



por Mario M. Knopf  
([homepage](#))

*Sobre el autor:*

Mario disfruta trabajando con Linux, redes y asuntos relacionados con la seguridad.

*Traducido al español por:*  
guadapeich ([homepage](#))

## CheckInstall

```
[root@deimos ~]# /usr/local/sbin/checkinstall -h
checkinstall 1.6.0beta4, Copyright 2002 Felipe Eduardo Sanchez Diaz Duran
This software is released under the GNU GPL.

This software is released under the GNU GPL.
Usage: checkinstall [options] [command [command arguments]]
Options:
*Package type selection*
-t, --type=<slackware|rpm|debian> Choose packaging system
-S                               Build a Slackware package
```

*Resumen:*

*Checkinstall* es una aplicación para construir automáticamente paquetes RPM/Debian o Slackware desde paquetes fuente tar.gz. Esto permite realizar una instalación y desinstalación "limpia" de la mayoría de los paquetes de código fuente tar.gz.

## Introducción

Frecuentemente sucede que el programa que deseas probar está sólo disponible como código fuente tar.gz (no hay paquete disponible rpm o Debian). En ese caso bajas el paquete fuente, lo desempaquetas y lo compilas manualmente. So far, so good. Sin embargo, ¿qué sucede si quieres borrar el programa?.

El *Makefile* en pocas ocasiones contiene una rutina apropiada para desinstalar el programa. De echo hay la posibilidad de instalar el programa en un directorio temporal y copiar aparte todos los archivos que fueron creados o modificados por la instalación para una posterior desinstalación. Pero este procedimiento es muy largo y penoso si los programas son frecuentemente compilados desde las fuentes. La herramienta *CheckInstall* [1] escrita por Felipe Eduardo Sánchez Díaz Durán resuelve este problema.

Como regla, compilas e instalas un programa compatible con *GNU Autoconf* con la conocida secuencia de órdenes

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

El script *configure* intenta suponer los valores correctos para distintas variables dependientes del sistema, los cuales son usados después durante la compilación. Comprueba si todos los requisitos para la compilación se cumplen y usa esos valores para crear un *Makefile* en cada directorio del paquete. Además el script *configure* genera archivos adicionales. Resumiendo estos son:

- uno o más *Makefile(s)* en cada directorio/subdirectorio
- un script denominado *config.status*
- un archivo de texto *config.log*

- otro script normalmente denominado *config.cache* (opcional)
- varios archivos cabeceras de C (\*.h) con definiciones para sistemas específicos (opcional)

Después de que el script *configure* ha terminado correctamente, tecleas *make* para compilar el paquete. Esto genera los binarios ejecutables. Existe la posibilidad de correr inmediatamente después de *make* algún test con *make check*. Pero este es un paso opcional, porque el paquete debe estar configurado para hacerlo. Si *make* ha realizado su trabajo, puedes instalar el programa compilado con la orden *make install* – por razones obvias, necesitas ser un usuario con "privilegios" para este paso. Después de que el programa fue instalado con éxito, puedes borrar el programa binario y los archivos objeto desde el directorios del código fuente tecleando *make clean*. Si prefieres borrar también los archivos creados por *configure*, entonces teclea *make distclean*. Sin embargo, los dos últimos pasos citados son, como *make check*, opcionales y normalmente usados por el desarrollador durante el desarrollo y prueba del paquete. Pueden ser utilizados por el usuario para liberar espacio de disco o para mantener limpia la estructura de directorios. Además *make distclean* hace posible compilar el paquete para un tipo de ordenador distinto.

Información detallada sobre *GNU Autoconf* puede encontrarse en el manual en línea en [2]. Además de una introducción básica, puedes aprender más sobre *GNU Build System*, haciendo tus propios scripts *configure*, programando en *M4* y creando tus propios macros, scripts ejecutables, etcetera.

## CheckInstall

Como se ha mencionado, la secuencia de órdenes para hacer un programa compatible *GNU Autoconf* desde sus fuentes es:

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

En este punto es el momento de usar *CheckInstall*. Este sustituye *make install* con su propia orden *checkinstall*. Las otras dos instrucciones permanecen sin ser afectadas y son usadas normalmente como antes. Así la nueva secuencia de órdenes con *CheckInstall* ahora es:

```
./configure && make && checkinstall
```

Sin embargo, la orden *checkinstall* no obstante comienza *make install* por defecto y sigue todas las escrituras que son hechos por la instalación. Para este propósito *CheckInstall* usa el programa *Installwatch* [3], que fué originalmente escrito por Pancrazio de Mauro. Después de que *make install* fué completado con éxito, *CheckInstall* genera un paquete Slackware–, Debian– o RPM y lo instala con el gestor de paquetes por defecto de la distribución y deja una copia del paquete en el directorio fuente actual o en el directorio normal de almacenamiento. Además, es posible cambiar el directorio normal de almacenamiento con la variable *PAK\_DIR* dentro del archivo de configuración. La copia guardada puede ser instalada, de echo bajo consideración de posibles dependencias de paquetes, en otras máquinas en la red sin compilar el paquete fuente cada vez.

