



von Mario M. Knopf
[\(homepage\)](#)

Über den Autor:

Mario beschäftigt sich leidenschaftlich gerne mit Linux, Netzwerken und sicherheitsrelevanten Themen.

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
```

Zusammenfassung:

Dieser Artikel behandelt die saubere Installation/Deinstallation von manuell aus dem Quellcode kompilierten Programmen unter Benutzung des Tools *CheckInstall*.

Einleitung

Oft kommt es vor, daß das ein oder andere Programm, welches man unbedingt testen möchte, nur im Quellcode vorliegt (kein rpm und kein src.rpm). In einem solchen Fall lädt man sich das passende Archiv herunter, entpackt es und übersetzt es anschließend manuell. So weit, so gut. Was aber geschieht, wenn man das Programm wieder deinstallieren möchte?

Eine adäquate Routine zur sauberen Deinstallation des Pakets ist nur in den seltensten Fällen im *Makefile* enthalten. Natürlich bestünde noch die Möglichkeit, das Programm in ein temporäres Verzeichnis zu installieren und sich die neu erstellten Dateien und Verzeichnisse für ein späteres Entfernen zu notieren. Diese Vorgehensweise ist allerdings mühsam und aufwendig, wenn häufig Programme aus den Quellen kompiliert werden. Abhilfe schafft das von Felipe Eduardo Sánchez Díaz Durán geschriebene Tool *CheckInstall* [1].

Im Regelfall wird ein *GNU Autoconf* konformes Programm unter Verwendung der allseits bekannten Befehlsfolge `./configure && make && make install` kompiliert und installiert. Dabei versucht das *configure*-Shellskript geeignete Werte für diverse systemabhängige Variablen abzuschätzen, die dann während der späteren Kompilierung verwendet werden und prüft somit, ob alle diesbezüglichen Voraussetzungen für einen erfolgreichen Durchlauf erfüllt werden. Jene Werte werden auch zur Erstellung eines oder mehrerer *Makefiles* benutzt. Weiterhin erzeugt das *configure*-Skript zusätzliche Dateien. Zusammenfassend lauten diese:

- ein oder mehrere *Makefile(s)* in jedem Verzeichnis respektive Unterverzeichnis
- ein Shellskript namens *config.status*
- eine Textdatei *config.log*
- ein weiteres Shellskript *config.cache* (optional)
- verschiedene C-Header-Dateien (*.h) mit systemspezifischen Definitionen (optional)

Nachdem das *configure*-Skript erfolgreich beendet wurde, wird der Befehl *make* aufgerufen. Dieser stößt die eigentliche Kompilierung an und erzeugt schließlich die entsprechenden Binaries, also ausführbaren Programmcode. Unter Umständen besteht unmittelbar nach dem Aufruf von *make* die Möglichkeit, mittels *make check* noch diverse Selbsttests zu fahren. Dies ist allerdings ein optionaler Schritt, da das Paket diesen Vorgang auch hinreichend unterstützen muss. Wenn *make* seine Arbeit erfolgreich verrichtet und die jeweiligen Binaries erstellt hat, kann das kompilierte Programm durch den Befehl *make install* ins System eingebunden werden – dazu sind aus naheliegenden Gründen jedoch privilegierte Benutzerrechte erforderlich. Da das Programm nun erfolgreich installiert wurde, können die durch die Kompilierung erzeugten Binaries durch den Befehl *make clean* bedenkenlos aus dem Quellverzeichnis gelöscht werden. Um auch die vom *configure*-Skript erstellten Dateien zu löschen, bedarf es eines Aufrufs von *make distclean*. Die beiden letztgenannten Schritte sind aber, ebenso wie der Aufruf von *make check*, optional und werden in der Regel nur dazu benutzt, um Festplattenkapazität zu sparen und die Verzeichnisstruktur übersichtlich zu halten. Ferner ermöglicht der Aufruf von *make distclean* die Kompilierung des Programms für andere Systeme.

Weiterführende Informationen zu *GNU Autoconf* lassen sich im Online-Manual unter [2] finden. Neben einer grundlegenden Einführung erfährt man mehr über das *GNU Build System*, die Erstellung von eigenen *configure*-Skripten, zur Programmierung in *M4* und der Erstellung eigener Makros, portabler Shell-Programmierung et cetera.

CheckInstall

Wie eingangs bereits erwähnt, lautet die Befehlsfolge zur manuellen Übersetzung und Installation eines *GNU Autoconf*-konformen Programms aus dem Quellcode folgendermaßen:

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

An dieser Stelle kommt nun *CheckInstall* ins Spiel und ersetzt *make install* durch den Befehl *checkinstall*. Die beiden anderen Kommandos bleiben davon unberührt und werden, ihrem Zweck entsprechend, weiterhin wie gewohnt genutzt. Die Befehlsfolge unter Verwendung des Tools *CheckInstall* lautet also wie folgt:

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

Das Kommando *checkinstall* startet aber per Default trotzdem *make install* und beobachtet daraufhin sämtliche Schreibvorgänge, die durch die Installation getätigt werden. Zu diesem Zweck verwendet *CheckInstall* das Programm *Installwatch* [3], welches ursprünglich von Pancrazio de Mauro geschrieben wurde. Nachdem *make install* seinen Durchlauf erfolgreich beendet hat, generiert *CheckInstall* wahlweise ein Slackware-, Debian- oder RPM-Paket, installiert dieses über den jeweils eingesetzten Paketmanager und hinterläßt eine Kopie des Pakets im aktuellen Quellverzeichnis oder am standardmäßigen Speicherort der benutzten Distribution. Zudem ist es auch möglich, den Speicherort mittels der Variable *PAK_DIR* innerhalb der Konfigurationsdatei selbst zu bestimmen. Die besagte Kopie läßt sich dann in der Regel auch, unter Berücksichtigung etwaiger Paketabhängigkeiten, auf anderen Maschinen im Netzwerk installieren, ohne jedesmal aufs Neue kompilieren zu müssen.

CheckInstall macht jedoch nicht zwingend von *make install* Gebrauch, sondern arbeitet auch problemlos mit anderen Installationsbefehlen zusammen. Lautet das Installationskript beispielsweise *setup.sh* sieht der Befehlssatz so aus:

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

Des weiteren besteht die Möglichkeit, *CheckInstall* mit diversen Optionen zu starten. Das nachfolgende Kommando liefert eine komplette Übersicht aller verfügbarer Optionen, welche wiederum in die Sektionen *Install options*, *Scripting options*, *Info display options*, *Package tuning options*, *Cleanup options* und *About CheckInstall* untergliedert sind:

