los CD o DVD añadidos en el fichero **/var/lib/apt/cdroms.list**. La forma de utilizarlo es:

apt-cdrom add

dpkg (Debian Package Management System)

dpkg

También podemos usar este comando para gestionar los paquetes.

dpkg --list → lista todos los paquetes instalados.

dpkg -L nombre-paquete → muestra los archivos que conforman el paquete.

dpkg -S texto → muestra los paquetes con algún archivo que coincida con texto.

dpkg -i paquete → instalación de paquetes .deb.

dpkg -r paquete → desinstala un paquete.

dpkg --purge paquete → desinstala un paquete y sus archivos de configuración.

dpkg --force -r paquete → fuerza la desinstalación de un paquete.

dpkg --force-all -r paquete → fuerza la desinstalación de un paquete si no ha funcionado la anterior.

dpkg -c paquete → muestra los ficheros de un paquete.

dpkg -L paquete → muestra todos los ficheros que se instalaron con un paquete.

dpkg -S fichero → muestra a qué paquete pertenece un fichero.

dpkg --get-selections → muestra todos los paquetes instalados en el sistema.

dpkg --reconfigure paquete → reconfigura paquetes instalados.

Instalación de diferentes entornos gráficos

Una de los usos que podemos darle al instalador de paquetes, es instalar otros gestores de ventanas. Los podemos instalar mediante el **Gestor de paquetes Synaptics**, o bien mediante los **comandos apt, aptitude o dpkg**.

Una vez instalados otros gestores de ventanas, para iniciar la sesión con ellos, en la ventana de entrada, elegimos Sesiones y nos sale un menú donde podremos elegir con qué gestor de ventanas queremos arrancar, y si lo queremos dejar por defecto. Si tenemos instalado más de un gestor de ventanas, podemos elegir cuál será el predeterminado: **gdm es el de Gnome y kdm es el de KDE**.

Instalamos el gestor de ventanas KDE desde la línea de comando.

```
paco@ubuntupaco:~$ sudo apt-get install kubuntu-desktop
paco@ubuntupaco:~$ sudo apt-get install kde-i18n-es
paco@ubuntupaco:~$ sudo apt-get install language-pack-kde-es
paco@ubuntupaco:~$ sudo apt-get install kubuntu-restricted-extras
```

Instalamos el gestor de ventanas XFCE desde la línea de comando.

```
paco@ubuntupaco:~$ sudo apt-get install xubuntu-desktop
paco@ubuntupaco:~$ sudo apt-get install xubuntu-restricted-extras
paco@ubuntupaco:~$ sudo apt-get install xfce4
```

Instalamos el gestor de ventanas Fluxbox desde la línea de comando.

```
paco@ubuntupaco:~$ sudo apt-get install fluxbox
paco@ubuntupaco:~$ sudo apt-get install fluxconf
```

Instalamos de IceWE desde la línea de comando.

```
paco@ubuntupaco:~$ sudo apt-get install IceWE
```

Cualquier gestor de ventanas, una vez instalado, si no lo queremos usar, lo podemos desinstalar mediante el comando **apt-get remove** y desinstalamos los paquetes que instalamos.

2. Otros tipos de paquetes

Paquetes tgz

Contienen el código fuente del programa con los archivos necesarios para compilar e instalar el programa mediante los comandos **tar** para **empaquetar** y **gzip** o **bzip** para **comprimir**. Este tipo de paquetes, propio de la distribución Slackware, se puede utilizar en otras distribuciones. El paquete está en un tipo de fichero **.tar.gz** o **.tar.bz2**. Una vez desempaquetado y descomprimido, el paquete se instala desde el directorio donde están las fuentes del programa, con los siguientes comandos:

```
./configure
make
make install
```

Comandos para comprimir y empaquetar

Durante muchos años, se utilizó en lugar de **gzip** el comando **compress**. Sin embargo, debido a varios factores (una disputa por una patente software contra el algoritmo de **compress** y el hecho de que **gzip** es mucho más eficiente que **compress**), **compress** ha quedado desfasado.

Los ficheros comprimidos con **compress** terminan en la extensión **.Z**. El comando **uncompress** se utiliza para expandir un fichero comprimido con **compress**; **gunzip puede descomprimir los ficheros comprimidos con compress**.

Ficheros gz

Comprimir

```
gzip -r fichero
```

La opción -r significa modo recursivo en los directorios inferiores.

Descomprimir

```
gzip -d fichero.gz  
gunzip fichero.gz
```

Ver contenido

```
gzip -c fichero.gz
```

Ficheros bz2

Comprimir

```
bzip2 fichero
```

Descomprimir

```
bzip2 -d fichero.bz2  
bunzip2 fichero.bz2
```

Ver contenido

```
bzip2 -c fichero.bz2
```

Ficheros zip

Comprimir

```
zip -r fichero.zip fichero
```

La opción -r significa modo recursivo en los directorios inferiores.

