



May The Source Be With You!

Programme aus dem Quelltext übersetzen

Markus Dahms

[<mad@automagically.de>](mailto:mad@automagically.de)



Gliederung (Grundlagen)

1. Einführung	4
2. Umgang mit Quellpaketen	5
3. Überlegungen vor der Installation	7
4. Konfigurations- und Build-Systeme	8



Gliederung (GNU Autotools)

5. Die GNU Autotools	10
6. Konfiguration	11
7. „Bauen“ des Programms	12
8. Tests & Installation	13
9. Fehler aufspüren	14



Einführung – warum aus Quellen

- oft neuere Versionen verfügbar
- exotische Software manchmal gar nicht als Distributions-Paket zu bekommen
- man kann mal eben 'ne Kleinigkeit ändern:
 - Optimierung auf vorhandene Hardware
 - Konfiguration von optionalen Funktionen
 - Verbesserungen einbauen...



Umgang mit Quellpaketen – Beschaffen

- Quellpaket („Tarball“) herunterladen
(***.tar.(gz|bz2)**, ***.zip**)
- Versionskontrollsystem anzapfen (CVS, Subversion, GIT, Mercurial, Darcs ...)
- Quellen von der Distribution beziehen
 - Debian: **apt-get source <paketname>**



Umgang mit Quellpaketen – Entpacken

- ***.tar.gz: tar zxvf \$PAKET**
- ***.tar.bz2: tar jxvf \$PAKET**
- ***.zip: unzip \$PAKET (Verzeichnis im Zip?)**



Überlegungen vor der Installation

- Lohnt sich die neue Version überhaupt?
- Sind alle Abhängigkeiten erfüllt?
- Wohin installieren? – Pfad-Präfix
- Ist genug Platz verfügbar?
- Root oder nicht Root?



Konfigurations- und Build-Systeme

- Konfiguration:
 - Prüfen der Abhängigkeiten
 - benutzte Verzeichnissen und Programmen
 - optionale Programmteile und Funktionen
- Build – das „Bauen“ eines Programms, also Kompilieren und Linken mit Bibliotheken



Konfigurations- und Build-Systeme – Vertreter

- **make** – der Klassiker
- **GNU Autotools** – sehr verbreitet
- **scons** – Python-basiert
- **ant** – alles in XML
- **imake, cons** . . . und viele andere mehr.



Die GNU Autotools

- **autoconf:**
 - erstellt das Konfigurationssystem
 - erweiterbar durch Skripte und M4-Makros
- **automake**
 - erstellt Steuerdateien für **make**
 - anpassbar durch Vorlagendateien
- diverse kleine Zusatzprogramme...



Konfiguration

- *Bourne-Shell-Skript*
- **./configure**
 - **--help**
 - **--prefix** *<Pfad>*
 - **--enable-\$X**, **--disable-\$X**, **--with-\$X**,
- Umgebungsvariablen werden beachtet:
CC, CPPFLAGS, LDFLAGS, LIBS ...



„Bauen“ des Programms

- **make** tut es meistens
 - **-j<n>** – parallel bauen
 - **-k** – Fehler ignorieren



Tests & Installation

- `make check` – Testen, wenn möglich
- `make install` – Installation
 - üblicherweise als *Super-User*



Fehler aufspüren

- Compiler-Fehler

```
bla.c: In function 'main':  
bla.c:6: error: syntax error before '}' token
```

- fehlende Header-Dateien
- falscher Suchpfad für Include-Dateien

- Linker-Fehler

```
bla.o: In function 'foo':bla.c:(.text+0x1d): undefined reference to 'bar'  
collect2: ld returned 1 exit status
```

- fehlende Bibliotheken
- falscher Suchpfad für Bibliotheken



Binärpakete bauen

- Warum Binärpakete?
 - einfache Installation und *Deinstallation* über Paketsystem der Distribution
 - einfache Weitergabe an Dritte bzw. Installation auf mehreren Rechnern



Paketaufbau (Debian)

```
$ ar tv moon-buggy_0.5.1-1_i386.deb
rw-r--r-- 0/0      4 May  7 16:55 2006 debian-binary
rw-r--r-- 0/0    528 May  7 16:55 2006 control.tar.gz
rw-r--r-- 0/0  35520 May  7 16:55 2006 data.tar.gz
```

- **debian-binary**: “2.0\n” – die Versionsnummer des Debian-Dateiformats
- **control.tar.gz**: ganz wilde Kontrolldateien, die Paketinhalt und Vorgänge zur (De-)Installation beschreiben
- **data.tar.gz**: enthält alle zu installierenden Dateien mit ihren Zielpfaden



control.tar.gz

```
$ ar x moon-buggy_0.5.1-1_i386.deb; tar -zxvf control.tar.gz
./md5sums
./control
$ cat md5sums
a17501c77509f6ce509e069251d11812 opt/bin/moon-buggy
2726e8f7b871dffdfc7793651b53feb1 opt/bin/maint-buggy
d4e1c7654b55753987e2bd9f6c7d64a1 opt/info/moon-buggy.info
035b434388f29e23d156ae3d12de4929 opt/man/man6/moon-buggy.6
806c529281b287fad4a41d5b93070917 opt/man/man6/maint-buggy.6
$ cat control
Package: moon-buggy
Version: 0.5.1-1
Section: games
Priority: optional
Architecture: i386
Depends: libc6 (>= 2.3.2.ds1-21), libncurses5 (>= 5.4-1)
Installed-Size: 128
Maintainer: Jan Tobias Muehlberg <muehlber@fh-brandenburg.de>
Description: Drive some car across the moon.
```