```
# checkinstall -h
```

Wird *CheckInstall* mit einer dieser Optionen aufgerufen, werden diese vorrangig zu den in der Konfigurationsdatei *checkinstallrc* definierten Werten behandelt.

Allerdings stößt *CheckInstall* bei Programmen, welche statisch gegen Bibliotheken gelinkt sind, an seine Grenzen, da es *Installwatch* dadurch nicht möglich ist, die durch die Installation modifizierten Dateien zu überwachen beziehungsweise aufzulisten und an *CheckInstall* zu übergeben. Grundsätzlich werden zwei Arten von Programmbibliotheken unterschieden: statisch oder dynamisch gelinkte Bibliotheken. Diese werden durch eine *include*-Anweisung in ein Programm eingebunden und können dann beliebig aufgerufen werden. Im Gegensatz zu dynamisch gelinkten Programmen besitzen statisch gelinkte bereits alle benötigten Bibliotheksfunktionen und müssen diese nicht mehr zur Laufzeit in den Arbeitsspeicher laden. Weiterhin sind sie unabhängig von den tatsächlich auf dem Zielsystem installierten Bibliotheken, da ein sogenannter *Linker* die Funktionen der Bibliothek beim Kompilervorgang mit dem ausführbaren Programm verbindet.

Installation

CheckInstall ist schon seit längerer Zeit fester Bestandteil des Softwarepools größerer Distributionen und kann über deren jeweiliges Paketmanagementsystem problemlos installiert werden. Sollte dies wider Erwarten einmal nicht so sein, stellt die Projektseite auf [4] geeignete Quellarchive und vorgefertigte Binär-Pakete für unterschiedliche Distributionen zum Download bereit.

Die Installation von *CheckInstall* gestaltet sich denkbar einfach und verläuft in wenigen Schritten – paradoxerweise ist für eine erfolgreiche Installation aber *CheckInstall* notwendig. Nach dem obligatorischen Aufruf von *make install* erfolgt die Eingabe des Befehls *checkinstall*, welcher aus dem soeben kompilierten Programm ein geeignetes Binär-Paket generiert, was sich wiederum ohne Probleme über das eingesetzte Paketmanagementsystem installieren und vor allem wieder sauber deinstallieren läßt. Allerdings gilt es – bevor das Paket durch *CheckInstall* erzeugt wird – noch eine Frage zum eingesetzten Paketmanager zu beantworten und die von *CheckInstall* bereits mit durchaus sinnvollen vorbelegten Informationsfelder auf Richtigkeit zu überprüfen. Jene erscheinen später auch im Header des Pakets.

Nachfolgend wird die Vorgehensweise zur Installation der neuen Beta-Version *checkinstall-1.6.0beta4.tgz* demonstriert. Dadurch werden sowohl *CheckInstall* als auch *Installwatch* und *makepak*, eine modifizierte Version von *makepkg*, installiert. Wer sich für die Änderungen in der neuen Version interessiert, sollte einen Blick in die *Release Notes* beziehungsweise in den *Changelog* auf [5] und [6] werfen.

