instalación de programas en GNU/Linux

por
Matías nnss Palomec
matias@programado.org

definiciones mínimas

- Paquete: Es un archivo que tiene dentro programas o librerías necesarios para cumplir una función en especial, un programa, o un conjunto de funciones
- Librería: Es un conjunto de archivos (o solo uno) que el usuario no puede ejecutar directamente, pero los programas las requieren. Usualmente comparten funcionalidades con varios programas, de forma que no tienen que incluir todo el código en los programas

definiciones mínimas

- Programa: es una utilidad que el usuario ejecuta, utiliza, y posiblemente también termine en algún momento.
- tarball: Literalmente es una pelota de alquitran, y es un archivo parecido al .zip de Windows®, pero empaquetado de otra forma.
- Compilador: interpreta el lenguaje y genera un archivo binario que un sistema operativo puede interpretar y ejecutar

apt (modo debian y derivados)

apt-cache search paquete
apt-cache show paquete
apt-get install paquete
dpkg -l “patron”
apt-get remove [--purge] paquete

ventajas de los paquetes .deb

Desde hace mucho tiempo tiene un mejor desempeño para la instalación y desinstalación de programas (lo que suele hacer un usuario principiante o un usuario final).

Los paquetes oficiales (sobre todo en Debian) suelen estar probados, y están correctamente indicadas las dependencias, con lo que es difícil que un programa se instale y no funcione por falta de una librería

desventajas de los .deb

No todos los projectos tienen disponibles los paquetes .deb para descargar

Los paquetes no oficiales no están siempre bien hechos (pueden no tener correctamente especificadas las dependencias o instalarse en cualquier lugar)

rpm (modo RH/Fedora y derivados)

rpm -i <archivo>
yum install <paquete>
yum update <paquete>
yum upgrade
yum list
yum search <string>
yum resolvdep <paquete>

desde el código fuente

Uno de los problemas principales que se pueden tener para la instalación de un paquete desde el código fuente es que no se tienen siempre todos los paquetes necesarios para compilarlo.

lo más básico de lo necesario

- gcc: Compiador de C
- make: procesador de macros

Elegir un programa

GPA
(front-end gráfico para manejar claves GnuPG)
http://wald.intevation.org/projects/gpa/

descomprimiendo el archivo

El archivo que se descarga puede ser algo como:
gpa_0.7.0-1.1.tar.gz

para descomprir el tarball:
tar -jxvf gpa_0.7.0-1.1.tar.gz

los modificadores del tar

- El tar tiene variso modificadores, los más básicos son:

- x -> extraer
- f -> luego de la F sigue el archivo (si hay espacio o no no importa)
- v -> muestra más cosas por pantalla (se puede poner dos veces para que muestre aún más información por cada archivo)
- j -> usa el programa bzip2 para descomprimir (extensión .tar.bz2 o .tbz2)
- z -> usa el programa gzip para descomprimir (extensión .tar.gz o .tgz)

los archivos de documentación

Usualmente los programas que se comparten con el código fuente tienen cierta estructura. La documentación que en esta parte más interesa es la que está en el archivo:

INSTALL

los pasos básicos

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

./configure

Al ejecutar el ./configure lo que se hace es ejecutar un script que prueba que esté todo lo necesario para que se pueda compilar el programa. Si no lo tiene, dice que es lo que falta (y en algunos casos dice de donde bajarlo).

Al terminar lo que hace es generar unos arlchivos de nombre Makefile, que son las instrucciones para el comando “make”

./configure --help

El ./configure tiene muchos modificadores, pero no son indispensables para saberlos y usarlos desde el principio (usualmente casi nunca se usan). El único que puede ser de utilidad es el:
--prefix=/path/to

Lo que hace este modificador es poner el directorio /path/to para la instalación del programa, en vez del /usr/local

make

El comando make lo que hace es buscar los archivos Makefile (creados pro el configure) y realiza en orden lo que allí está indicado

make install

Aunque se puede hacer directamente make install, en vez de hacer primero make y luego make install, se suele usar la última por costumbre.

Lo que hace esto es verificar que esté todo compilado, y luego lo instala en el lugar donde el –prefix (o el lugar por defecto) le indicaba.

cando la compilación falla

Hay muchos desarrolladores que el archivo del código fuente es una copia de la última versión de desarrollo, con lo cual suele fallar por problemas de código.
También se puede topar con algún programa que esté hecho con alguna mejora en especial de una nueva versión del GCC, con lo cual se debe usar esa versión.

donde pedir ayuda

Lo mejor, ante un problema con un programa, es pedir ayuda al programador. El lenguaje debe ser amable y cortes, pero sobre todo, hablar en inglés.

También se puede pedir ayuda en listas generales, sobre todo si es algún programa que ya tiene cierto tiempo de desarrollo y es popular.

url de la presentación:
http://programado.org/charlas

licencia de la charla:
http://creativecommons.org/licenses/by-nc-sa/2.5/ar/