



Gestión de Paquetes de Software en Debian





Instalación de SW en GNU/Linux

- La instalación de programas es diferente a la de otros sistemas operativos.
- Los desarrolladores siguen un estándar para su software y su instalador.
- Cada programa a ser instalado se representa como un paquete de software.
- Garantizan la compatibilidad y el correcto funcionamiento del software.





Paquetes de Software



- Un paquete es un conjunto de archivos comprimidos que contiene lo necesario para instalar, desinstalar, configurar y ejecutar un programa.
- Puede incluir bibliotecas, archivos de configuración, documentación y otros recursos.
- Se administran con **herramientas de gestión de software del SO.**
- Un programa puede estar contenido en un solo paquete o consistir en varios paquetes relacionados entre ellos.
- Varios programas pequeños y relacionados entre sí pueden estar en el mismo paquete (ej: paquete fileutils que contiene varios comandos como ls, cp, etc.)



Relación entre Paquetes

Al instalar un programa, los paquetes de software pueden tener relaciones que determinan si se pueden o deben instalar juntos.

Dependencia

Un paquete depende de otro para funcionar correctamente.

Recomendación

Un paquete recomienda otro si el mantenedor considera que la mayoría de los usuarios querrán la funcionalidad del segundo.

Sugerencia

Un paquete sugiere otro si el segundo contiene archivos que mejoran la funcionalidad del primero, pero no son necesarios.

Conflicto

Dos paquetes entran en conflicto si no se pueden instalar al mismo tiempo.

Reemplazo

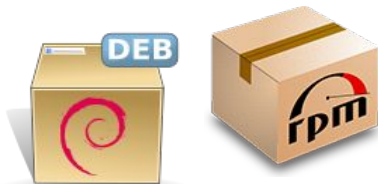
Un paquete reemplaza a otro cuando los archivos del segundo se eliminan y se sobrescriben con los del primero.



Tipos de paquetes de SW

Paquetes Binarios

Contienen los archivos para reconstruir una aplicación dentro del sistema. Es software precompilado. Se reconocen por ser **.deb** (Debian y derivados) y **.rpm** (Red Hat, SUSE, etc.)



Paquetes de Código

Contienen los códigos fuente y los archivos necesarios para compilar e instalar en forma manual los programas. Son comprimidos y se reconocen por ser **.tgz**, **.tar.gz** o **.tar.bz2**





Formas de instalar SW

GESTOR DE PAQUETES

APT

Comandos de alto nivel para administrar paquetes. Gestiona dependencias.

Interfaz gráfica de apt:
Synaptic

Centro de Software (GUI)

DPKG

Programa de bajo nivel que gestiona paquetes .deb. No gestiona dependencias

Descargar, extraer el código fuente, compilarlo e instalarlo.

CÓDIGO FUENTE

1

Gestor de Paquetes

Programa para administrar paquetes



Gestor de Paquetes: apt

- Es una colección de herramientas (comandos) para automatizar el proceso de instalación, actualización, configuración y eliminación de paquetes de software. Debian y derivados utilizan las herramientas de alto nivel apt, apt-get.
- Mantiene un registro del SW está instalado lo que permite una gestión de sencilla y centralizada de los programas.
- Forma parte del sistema operativo y posee un único formato de paquetes (estándar).
- Utiliza la información incluida en los paquetes nombre completo, descripción de funcionalidad, versión, y paquetes requeridos para su funcionamiento).
- Resuelve automáticamente las dependencias.



Pasos con gestor apt

El usuario solicita instalación de un programa por medio del comando del Gestor (#apt-get install tree es el pedido para instalar el comando tree)

El Gestor busca el paquete de software del programa en el repositorio definido en el archivo /etc/apt/sources.list

El Gestor ubica los paquetes necesarios, los descarga, desempaqueta e instala. Muestra sugerencias de la instalación. El programa queda listo para ser utilizado.