```
$ 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 #
```

Unter Debian-basierten Distributionen kann das soeben erzeugte Paket mit *dpkg -i* installiert werden. Unter Slackware wird für gewöhnlich *installpkg* zu diesem Zweck benutzt.

Durch eine Abfrage des jeweils eingesetzten Paketmanagers, in diesem Fall RPM, läßt sich kontrollieren, ob das Programm sauber in die Datenbank integriert wurde. Nebenbei lassen sich auch gleich die generierten Informationen des Paketheaders auslesen und auf Richtigkeit überprüfen.

```
$ 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.
```

Konfiguration

Zur grundlegenden Konfiguration dient die Textdatei *checkinstallrc*, welche im Verzeichnis */usr/lib/local/checkinstall/* zu finden ist. Diese läßt sich beliebig an die eigenen Präferenzen anpassen und ist zudem gut kommentiert. Wie sich unschwer am Anfang der Datei erkennen läßt, ist es wichtig zu wissen, daß manuell übergebene Parameter vorrangig zu den hier definierten behandelt werden.

Da *CheckInstall* bei jedem Start erneut nachfragt, welcher Pakettyp erzeugt werden soll, ist es sinnvoll, den gewünschten Wert fest bei *INSTYPE* einzutragen. Ebenfalls eine genauere Betrachtung wert sind die Variablen *INSTALL*, *PAK_DIR* und *RPM_FLAGS* beziehungsweise *DPKG_FLAGS*. Mit letzteren lassen sich spezifische Optionen der verschiedenen Paketmanager definieren, *PAK_DIR* wurde weiter oben bereits kurz erwähnt und ändert den standardmäßigen Speicherort für die Paketkopie, und durch *INSTALL* kann angegeben werden, ob die Pakete nur generiert oder auch gleich installiert werden sollen.

```
$ 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

```

Fazit

CheckInstall ist ein geniales Tool, welches die Administration einer Linux-Box ungemein erleichtern kann. Wenn Programme des öfteren manuell übersetzt werden müssen, wird es durch *CheckInstall* möglich, jene wieder sauber zu deinstallieren, ohne aber einen inkonsistenten Zustand des Systems zu riskieren. Zudem lassen sich die mit *CheckInstall* erzeugten Pakete auch auf anderen Maschinen installieren, ohne jedesmal neu kompilieren zu müssen – natürlich vorausgesetzt, daß die Paketabhängigkeiten erfüllt werden. Dies stellt aber in der Regel bei identischen Distributionen kein allzu großes Problem dar.

Links

- [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>Der LinuxFocus Redaktion schreiben © Mario M. Knopf "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Autoren und Übersetzer: de --> --- : Mario M. Knopf (homepage)</p>
---	---