*CheckInstall* no sólo hace uso de *make install*, también coopera con otras instrucciones de instalación. Si el script de instalación es por ejemplo: *setup.sh* la orden es:

```
./configure && make && checkinstall setup.sh
```

También es posible comenzar *CheckInstall* con varias opciones. La siguiente orden muestra una completa lista de todas las opciones disponibles que son divididas en las secciones *Install options*, *Scripting options*, *Info display options*, *Package tuning options*, *Cleanup options* and *About CheckInstall*:

```
# checkinstall -h
```

Si *CheckInstall* es iniciado con una de esas opciones, sobrescribirá los valores incluidos en el archivo de configuración *checkinstallrc*.

Pero también *CheckInstall* tiene limitaciones. No puede manejar programas enlazados estáticamente, porque *Installwatch* no es capaz de ver los archivos modificados durante el proceso de instalación. En general hay dos tipos de librerías: librerías enlazadas estática o dinámicamente. Estas librerías están integradas en el programa por una directiva *include*. Programas enlazados estáticamente ya tienen todas las funciones de las librerías necesarias y no deben cargarlas en la RAM al correr. Además son independientes de las librerías actualmente instaladas en el sistema destino, porque un *Linker* tiene incluido la librería dentro del programa cuando se compila.

## Instalación.

*CheckInstall* es ya desde hace bastante tiempo parte de la lista de programas de las grandes distribuciones y puede ser instalado con su respectivo sistema manejador de paquetes. Si este no es el caso, tu puedes bajar tar-balls satisfactorios o paquetes fuente para varias distribuciones desde la página del proyecto en [4].

La instalación de *CheckInstall* es bastante simple y corre en pocos pasos – pero para una instalación afortunada de *CheckInstall* tu necesitarás paradójicamente *CheckInstall*. Después de la orden obligatoria *make install* escribe *checkinstall*, lo que crea un paquete binario apropiado desde el programa compilado. Ahora tu puedes instalar este paquete con tu gestor de paquetes y también es posible desinstalar el paquete limpiamente. Pero antes de que *CheckInstall* cree el paquete, debes constestar una pregunta sobre el gestor de paquetes usado y comprobar la información suministrada para su corrección. Esto aparece en la cabecera del paquete después.

El procedimiento para instalar la nueva versión de prueba *checkinstall-1.6.0beta4.tgz* se muestra en el siguiente bloque de código. Instalará *CheckInstall*, *Installwatch* y *makepak*, una versión modificada de *makepkg*. Si estás interesado en los cambios de la nueva version, mira en *Release Notes* [5] y/o en el *Changelog* [6].

```
$ tar xzf checkinstall-1.6.0beta4.tgz
$ cd checkinstall-1.6.0beta4
checkinstall-1.6.0beta4 $ make
[...]
checkinstall-1.6.0beta4 $ su
Password:
checkinstall-1.6.0beta4 # make install
[...]
checkinstall-1.6.0beta4 # checkinstall

checkinstall 1.6.0beta4, Copyright 2002 Felipe Eduardo Sanchez Diaz Duran
This software is released under the GNU GPL.

Please choose the packaging method you want to use.
Slackware [S], RPM [R] or Debian [D]? R

*****
*** RPM package creation selected ***
*****
```

This package will be built according to these values:

```
1 - Summary: [ CheckInstall installations tracker, version 1.6.0beta4 ]
2 - Name: [ checkinstall ]
3 - Version: [ 1.6.0beta4 ]
4 - Release: [ 1 ]
5 - License: [ GPL ]
6 - Group: [ Applications/System ]
7 - Architecture: [ i386 ]
8 - Source location: [ checkinstall-1.6.0beta4 ]
9 - Alternate source location: [ ]
10 - Provides: [ checkinstall ]
11 - Requires: [ ]
```

Enter a number to change any of them or press ENTER to continue:

Installing with make install...

```
===== Installation results =====
[...]
```

```
===== Installation successful =====
```