Repositorios

- Funcionan como almacenes centralizados de SW que permiten instalar y gestionar programas en el sistema operativo. Son similares a las tiendas de aplicaciones de otros sistemas operativos.
- Instalan software de forma eficiente y segura, libre de malware y virus
- Cada distribución mantiene organizados los paquetes en repositorios, lo que permite actualizar e instalar por red todo el sistema desde una localización confiable (repositorios oficiales).
- Pueden ser:
 - Oficiales, mantenidos por la distribución.
 - Comunitarios, mantenidos por la comunidad.
 - Personales, creados y mantenidos por individuos

Los repositorios son configurados en el
archivo

`/etc/apt/sources.list`



Archivo sources.list

- Archivo de texto que contiene una lista de direcciones (fuentes) con las ubicaciones de los repositorios de paquetes de nuestra distribución.
- Antes de instalar programas actualizarlo.
- Se puede editar y agregar otras fuentes:

```
#nano /etc/apt/sources.list
```

- Luego de editar sources.list, siempre actualizar:

```
#apt update
```

```
GNU nano 7.2 /etc/apt/sources.list *
#deb cdrom:[Debian GNU/Linux 12.6.0 _Bookworm_ - Official amd64 NETINST with firmware 20240629-10:1]

deb http://deb.debian.org/debian/ bookworm main non-free
deb-src http://deb.debian.org/debian/ bookworm main non-free

deb http://security.debian.org/debian-security bookworm-security main non-free
deb-src http://security.debian.org/debian-security bookworm-security main non-free

# bookworm-updates, to get updates before a point release is made;
# see https://www.debian.org/doc/manuals/debian-reference/ch02.en.html#_updates_and_backports
deb http://deb.debian.org/debian/ bookworm-updates main non-free
deb-src http://deb.debian.org/debian/ bookworm-updates main non-free

# This system was installed using small removable media
# (e.g. netinst, live or single CD). The matching "deb cdrom"
# entries were disabled at the end of the installation process.

^G Ayuda      ^O Guardar    ^W Buscar    ^K Cortar    ^T Ejecutar  ^C Ubicación  M-U Deshacer
^X Salir      ^R Leer fich. ^_ Reemplazar ^U Pegar     ^J Justificar ^/ Ir a línea  M-E Rehacer
```

- ▣ Ubicaciones de repositorios
- ▣ Secciones
- ▣ # comentarios



```
deb http://deb.debian.org/debian/ bookworm main non-free
deb-src http://deb.debian.org/debian/ bookworm main non-free
```

- **deb** (típico paquete binario de Debian) y **deb-src** (código fuente oficial del paquete)
- **uri**: Identificador Universal de Recursos, tipo de recurso de la cual se obtienen los paquetes.
 - CD-ROM: APT usa la unidad de CD-ROM local.
 - FTP: Especifica un servidor FTP como archivo.
 - HTTP: Especifica un servidor HTTP como archivo.
 - FILE: Permite considerar como archivo a cualquier fichero en el sistema de ficheros.
- **distribución** : Distribución instalada.
- **Secciones** : indican las ramas de software disponibles, como main, contrib, y non-free. Estas secciones especifican el tipo de software incluido en cada repositorio.



Secciones

main

- Directorio con paquetes 100% libres; esto quiere decir que cumplen o están de acuerdo con las directivas de Debian, en donde marcan cuando un paquete se le puede considerar que es 100% software libre.

contrib

- Incluye software libre que depende de componentes no libres para funcionar correctamente. Aunque el software en esta sección es libre, puede requerir otros programas o bibliotecas que no son libres.

non-free

- Contiene software que no cumple con las Directrices de Software Libre de Debian. Esto incluye software propietario o con restricciones en su uso, modificación o redistribución.



