

# Installation von Programmen aus dem Source-Code

- Kurzakte -

<i>Voraussetzungen</i>	
<ul style="list-style-type: none"> <li>• installierte Pakete gunzip, gzip, bzip, bzip2, tar, make, gcc und devel-libraries je Anforderung des Programms</li> <li>• Quellcode (-Archiv) des Programms z. B. von <a href="http://sourceforge.net/">http://sourceforge.net/</a>, <a href="http://freshmeat.net/">http://freshmeat.net/</a>, <a href="http://www.gnu.org">http://www.gnu.org</a></li> </ul>	
<i>Erstellen eines Installationsverzeichnisses</i>	
mkdir /usr/local/src/Paketname	Erstellen eines Paketverzeichnisses
cd /usr/local/src/Paketname	Hineinwechseln
cp ... /usr/local/src/Paketname	Hineinkopieren des Quellcode-Archivs
<i>Auspacken des Software-Archivs</i>	
tar -xjf Archiv.tar.bz2 oder bzipcat Archiv.tar.bz2   tar -xvf -	Dekomprimieren und Auspacken eines bz2-gepackten Archivs, x=extract, v=verbose-Anzeige, f=file=Datei, j=entpacken von bz2-Archiven
tar -xvzf Archiv.tar.gz oder gunzip Archiv.tar.gz; tar xf Archiv.tar	Dekomprimieren und Auspacken eines tar-Archivs, x=extract, v=verbose-Anzeige, z=zip oder gzip(*.bz), f=file=Datei,
cd Softwareverzeichnis	ins soeben erzeugte Quellcode-Verzeichnis wechseln Lesen der README und INSTALL für das weitere Vorgehen
<i>Anpassung an die aktuellen Gegebenheiten des Linux-Systems</i>	
./configure	ein speziell auf dieses Softwarepaket zugeschnittenes Shellskript <ul style="list-style-type: none"> <li>• Analyse des Linuxsystems, welcher Compiler ist installiert, welcher Parser usw.</li> <li>• Erzeugen eines optimalen Makefiles bei fehlenden, nicht essentiellen Voraussetzungen wird das Makefile um die entsprechende Komponente gekürzt</li> <li>• Beeinflussung des Makefiles nach den eigenen Vorstellungen (z. B.: Umbiegen des standardmäßigen Installations-Zielverzeichnisses mit Option --prefix=\$HOME/Paketname, wenn man keine root-Rechte hat)</li> <li>• Hilfe mit Option --help</li> <li>• auftretende Fehler können an einer fehlenden oder veralteten library liegen</li> <li>• Erfolg wird mit "Creating Makefile" quittiert</li> </ul>
<i>Makefile</i>	
CC = gcc	besteht im Wesentlichen aus Variablenzuweisungen und Zielen (targets)
install-conf: kommando 1 kommando 2	<ul style="list-style-type: none"> <li>• Variablen (srcdir, prefix, installdir, mandir, confdir, CC) Zuweisung: prefix = /usr/local/Paketname Auslesen: exec_prefix = \${prefix}</li> <li>• targets (doc, install-man) – Es werden nur die gewünschten Teile gebaut/kopiert</li> </ul>
<i>Übersetzung des Programmpakets</i>	
make oder make [target]	Übersetzen *.c-Dateien und Linken der *.o-Dateien des Programmpakets <ul style="list-style-type: none"> <li>• sucht im Arbeitsverzeichnis nach der Steuerdatei Makefile und arbeitet sie ab</li> <li>• Steuerdatei besteht überwiegend aus Variablen (installdir, mandir, confdir) und Zielen (targets)</li> <li>• kompiliert wird mit dem ersten Target (make ist Kurzform von make all) oder lt. Param.</li> <li>• falls die Zieldatei (Objektdatei) nicht existiert, wird der Übersetzer gestartet</li> <li>• existiert die Zieldatei, wird der Übersetzer nur gestartet, wenn die Objektdatei älter ist als die Vorbedingungen (Quelltext-Dateien und/oder Bibliotheken) → Objektdateien können mit make clean gelöscht werden</li> </ul>
<i>Installation des Programmpakets</i>	
su make install	Programme, man, libs werden installiert, das heißt in die Zielverzeichnisse kopiert <ul style="list-style-type: none"> <li>• dazu sind meist Administratorrechte nötig (außerhalb von home)</li> <li>• Oft wird alles unter /usr/local/Programmname abgelegt, falls durch die configure-Optionen nichts anderes festgelegt wurde</li> <li>• Dann müssen folgende Anpassungen vorgenommen werden:               <ul style="list-style-type: none"> <li>○ Erweiterung von PATH um die entsprechenden bin- und sbin-Verzeichnisse</li> <li>○ Erweiterung von LD_LIBRARY_PATH oder /etc/ld.so.cache mit ldconfig</li> <li>○ Erweiterung von MANPATH, Aufruf von mandb für apropos und whatis</li> </ul> </li> </ul>