Descomprimir

```
unzip archivo.zip
```

Ver contenido

```
unzip -v archivo.zip
```

Ficheros rar

No viene instalado por defecto. Para instalar el paquete:

```
sudo apt-get install rar  
sudo apt-get install unrar
```

Comprimir

```
rar -a archivo.rar directorio
```

Descomprimir (se extrae en el directorio actual)

```
unrar e -r archivo.rar
```

Descomprime (x extrae donde se indique)

```
unrar x -r archivo.rar directorio
```

Ver contenido

```
unrar -v archivo.rar  
untar -l archivo.rar
```

Mientras que zip comprime y empaqueta, gzip o bzip2 solo comprimen ficheros, no directorios. Para empaquetar con gzip o bzip2, utilizamos tar.

Ficheros tar

Empaquetar (ficheros o directorios)

```
tar -vcf archivo.tar fichero1 fichero2
```

Desempaquetar

```
tar -vxf archivo.tar
```

Ver contenido

```
tar -vtf archivo
```

Opciones de tar:

-c | **--create** → crea un nuevo archivo.

-f → si se usa con la opción -c, usa el nombre del fichero especificado para la creación del fichero tar. Si se usa con la opción -x, retira del archivo el fichero especificado.

-t | **--list** → muestra la lista de los ficheros que se encuentran en el fichero tar.

-v | **--verbose** → muestra el proceso de archivo de los ficheros.

-x | **--extract** → extrae los ficheros de un archivo.

-z | **--gzip** → comprime el fichero tar con gzip.

-j | **--bzip2** → comprime el fichero tar con bzip2.

Para comprimir varios ficheros y empaquetarlos en un solo archivo hay que combinar el tar y el gzip o el bzip2.

Ficheros tar.gz (tgz)

Empaquetar y comprimir

```
tar -czvf archivo.tgz directorio
```

Desempaquetar y descomprimir

```
tar -xzvf archivo.tgz
```

Ver contenido

```
tar -tzvf archivo.tgz
```

Ficheros tar.bz2 (tbz2)

Empaquetar y comprimir

```
tar -jvcf archivo.tbz2 directorio
```

Desempaquetar y descomprimir

```
tar -jvxf archivo.tbz2
```

Ver contenido

```
tar -jvtf archivo.tbz2
```

Creamos un directorio en nuestro directorio personal llamado **ejercicios** que contiene dos archivos de texto, **a** y **b** además de un directorio **ejercc**, que contiene el archivo de texto **c**. A continuación, desde el directorio personal, comprimimos el directorio ejercicios y su contenido, utilizando **gzip**, **bzip2** y **zip**. Una vez creados, comprobamos el contenido de los ficheros comprimidos.

```
paco@ubuntupaco:~$ mkdir ejercicios
paco@ubuntupaco:~$ cd ejercicios
paco@ubuntupaco:~$ nano a
paco@ubuntupaco:~$ nano b
paco@ubuntupaco:~$ mkdir ejercc
paco@ubuntupaco:~$ cd ejercc nanoc
paco@ubuntupaco:~$ cd zip -r ejercicios zip /ejercicios
paco@ubuntupaco:~$ unzip -v ejercicios.zip
paco@ubuntupaco:~$ gzip -r ejercicios
```

La aplicación gzip no empaqueta, sino que comprime cada fichero.

```
paco@ubuntupaco:~$ ls -lR ejercicios
paco@ubuntupaco:~$ gunzip -r ejercicios
paco@ubuntupaco:~$ bzip2 ejercicios/*
```

La aplicación bzip2 no empaqueta, sino que va comprimiendo cada fichero.

```
paco@ubuntupaco:~$ bzip2 ejercicios/ejercc/c
paco@ubuntupaco:~$ ls -lR ejercicios
```

Paquetes rpm

Los **paquetes RPM (Red Hat Package Manager)**, deben su nombre a que **fueron creados por Red Hat**, aunque más adelante fue **adoptándose por otras distribuciones**. Los paquetes se identifican por la siguiente sintaxis:

nombre-versión_aplicación-versión_paquete.arquitectura.rpm

Para los paquetes rpm tenemos las utilidades de gestión de paquetes **YaST** (Yet another Setup Tool) y **yum**. YaST cuenta con una interfaz gráfica llamada YaST2.

Para instalar paquetes RPM en modo texto usaremos el siguiente comando:

rpm

Instala, desinstala y actualiza los paquetes RPM.

Sintaxis1:

rpm [opciones] [paquetes]

Sintaxis2:

crontab fichero

Opciones:

-i | --install → instala un paquete.

-v | --verbose → muestra por pantalla el trabajo que se está realizando.

-h | --hash → marca el proceso de instalación con el carácter # mientras se instala el paquete.

--percent → muestra el porcentaje de ficheros de un paquete que se van extrayendo.

--test → realiza una comprobación de instalación de un paquete, para ver que no da problemas, pero no lo instala.

--nodeps → no verifica las dependencias antes de instalar un paquete.

--force → fuerza la instalación de un paquete aunque surjan problemas.

-U | --upgrade → actualiza los paquetes ya instalados.

-e → desinstala los paquetes.