



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

*Over de auteur:*

Mario heeft plezier aan het werken met Linux, netwerken en beveiliging-gerelateerde onderwerpen.

*Vertaald naar het Nederlands door:*  
Guus Snijders  
<[ghs\(at\)linuxfocus.org](mailto:ghs(at)linuxfocus.org)>

## 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
```

*Kort:*

*Checkinstall* is een utility om automatisch RPM/Debian of Slackware pakketten te maken van tar.gz bron-archieven. Dit maakt een "schone" installatie en de-installatie van bijna ieder tar.gz broncode archief mogelijk.

## Introductie

Het komt vaak voor dat een programma dat je graag wil testen, alleen beschikbaar is als broncode in een tar.gz archief (en dus geen rpm of Debian pakket beschikbaar heeft). In een dergelijk geval downloadt je de broncode, pak deze uit en compileer je het handmatig. Tot zover geen probleem. Echter, wat als je het programma wilt verwijderen?

De *Makefile* bevat slechts in enkele gevallen de juiste routine voor het verwijderen van het programma. Natuurlijk kun je het programma in een tijdelijke directory installeren en een lijst maken van alle nieuwe of gewijzigde bestanden, zodat deze later verwijderd kunnen worden. Dit is echter een nogal pijnlijk procedure en erg arbeids-intensief als je vaak programma's compileert. De tool *CheckInstall* [1], van Felipe Eduardo Díaz Durán, biedt een oplossing voor dit probleem.

Als vuistregel kun je een *GNU Autoconf*-compliant programma compileren en installeren met de overbekende commando's

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

Het *configure* shell-script probeert de juiste waarden te raden voor verschillende systeem-afhankelijke variabelen, die later worden gebruikt tijdens de compilatie. Het controleert of aan alle voorwaarden voor compilatie is voldaan en gebruikt die waarden om een *Makefile* in iedere directory van het pakket te maken. Verder genereert het *configure* script nog enkele bestanden. In het kort zijn dit:

- een of meerdere *Makefile(s)* in iedere (sub)directory

- een shell script genaamd *config.status*
- een tekstbestand *config.log*
- een ander shell script, meestal *config.cache* genaamd (optioneel)
- verschillende C header bestanden (\*.h) met systeem-specifieke definities (optioneel)

Als het *configure* script zonder fouten beëindigd is, type je *make* om het pakket te compileren. Dit genereert de uitvoerbare bestanden. Het is mogelijk om na *make* een aantal zelf-tests uit te voeren met *make check*, maar dit is een optionele stap omdat het pakket dit proces ook moet ondersteunen. Als *make* zijn werk goed heeft gedaan, kun je het programma installeren met *make install* – om voor de hand liggende redenen moet je voor deze stap root zijn. Als het programma met succes geïnstalleerd is, kun je de programma binaries en object bestanden uit de brondirectory verwijderen met *make clean*. Eventueel kun je ook de bestanden die door *configure* zijn gemaakt verwijderen met *make distclean*. De 2 laatstgenoemde stappen zijn, net als *make check* optioneel en worden meestal door de ontwikkelaar gebruikt tijdens het ontwikkelen en testen. Ze kunnen ook door de gebruiker worden gebruikt om schijfruimte te besparen of om de directory-structuur schoon te houden. Verder maakt *make distclean* het mogelijk om het pakket voor een ander soort computer te compileren.

Gedetailleerde informatie over *GNU Autoconf* is te vinden in de Online-Manual op [2]. Naast een basis introductie, kun je er leren over het *GNU Build System*, over het maken van je eigen *configure* scripts, programmeren in *M4* en het maken van je eigen macros, portable shell programmeren, enzovoort.

## CheckInstall

Zoals eerder genoemd, de serie commando's om een *GNU Autoconf* compliant programma vanuit de broncode te bouwen is:

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

Op dit punt wordt het tijd om *CheckInstall* te introduceren. Deze vervangt *make install* met zijn eigen commando *checkinstall*. De twee andere instructies blijven hetzelfde en worden net als eerder gebruikt. De nieuwe commando serie met *CheckInstall* is dus:

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

De instructie *checkinstall* begint standaard echter met *make install* en monitort alle schrijfacties die tijdens de installatie plaatsvinden. Hiervoor gebruikt *Checkinstall* het programma *Installwatch*[3], van de auteur Pancrazio de Mauro. Als *make install* met succes werd beëindigd, genereert *CheckInstall* een Slackware-, Debian- of RPM-pakket en laat het een kopie van het pakket achter in de standaard opslag-directory. Verder is het mogelijk om de standaard opslag-directory te veranderen met de variabele *PAK\_DIR* in het configuratie bestand. De geadresseerde kopie kan gebruikt worden om het pakket op andere machines te installeren, met inachtneming van mogelijke pakket afhankelijkheden en zonder iedere keer weer de broncode te compileren.

*CheckInstall* gebruikt niet alleen *make install*, maar kan ook overweg met andere installatie instructies. Als het installatie script bijvoorbeeld *setup.sh* is, wordt de instructie:

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

Verder is er de mogelijkheid om *CheckInstall* met verschillende opties te starten. Het volgende commando geeft een overzicht van alle beschikbare opties, welke onderverdeeld in de secties *Install options*, *Scripting options*, *Info display options*, *Package tuning options*, *Cleanup options* en *About CheckInstall*:

```
# checkinstall -h
```

Als *CheckInstall* wordt gestart met een of meer van de opties, zal het dezelfde opties in het configuratie bestand *checkinstallrc* negeren.

Maar *CheckInstall* heeft limieten. Het kan niet overweg met statisch gelinkte programma's, omdat *Installwatch* de gewijzigde bestanden tijdens de installatie niet kan monitoren. Globaal zijn er twee soorten programma libraries: statisch en dynamisch gelinkte libraries. Deze libraries zijn geïntegreerd in een programma met een *include*-directive. Statisch-gelinkte programma's beschikken zelf over alle nodige library functies en dus hoeven die tijdens het starten niet meer in het RAM worden geladen. Verder zijn ze onafhankelijk van de libraries die zijn geïnstalleerd op het doel-systeem, doordat de zogenoemde *Linker* de library in het uitvoerbare bestand heeft opgenomen tijdens het compileren.

## Installatie

*CheckInstall* is al langere tijd onderdeel van de software pools van de grotere distributies en kan geïnstalleerd worden met de respectievelijke pakket-management systemen. Als dit niet het geval is met jouw distributie, kun je de nodige tar-balls of voorgebouwde pakketten voor de verschillende distributies downloaden vanaf de project-website op [4].

De installatie van *CheckInstall* is vrij simpel en werkt in een paar stappen – maar voor een succesvolle installatie van *CheckInstall* heb je paradoxaal genoeg *CheckInstall* nodig. Na de (voor de hand liggende) instructie *make install*, type je *checkinstall*, om een binair pakket te bouwen van het gecompileerde programma. Nu kun je dit pakket installeren met je package-manager en is het mogelijk om het programma veilig te verwijderen. Maar voordat *CheckInstall* het pakket bouwt, moet je vragen beantwoorden over de gebruikte package manager en de juistheid van de informatie velden controleren. Die verschijnen later in de package header.

De procedure om de nieuwe beta release *checkinstall-1.6.0beta4.tgz* te installeren, is gedemonstreerd in de volgende code-box. Dit zal *CheckInstall*, *Installwatch* en *makepak* – een aangepaste versie van *makepkg* – installeren. Als je geïnteresseerd bent in de veranderingen van de nieuwe versie, zie de *Release Notes* [5] en/of het *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 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 #
```

Gebruikers van Debian-gebaseerde distributies kunnen het pakket met *dpkg -i* installeren. Slackware gebruikers kunnen hiervoor *installpkg* gebruiken.

Door de query functie van je pakket manager te gebruiken, in dit geval RPM, kun je controleren of het pakket goed is opgenomen in diens database en de additionele informatie velden in de pakket header zien.

```
$ 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.
```

## Configuratie

Je kunt het goed-becommentarieerde tekst bestand */usr/lib/local/checkinstall/checkinstallrc* aanpassen om het standaard gedrag van *CheckInstall* te veranderen.

Daar *CheckInstall* je bij iedere start vraagt welk type pakket moet worden geproduceerd, is het handig om deze waarde permanent op te geven door *INSTYPE* in te stellen. Ook is het een goed idee om te kijken naar de variabelen *INSTALL*, *PAK\_DIR* en *RPM\_FLAGS* en eventueel *DPKG\_FLAGS*. Met de laatste twee variabelen kun je enkele optionele installatie flags opgeven, door *PAK\_DIR* aan te passen kun je een andere opslag directory opgeven voor de pakket kopie. *INSTALL* geeft je de keuze tussen alleen het genereren van het pakket en/of deze te installeren.

```
$ 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
```

## Conclusie

*CheckInstall* is een briljante tool die het beheer van een Linux machine een stuk gemakkelijker kan maken. Vooral als er regelmatig programma's vanuit de broncode moeten worden gecompileerd. *CheckInstall* biedt je de mogelijkheid om de programma's netjes te verwijderen, zonder het risico van een inconsistent systeem. Verder kun je deze pakketten ook op andere machines installeren zonder het programma iedere keer weer te compileren – natuurlijk moet je wel eventuele pakket dependencies (afhankelijkheden) controleren. Dit is echter niet zo'n heel groot probleem voor identieke machines.

## Links

- [1] <http://asic-linux.com.mx/~izto/checkinstall/> [Thuis van 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>Site onderhouden door het LinuxFocus editors team</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>Vertaling info: de --&gt; -- : Mario M. Knopf (<a href="#">homepage</a>) de --&gt; en: Mario M. Knopf (<a href="#">homepage</a>) en --&gt; nl: Guus Snijders &lt;<a href="mailto:ghs(at)linuxfocus.org">ghs(at)linuxfocus.org</a>&gt;</p>
--	--

2005-05-24, generated by lfparsr\_pdf version 2.51