Debianpakete erstellen:

Benötigte Pakete: (alles was zum Installieren von Source-Code Software benötigt wird.)

- Compeiler (gcc oder c++)
- ncurses (libncurses-dev)
- make
- dh-make
- libglibc2-dev
- dpkg-dev
- debhelper
- fakeroot

Das Source-Paket entpacken:

```
mkdir ~/debs
cp less-418.tar.gz ~/debs
tar -xvzf less-418.tar.gz
cd less-418
```

Nun das "Debianisieren":

```
dh_make -f ../less-418.tar.gz
```

Nun im Verzeichnis ~/debs/less-418/debian alle nötigen Dateien anpassen. Hier befinden sich einige Beispieldateien z.B.: crontabeinträge, startscripte, post/pre rm und Install scripte. Die wichtigste Datei ist die "***control***"

```
vi ~/debs/less-418/debian/control
```

```
### Notwendige einträge ###
Package: less              → Der Paketname
Version: 3.7.2             → Paketversion
Maintainer: dozent         → Paketersteller mit Mail-Adresse
Description: Less als ersats für more   → Kurzbeschreibung
 Text bla bla bla          → Lange Beschreibung muss eingerückt
                             werden

### Optionale Einträge ###
Build-Depends: debhelper (>= 5)    → Diese Pakete werden benötigt
                                     um das Paket zu erstellen
Depends: coreutils                 → Pakete die Von dem Paket
                                     benötigt werden, sind diese
                                     Nicht vorhanden, lässt sich
                                     das Paket nicht installieren
Pre-Depends: sed                   → Diese Pakete müssen
                                     installiert und Konfiguriert
                                     sein, damit sich das Paket
                                     installieren lässt
Recommends: textutils              → Liste von Paketen, die
                                     eigentlich immer mit diesem
```

	Paket auf einem System Installiert sind. Aber nicht notwendiger weise vorhanden sein müssen
Suggests: <Paketliste>	→ Pakete die dieses Paket um einige nützliche Funktionen erweitern können, aber nicht zwingend benötigt werden.
Section: text	→ Diese Paket gehört zu den Text-Tools
Architecture: i386	→ Diese Paket ist nur auf i386 Aritekturen lauffähig und lässt sich auch nur da Installieren
Essential: yes	→ Dieses Paket ist für den Betrieb des Systems notwendig, und lässt sich nicht deinstalliern
Source: less.tar.gz	→ Angabe des Original Source-Paketes
Conflicts: more	→ Unser Paket und more können nicht gleichzeitig installiert werden.
Replaces: coreutils	→ Diese Paket darf Dateien aus dem Paket coreutils ersetzen.
Provides: viewer	→ Dieses Paket stellt die Funktionalität eines viewer's zur Verfügung

Wenn alle einstellungen an den Datein vorgenommen wurden:

cd ~/debs/less-418/

Zum erstellen des Paketes nun dpkg-buildpackage aufrufen.

als user: ***fakeroot dpkg-buildpackage***
als root: ***dpkg-buildpackage***

Nun sollte das Fertige Paket im Aktuellen Ordner liegen.

RPM-Pakete:

Rpm-Pakete bestehen aus einem modifizierten CPIO-Archiv, dieses kann mit "rpm2cpio" wieder in ein CPIO-Archiv umgewandelt werden und dan mit "cpio -ivumd < paket" ausgepackt werden, es enthält dann nur noch die Datein, die ins System Kopiert werden die Informationen aus der SPEC-Datei gehen bei "rpm2cpio" verloren.

Bauen eines RPM-Paketes:

Benötigt wird: der Original Quellcode (MakeFile)
README
eventuelle Patches
SPEC-File

-> hallo-1.0.0.tar.gz

Standartmässig gibt es auf RPM-Systemen einen Ordner "***/usr/src/packages***", in diesem Ordner gibt es 5 Unterverzeichnisse:

BUILD
RPMS
SOURCES
SPECS
SRPMS

Zuerst wird der Originalquellcode in den SOURCES Ordner Kopiert.

```
cp hallo-1.0.0.tar.gz /usr/src/packages/SOURCES
```

Nun muß das SPEC-File im Verzeichnis SPECS erstellt werden

```
vi /usr/src/packages/SPECS/hallo-1.0.0.spec
```

```
Vendor:         Marko Hartig
Name:           hallo
Summary:        Das ist n kleines Hallo Welt Progamm
Version:        1.0.0
Release:        6a
Licence:        GPL
group:          none
Provides:       hallo
Requires:       bash, gcc >= 3.3, make,
Conflicts:      ncurses
Prefix:         /opt

%description
 Hier kann jetzt eine geneuere Erleuterung zum Programm
 stehen.

%prep # Vorbereiten des Kompeil- Vorgangs.
rm -rf $RPM_BUILD_DIR/hallo-1.0.0
zcat $RPM_SOURCE_DIR/hallo-1.0.0.tar.gz | tar -xvf -
zcat $RPM_SOURCE_DIR/hallo-1.0.0.patch.gz | patch -p0

%build
make clean
./configure --prefix=/opt
make
```

```
%install
make install

%files
/opt/hallo
/opt/README

%doc
README

%clean
rm -rf /tmp/rpm*

%pre      # Vor der Installation
%post     # Nach der Installation
%preun    # Vor der Deinstallation
%postun   # Nach der Deinstallation
```

Nun kann das RPM-Paket erstellt werden:

rpm -bb hallo-1.0.0.spec → Es wird ein Binärpaket erstellt, dieses ligt dann unter /usr/src/package/RPMS

rpm -bs hallo-1.0.0.spec → Es wird ein Source-Rpm erstellt, dieses ligt dann unter /usr/src/package/SRPMS

rpm -ba hallo-1.0.0.spec → Erstellt das Source und das Binär RPM-Paket

Sollte "***rpm -b***" nicht Funktionieren, müssen folgende Einträge in der "***/etc/popt***" hinzugefügt werden:

rpm	***exec --bp***	***rpmb -bp***
rpm	***exec --bc***	***rpmb -bc***
rpm	***exec --bi***	***rpmb -bi***
rpm	***exec --bl***	***rpmb -bl***
rpm	***exec --ba***	***rpmb -ba***
rpm	***exec --bb***	***rpmb -bb***
rpm	***exec --bs***	***rpmb -bs***
rpm	***exec --tp***	***rpmb -tp***
rpm	***exec --tc***	***rpmb -tc***
rpm	***exec --ti***	***rpmb -ti***
rpm	***exec --tl***	***rpmb -tl***
rpm	***exec --ta***	***rpmb -ta***
rpm	***exec --tb***	***rpmb -tb***
rpm	***exec --ts***	***rpmb -ts***
rpm	***exec --rebuild***	***rpmb --rebuild***
rpm	***exec --recompile***	***rpmb --recompile***
rpm	***exec --clean***	***rpmb --clean***
rpm	***exec --rmsource***	***rpmb --rmsource***
rpm	***exec --rmspec***	***rpmb --rmspec***
rpm	***exec --target***	***rpmb --target***
rpm	***exec --short-circuit***	***rpmb --short-circuit***

Nun sollten Problemlos RPM-Pakete erstellt werden können.