Algunos Comandos de apt

- Instalar software: `#apt install <paquete>`
- Actualizar la lista de paquetes disponibles: `#apt update`
- Buscar palabras en la lista de paquetes disponibles: `#apt search <palabra>`
- Buscar palabras en la lista de paquetes disponibles: `#apt search <palabra>`
- Listar todos los paquetes disponibles para instalar o instalados en el sistema: `#apt list`
- Reparar e instalar dependencias pendientes: `#apt -f install`
- Desinstalar el paquete del sistema: `#apt remove <paquete>`
- Muestra la descripción completa del paquete: `#apt-cache show <paquete>`
- Obtener el código fuente de los paquetes: `#apt source <paquete>`

2

dpkg

El comando `dpkg` permite instalar, eliminar y gestionar paquetes `.deb`. Es el gestor de paquetes principal para Debian y sistemas basados en Debian.



- Instala paquetes descargados de fuentes distintas al gestor de paquetes o repositorios predeterminados
- A diferencia de apt no instala automáticamente las dependencias. Se limita a indicarlas durante el proceso de instalación.
- Una vez instalado un paquete .deb con dpkg, se debe ejecutar la línea de comandos para completar las dependencias faltantes o instalarlas de a una con dpkg si se dispone de los paquetes de las dependencias.



Algunos usos de dpkg

- Instalar software: `#dpkg -i <paquete.deb>`
- listar los paquetes instalados: `#dpkg -l`
- Desinstalar un paquete: `#dpkg -r <paquete.deb>`
- Comprobar si un paquete este instalado: `#dpkg -s <paquete.deb>`
- Saber la ubicación, archivos y documentación de un paquete.: `#dpkg -L <paquete.deb>`
- Comprobar la licencia del paquete dpkg: `#dpkg --version`
- Ver ayuda del comando dpkg: `#dpkg --help`



Ejemplo de instalación con dpkg

- Abrimos una terminal y nos logueamos como superusuario o usuario con privilegios de root (sudo). Nos ubicamos en el directorio en el cual descargamos el paquete:

```
#cd /home/usuario/Descargas
```

- Ejecutamos la instalación:

```
# dpkg -i paquete.deb
```

- El paquete se instalará y puede suceder que muestre un error si hay dependencias no satisfechas, en este caso, se resuelven con:

```
# apt -f install
```

- El software queda listo para ejecutar.

3

Instalación desde Código fuente

Compilar y crear un ejecutable compatible con nuestra distribución

- Si no hay paquetes hechos para nuestra distribución, o no se encuentran actualizados en los repositorios, podemos descargar la versión de ese programa en CÓDIGO FUENTE (Source Code) desde la página oficial del programa o de desarrollo del mismo .
- Es el programa escrito por el programador. Entonces para poder usar el programa debemos compilarlo y crear un ejecutable compatible con nuestra distribución. Generalmente estos códigos vienen en lenguaje C y C++, aunque pueden aparecer en Python, Ruby, etc.
- Suelen presentarse en formato .tar.gz o tar.bz2 (o sea compactado con tar y comprimido con gzip o bzip).
- Generalmente tienen información en el archivo README o INSTALL





Ejemplo de Instalación desde el código fuente

1º) Descargar el código fuente (con apt source o de web) miprograma.tar.gz en un directorio (/opt) y situarse en el directorio: `root@mipc:~#cd /opt`

2º) Descomprimir el archivo: `root@mipc:/opt#tar -xzvf miprograma.tar.gz`

3º) Al descomprimir, se creará un directorio con el nombre del archivo comprimido. Situarse en el directorio producto de la descompresión: `root@mipc:/opt#cd miprograma`

4º) Ejecutar el script configure el cual prepara al sistema para compilar el programa:

```
root@mipc:/opt/miprograma#./configure
```

5º) Compilar el código fuente con el comando make para generar el código máquina o binarios ejecutables del programa: `root@mipc:/opt/miprograma#make`

6º) Copiar los binarios generados y archivos necesarios a los respectivos directorios con make install:

```
root@mipc:/opt/miprograma#make install
```

¡Gracias!



¿Preguntas?