



Preparación para el examen LPI 101

Tema 102.3 Instalar programas desde los fuentes

Créditos y licencia de uso

Coordinación:

Kiefer Von Jammo (Kiefer) kiefer@khrooon.net

Traducción:

Manuel Guillán (xLekOx) lpi@xleko.org

Maquetación:

Manuel Guillán (xLekOx) lpi@xleko.org

Javier Pulido (jpulido) javier.pulido@wanadoo.es

Versión 1.1 (19-09-2004 13:00)

Distribuido por FreeUOC (www.freeuoc.org) bajo licencia: Attribution-NonCommercial-ShareAlike2.0 de commons creative



<http://creativecommons.org/licenses/by-nc-sa/2.0/>

ÍNDICE

Índice de contenido

Tema 102.3

Instalar programas desde los fuentes.....	1
Créditos y licencia de uso.....	2
ÍNDICE.....	3
Introducción.....	4
Instalando Software desde el código fuente.....	5
Obteniendo el código fuente.....	5
Descomprimiendo el tarball.....	6
Ejecutando el script de configuración.....	6
Haciendo cambios al fichero Makefile.....	7
Compilando el software.....	7
Instalando el software.....	7
Preguntas TEST.....	8
Respuestas TEST.....	9
Bibliografía y enlaces recomendados.....	10

Introducción

En este capítulo se verá como construir e instalar un programa desde sus ficheros fuentes, se verán algunas opciones en el proceso de compilado.

Los comandos que se verán en este tema son:

- gunzip
- gzip
- bzip2
- tar
- configure
- make

Este tema tiene un peso (importancia) de 5 de cara al examen final de la certificación LPI 101. El total de la suma de pesos de todos los temas es de 106.

Instalando Software desde el código fuente



Hace unos años instalar software desde el código fuente era de lo más habitual, con la aparición de nuevas distribuciones y los paquetes binarios se perdió un poco esta costumbre, distribuciones como Gentoo rescatan este modo de administración de paquetes, si bien puede ser empleado en cualquier otra distribución. Los beneficios de instalar los paquetes desde su código fuente son muy numerosos, mayor personalización en cada PC que se instale con unas determinadas características, portabilidad de sistemas y arquitecturas y un largo etcetera.

Porqué motivo bajarse el código fuente si se dispone de paquetes binarios que ahorran el tiempo de espera? Mucha gente lo hace porque le gusta bucear entre el código, otros porque son desarrolladores y les interesa revisar este código en busca de líneas extrañas, backdoors, agujeros de seguridad, fallos en la programación, etc. No toda la gente se para en estos detalles, bien por falta de tiempo, por desconocimiento o simplemente porque no le interesa, en la mayoría de los casos, el código fuente está disponible antes de que salgan los binarios del programa en si.

La gran mayoría de los proyectos con licencia libre, tienen disponible para descargar el código fuente así como las últimas versiones del CVS (trabajo en desarrollo), algunos de los sitios más famosos son <http://sourceforge.net> o www.freshmeat.net. Normalmente vienen empaquetados con la extensión tar.gz. Una vez descomprimido el fichero se procede a la instalación, normalmente vienen instrucciones en los ficheros REAME o INSTALL, algunos de los paquetes traen un Makefile preconfigurado, mientras que otros lo generan para cada sistema. El Makefile es un fichero de texto que le indica al compilador como ha de hacer la tarea, los pasos básicos para la instalación de un nuevo software son:

1. Obtener el código fuente
2. Descomprimirlo a un directorio temporal
3. Leer los ficheros REAME e INSTALL
4. Ejecutar el comando ./configure
5. Hacer cambios si se requieren al fichero Makefile
6. Compilar el programa con el comando make
7. Finalmente, instalar el programa con make install



Obteniendo el código fuente



Como se dijo anteriormente lo normal es que en la propia web del proyecto esté disponible el código fuente en diversos formatos (tar.gz, zip, tar, rar ...). En los cds de las distribuciones, los últimos cds (que no se usan en la instalación) traen el código fuente de todos, o la gran mayoría, de los paquetes que se incluyen con la distro, caso por ejemplo de la SUSE (cd's 6 y 7).