Copying documentation directory...

```
./
./NLS_SUPPORT
./README
./FAQ
./TODO
./CREDITS
./INSTALL
./Changelog
./BUGS
./installwatch-0.7.0beta4/
./installwatch-0.7.0beta4/README
./installwatch-0.7.0beta4/TODO
./installwatch-0.7.0beta4/VERSION
./installwatch-0.7.0beta4/INSTALL
./installwatch-0.7.0beta4/CHANGELOG
./installwatch-0.7.0beta4/BUGS
./installwatch-0.7.0beta4/COPYING
./RELNOTES
./COPYING
```

Copying files to the temporary directory...OK

Striping ELF binaries and libraries...OK

Compressing man pages...OK

Building file list...OK

Building RPM package...OK

NOTE: The package will not be installed

Erasing temporary files...OK

Writing backup package...OK

Deleting temp dir...OK

```
*****
```

Done. The new package has been saved to

```
/usr/src/redhat/RPMS/i386/checkinstall-1.6.0beta4-1.i386.rpm  
You can install it in your system anytime using:
```

```
rpm -i checkinstall-1.6.0beta4-1.i386.rpm
```

\*\*\*\*\*

```
checkinstall-1.6.0beta4 # cd /usr/src/redhat/RPMS/i386/  
i386 # rpm -i checkinstall-1.6.0beta4-1.i386.rpm  
i386 #
```

Los usuarios de distribuciones basadas en Debian pueden instalar el paquete con *dpkg -i*. Los usuarios de Slackware pueden usar *installpkg* para este propósito.

Pero usando la función de consulta de tu gestor de paquetes, en este caso RPM, tu puedes comprobar si el paquete fué limpiamente integrado en su base de datos y ver los campos de información adicional de la cabecera del paquete.

```
$ rpm -qi checkinstall  
Name           : checkinstall           Relocations: (not relocatable)  
Version        : 1.6.0beta4           Vendor      : (none)  
Release        : 1                    Build Date  : Mo 06 Dez 2004 17:05:45 CET  
Install Date: Di 07 Dez 2004 01:41:49 Build Host  : deimos.neo5k.lan  
Group          : Applications/System    Source RPM  : checkinstall-1.6.0beta4-1.src.rpm  
Size           : 264621                License    : GPL  
Signature      : (none)  
Packager       : checkinstall-1.6.0beta4  
Summary        : CheckInstall installations tracker, version 1.6.0beta4  
Description    :  
CheckInstall installations tracker, version 1.6.0beta4
```

```
CheckInstall keeps track of all the files created or  
modified by your installation script ("make install"  
"make install_modules", "setup", etc), builds a  
standard binary package and installs it in your  
system giving you the ability to uninstall it with your  
distribution's standard package management utilities.
```

## Configuración

Puedes modificar el bien comentado fichero de texto */usr/lib/local/checkinstall/checkinstallrc* para cambiar la configuración que *CheckInstall* trae por defecto.

Como *CheckInstall* te pregunta en cada comienzo que tipo de paquete debe crear, es sabio añadir este valor permanentemente editando *INSTYPE*. Es también una buena idea mirar las variables *INSTALL*, *PAK\_DIR* y *RPM\_FLAGS* o alternativamente *DPKG\_FLAGS*. Con las dos últimas variables puedes definir algunas banderas de instalación opcionales, y modificando *PAK\_DIR* puedes especificar otro directorio para la copia del paquete. *INSTALL* te permite decidir entre sólo crear el paquete y/o instalarlo.

```
$ cat /usr/lib/local/checkinstall/checkinstallrc
```

```
#####  
# CheckInstall configuration file #  
#####
```

```

#####
# These are default settings for CheckInstall, modify them as you #
# need. Remember that command line switches will override them. #
#####

# Debug level
# 0: No debug
# 1: Keep all temp files except the package's files
# 2: Keep the package's files too

DEBUG=0

# Location of the "installwatch" program
INSTALLWATCH_PREFIX="/usr/local"
INSTALLWATCH=${INSTALLWATCH_PREFIX}/bin/installwatch

# Location of the makepkg program. "makepak" is the default, and is
# included with checkinstall. If you want to use Slackware's native "makepkg"
# then set this to "makepkg"

MAKEPKG=/sbin/makepkg

# makepkg optional flags. These are recommended if running a newer Slackware
# version: "-l y -c n"

MAKEPKG_FLAGS="-l y -c n"

# Is MAKEPKG running interactively? If so, you might want
# to see what it's doing:

SHOW_MAKEPKG=0

# Where will we keep our temp files?
BASE_TMP_DIR=/var/tmp ## Don't set this to /tmp or / !!

# Where to place the installed document files
DOC_DIR=""

# Default architecture type (Leave it empty to allow auto-guessing)
ARCHITECTURE=""

# Default package type. Leave it empty to enable asking everytime
# S : Slackware
# R : RPM
# D : Debian

INSTYPE="R"

# Storage directory for newly created packages
# By default they will be stored at the default
# location defined for the package type

PAK_DIR=""

# RPM optional flags
RPM_FLAGS="--force --nodeps --replacepkgs "

# dpkg optional flags
DPKG_FLAGS=""

## These are boolean. Set them to 1 or 0

# Interactively show the results of the install command (i.e. "make install")?

