

Instalando software desde fuentes en Linux

Existen muchas distribuciones de software para Linux que sólo es posible instalarlas desde el código fuente.

A pesar de lo que se lee comúnmente en internet, compilar e instalar software desde una distribución fuente para Linux, no es tan difícil como puede sonar.

Procedimiento

El procedimiento para instalar software que viene en una distribución fuente en paquetes .tar.gz o .tar.bz2 no es siempre la misma, pero generalmente es como se muestra a continuación:

```
# tar xvzf package.tar.gz (or tar xvjf package.tar.bz2)
# cd package
# ./configure
# make
# make install
```

Con suerte, ejecutando estos simples comandos, se desempaca, configura, compila e instala un paquete de software, y sin siquiera saber qué se está haciendo. Sin embargo, lo mejor es mirar más profundamente el procedimiento de instalación y ver lo que significan estos pasos.

Paso1. Desempacando.

Tal vez se haya dado cuenta que el paquete que contiene el código fuente del programa tiene una extensión .tar.gz o .tar.bz2. Esto significa que el paquete es un archivo tar comprimido, también conocido como “tarball”. Cuando se hace un paquete, el código fuente y los demás archivos, necesitan ser apilados juntos dentro de un único archivo tar. Después de apilarlos todos en el archivo tar, el archivo fue comprimido con gzip, de aquí viene la extensión .tar.gz.

Algunas personas comprimen el archivo tar con bzip2 y no con gzip. En estos casos el paquete tiene una extensión .tar.bz2. Estos paquetes se instalan exactamente de la misma forma que un .tar.gz, pero se usa un comando distinto cuando se desempaca.

No importa dónde se ponen los “tarballs” que se descargan de internet, pero se sugiere crear un directorio especial para “tarballs” descargados. En este tutorial se asume que los “tarballs” son almacenados en un directorio llamado “dls” que se ha creado bajo el directorio home. Sin embargo el directorio “dls” es sólo un ejemplo. Los paquetes de software .tar.gz o .tar.bz2 se pueden poner en cualquier directorio que se desee. En este ejemplo se asume que su nombre de usuario es “me” y que se ha descargado un paquete llamado pkg.tar.gz en el directorio “dls” que se supone ya creado (/home/me/dls).

Para desempacar el “tarball” después de descargarlo, se corre este comando:

```
me@puter: ~/dls$ tar xvzf pkg.tar.gz
```

Como se puede ver, se usa el comando tar con las opciones apropiadas (xvzf) para desempacar el “tarball”. Si se tiene un archivo con extensión .tar.bz2, se debe indicar al tar que no es un archivo tar de gzipped. Esto se hace usando la opción j en vez de la z, de esta forma:

```
me@puter: ~/dls$ tar xvjf pkg.tar.bz2
```

Lo que pasa después de desempacar, depende del paquete, pero en la mayoría de los casos, se crea un directorio con el mismo nombre que el paquete fue creado. El directorio nuevo se crea bajo el directorio donde se está en este momento. Para estar seguro, se puede usar el comando ls:

```
me@puter: ~/dls$ ls
pkg pkg.tar.gz
me@puter: ~/dls$
```

En nuestro ejemplo, al desempacar pkg.tar.gz se creó un directorio con el nombre del paquete. Ahora se debe entrar en ese nuevo directorio:

```
me@puter: ~/dls$ cd pkg
me@puter: ~/dls/pkg$
```

Se recomienda leer cualquier documentación que se encuentre en este directorio, por ejemplo los archivos README y/o INSTALL son muy comunes. Lea antes de continuar.

Paso 2. Configuración

Después de que se ha cambiado al directorio del paquete, y leer el manual, es momento de configurar el paquete. Usualmente, pero no siempre (es por eso que se deben leer los archivos README y/o INSTALL) se hace corriendo el script de configure.

El script de configuración se hace con este comando:

```
me@puter: ~/dls/pkg$ ./configure
```

Cuando se corre el script de configuración, aún no se compila nada. “configure” sólo verifica el sistema y asigna valores para las variables dependientes de sistema. Esos valores son usados para generar un “makefile”. El “makefile” es usado para generar el archivo binario.

Cuando se corre el script de configuración, se ven muchos mensajes raros saliendo en la consola. Esto es normal y no hay de qué preocuparse. Si el script encuentra un error, anuncia la existencia del error y sale. Sin embargo, si todo sale como debería, el script no informa ningún error, sale y termina.

Si configure sale sin errors, es hora de avanzar al próximo paso.

Paso 3. Construcción

Es momento de construir el binario, que es el programa ejecutable, desde el código fuente. Esto se hace ejecutando el comando “make”:

```
me@puter: ~/dls/pkg$ make
```

Note que “make” necesita el Makefile para construir el programa. De otra forma no sabe qué hacer. Es por esto que el script “configure” debe terminar exitosamente, o generar el Makefile de alguna otra forma.

Cuando se corre make, nuevamente saldrán en la consola muchos mensajes extraños. Esto también es perfectamente normal y no hay nada de lo que se deba preocupar. Este paso puede tomar algo de tiempo, dependiendo del tamaño del programa y de la velocidad de su computador.

Si todo sale como debería, el ejecutable está terminado y listo para correr después de que make ha hecho su trabajo. Ahora el paso final es instalar el programa.

Paso 4. Instalación

Ahora es el momento de instalar el programa. Para hacer esto debe ser root. Si ha estado haciendo las cosas como un usuario normal, se puede volver root con el comando su. El password de root será preguntado y entonces estará listo para el paso final.

```
me@puter: ~/dls/pkg$ su
Password:
root@puter: /home/me/dls/pkg#
```

Ahora que está logeado como root, puede instalar el programa con el comando install:

```
root@puter: /home/me/dls/pkg# make install
```

Nuevamente, aparecerán algunos mensajes raros en la consola. Después de que se detiene el software ha terminado la instalación y está listo para ser usado.

Como en este ejemplo no se cambió el comportamiento del script configure, el programa fue instalado en el lugar por defecto. En muchos casos es /usr/local/bin. Si /usr/local/bin (o en el lugar donde el programa fue instalado) está dentro del PATH, el programa se puede correr solamente escribiendo su nombre.

Limpiando el desorden.

Durante la ejecución del script “make” son creados todo tipo de archivos que eran necesarios durante el proceso de construcción pero son inútiles ahora y solamente ocupan espacio en el disco. Por este motivo se recomienda hacer un “make clean”:

```
me@puter: ~/dls/pkg$ make clean
```

Sin embargo, asegúrese de mantener el archive “Makefile”. Es necesario para después desinstalar el programa y hacerlo de la forma más fácil.

Desinstalación.

Lo primero que se debe hacer es buscar el manual de instalación o desinstalación del programa que se desea eliminar.

Si no se borró el archivo Makefile, es posible remover el programa haciendo un “make uninstall”:

```
root@puter: /home/me/dls/pkg# make uninstall
```

Ver algún texto extraño en la pantalla, es una buena señal. Si make comienza a presentar errores (quejarse) es una mala señal, por lo que se deberán remover los archivos del programa manualmente.

Si sabe dónde fue instalado el programa, deberá borrar manualmente los archivos instalados o el directorio donde está el programa. Si no tiene idea dónde están todos los archivos, se debe leer el Makefile y mirar dónde quedaron instalados, y luego borrarlos.

Traducción del artículo original en <http://www.tuxfiles.org>