Descomprimiendo el tarball



Los ficheros fuentes que se bajan normalmente son tarballs, es decir, un conjunto de ficheros agrupados con la utilidad tar y comprimidos con gzip (extensión tar.gz) para descomprimirlos basta con ejecutar el comando:

```
gunzip fichero.tar.gz
```

... o bien:

```
gzip -d fichero.tar.gz
```

... lo que dará como resultado el fichero descomprimido fichero.tar. Para extraer los ficheros del tar, se ejecuta el comando:

```
tar xvf fichero.tar
```

... obteniendo todos los ficheros del tar en el directorio actual. Normalmente las distribuciones traen versiones del tar que ya soportan la descompresión, de modo que ejecutando un solo comando se descomprime y se extraen los ficheros del documento. La opción a añadir es la z, de modo que el comando quedaría:

```
tar zxvf fichero.tar.gz
```

Comando tar



El comando tar es uno de los más usados en todas las distribuciones, además de servir para realizar backups también es una forma excelente para la distribución de archivos ya que combina el empaquetado y la compresión en un solo archivo. Generalmente tienen extensión .tar (aunque no es obligatoria) si se comprimen pueden tener extensión .tar.gz o bien .tgz (man tar para más info)

Comando gzip



gzip se usa tanto para comprimir como para descomprimir, esta utilidad conserva los permisos y la hora de creación de los ficheros, normalmente tienen la extensión .gz (man gzip para más info)



Comando gunzip

gunzip se usa para la descompresión de los ficheros comprimidos con gzip (man gunzip más info)



Comando bzip2

bzip2 es otro modo de compresión, su extensión más común es .bz2 (man bzip2 más info)



Comando bunzip2

Esta orden se usa para descomprimir ficheros creados con la utilidad bzip2. (man bunzip2 más info)

Ejecutando el script de configuración



El script ./configure automatiza la tarea de creación del fichero Makefile, chequeando problemas con las dependencias de compiladores y componentes. Si se quiere más información mientras se ejecuta el configure, se puede usar la opción debug: ./configure -enable-debug

Haciendo cambios al fichero Makefile



Hoy día gracias al script configure es muy raro tener que realizar cambios en el fichero Makefile resultante, no obstante, bajo algunas circunstancias, puede ser recomendable o necesario realizar algunos cambios: modificar directorios de destino, paths de los programas, etc.

Compilando el software



El siguiente paso es la compilación del software, basta con ejecutar el comando make una vez creado el Makefile, el resultado será el programa compilado y listo para instalar.

Instalando el software



Como última tarea nos queda instalar el software recién compilado, esto se hace con el comando make install, el cual instalará todos los archivos, las páginas del manual, etc. Normalmente el software viene acompañado de un fichero README donde se indica el nombre del ejecutable y en que directorios se se instalaron los ficheros del programa, las distribuciones actuales ya generan iconos o accesos directos en los menús una vez instalados.

Preguntas TEST

1. Que comando genera el fichero Makefile para tu sistema ?
 - A. ./gen
 - B. ./genmake
 - C. ./configure
 - D. ./config

2. Que comando instala software compilado ?
 - A. make
 - B. ./install
 - C. make setup
 - D. make install

Respuestas TEST

1. C
2. D

Bibliografía y enlaces recomendados

LPIC 1 Certification Bible (Bible) by Angie Nash, Jason Nash
John Wiley & Sons; Bk&CD-Rom edition (July 1, 2001) ISBN: 0764547720

LPI Linux Certification in a Nutshell by Jeffrey Dean
O'Reilly & Associates; 1st ed edition (May 15, 2001) ISBN: 1565927486

CramSession's LPI General Linux Part 1 : Certification Study Guide
CramSession.com; ISBN: B000079Y0V; (August 17, 2000)

Referencias Unix Reviews
<http://www.unixreview.com/documents/s=7459/uni1038932969999/>

Página LPI: www.lpi.org

Apuntes IBM: <http://www-106.ibm.com/developerworks/edu/l-dw-linux-lpir21-i.html>

Manuales GPL: <http://www.nongnu.org/lpi-manuals/>