```

```

# This is useful for interactive installation commands
SHOW_INSTALL=1

# Show Slackware package installation script while it runs? Again, useful if
# it's an interactive script
SHOW_SLACK_INSTALL=0

# Automatic deletion of "doc-pak" upon termination?
DEL_DOCPAK=1

# Automatic deletion of the spec file?
DEL_SPEC=1

# Automatic deletion of "description-pak"?
DEL_DESC=1

# Automatically strip all ELF binaries?
STRIP_ELF=1

# Automatically strip all ELF shared libraries?
# Note: this setting will automatically be set to "0" if STRIP_ELF=0
STRIP_SO_ELF=1

# Automatically search for shared libraries and add them to /etc/ld.so.conf?
# This is experimental and could mess up your dynamic loader configuration.
# Use it only if you know what you are doing.
ADD_SO=0

# Automatically compress all man pages?
COMPRESS_MAN=1

# Set the umask to this value
CKUMASK=0022

# Backup files overwritten or modified by your install command?
BACKUP=1

# Write a doinst.sh file that installs your description (Slackware)?
AUTODOINST=1

# Are we going to use filesystem translation?
TRANSLATE=1

# Reset the owner/group of all files to root.root?
RESET_UIDS=0

# Use the new (8.1+) Slackware description file format?
NEW_SLACK=1

# Comma delimited list of files/directories to be ignored
EXCLUDE=""

# Accept default values for all questions?
ACCEPT_DEFAULT=0

# Use "-U" flag in rpm by default when installing a rpm package
# This tells rpm to (U)pdate the package instead of (i)nstalling it.
RPM_IU=U

# Inspect the file list before creating the package
CK_INSPECT=0

# Review the .spec file before creating a .rpm
REVIEW_SPEC=0

```

```
# Review the control file before creating a .deb
REVIEW_CONTROL=0

# Install the package or just create it?
INSTALL=0
```

## Conclusión

*CheckInstall* es una brillante herramienta que puede hacer la administración de un equipo Linux un poco más fácil. Particularmente si lo programas deben ser frecuentemente compilados desde las fuentes, *CheckInstall* te da la posibilidad de desinstalar limpiamente el programa sin el riesgo de un sistema inestable. Además puedes instalar estos paquetes también en otra máquinas sin tener que compilar el programa cada vez de nuevo – de echo debes considerar posibles dependencias de paquetes. Sin embargo, esto normalmente no es un gran problema para máquinas idénticas.

## Enlaces

- [1] <http://asic-linux.com.mx/~izto/checkinstall/> [Home of CheckInstall]
- [2] <http://www.gnu.org/software/autoconf/manual/autoconf-2.57/autoconf.html> [GNU Autoconf Online Manual]
- [3] <http://asic-linux.com.mx/~izto/checkinstall/installwatch.html> [Installwatch]
- [4] <http://asic-linux.com.mx/~izto/checkinstall/download.php> [CheckInstall Downloads]
- [5] <http://asic-linux.com.mx/~izto/checkinstall/docs/RELNOTES> [Release Notes]
- [6] <http://asic-linux.com.mx/~izto/checkinstall/docs/Changelog> [Changelog]

---

<p><u>Contactar con el equipo de LinuFocus</u> © Mario M. Knopf "some rights reserved" see <a href="http://linuxfocus.org/license/">linuxfocus.org/license/</a> <a href="http://www.LinuxFocus.org">http://www.LinuxFocus.org</a></p>	<p>Información sobre la traducción: de --&gt; -- : Mario M. Knopf (<a href="#">homepage</a>) de --&gt; en: Mario M. Knopf (<a href="#">homepage</a>) en --&gt; es: guadapeich (<a href="#">homepage</a>)</p>
---	--