

Proseminar im Sommersemester 2008:
Linux für Umsteiger
Paketverwaltung von RPM Paketen

Philipp Stekl

13. Mai 2008

Inhaltsverzeichnis

1	Was ist RPM?	2
1.1	Definition	2
1.2	Aufbau eines Pakets	2
2	Arbeiten mit Paketen	2
2.1	Hilfsprogramme	2
2.2	Repositories	3
3	Erstellen von RPM Paketen	3
3.1	Aufbau der .spec Datei	3
3.2	Verzeichnisstruktur für rpmbuild	3
3.3	Beispiel Kernel als RPM	4
4	Quellen	5
4.1	ManPages	5
4.2	Webseiten	5

1 Was ist RPM?

1.1 Definition

Definition: RPM Package Manager (RPM, Akronym, ursprünglich Red Hat Package Manager) ist ein freies (GPL) Software-Paketverwaltungssystem entwickelt von der Firma Red Hat. Es umfasst das Paketformat und alle nötigen Programme, um RPM-Pakete zu erstellen, zu installieren und zu verwalten. RPMs (Dateiendung „.rpm“) sind Archive, die ein Softwarepaket enthalten (grob ähnlich dem .zip-Format). Das RPM-Format ist Teil der Linux Standard Base. Verwendung in vielen Linux-Distributionen (Red Hat und dessen Derivate: CentOS, Fedora, Mandriva und andere wie Suse...)

1.2 Aufbau eines Pakets

Die RPM-Dateien sind komprimiert, die einzelnen Teile können jedoch einfach nach bestimmten Informationen durchsucht werden, denn ein Kopfdatenbereich im Binärformat ist an jedes Paket angefügt. Diese Kopfdaten sind nicht komprimiert und enthalten alle wichtigen Informationen. Das erleichtert das schnelle Durchsuchen von RPM-Paketen.

BinaryRPM Pakete bestehen aus vier Teilen.

1. Der Vorspann (Lead) enthält in erster Linie Informationen zum Dateityp
2. Im zweiten Teil stehen die MD5-Summen und digitale Unterschriften (Signature)
3. Der dritte Teil (Header) liefert wichtige Metadateien und -verzeichnisse
4. Der vierte Teil enthält das CPIO-Archiv

Beispielsweise das CPIO Archiv kann mit den Programmen rpm2cpio und cpio entpackt werden.

RPM benötigt eine Datenbank, um die Informationen, welche zu jedem Paket gehören abzuspeichern. In /var/lib/rpm ist die Datenbank gespeichert. Dadurch kann gewährleistet werden, dass die nötigen Abhängigkeiten installiert werden.

2 Arbeiten mit Paketen

2.1 Hilfsprogramme

Es gibt viele Programme die RPM Dateien verarbeiten können. Sie erleichtern den Umgang mit den Paketen, bieten zusätzliche Funktionalität oder besitzen eine grafische Benutzeroberfläche. Beispiele sind: yum, yast, yumex/kyum ...

Zu den grafischen Hilfsprogrammen liegen Bilder bei.

2.2 Repositories

Software-Repository Hierbei enthält das Repository Programmpakete und zugehörige Metadaten, z.B. Beschreibungen der Pakete, Abhängigkeitsinformationen und Changelogs. Das Installieren bzw. Aktualisieren der Software aus dem Repository übernimmt ein Paketmanager. Daher wird in diesem Zusammenhang auch oft die deutschsprachige Bezeichnung Paketquelle benutzt.

3 Erstellen von RPM Paketen

3.1 Aufbau der .spec Datei

SPEC Dateien sind erforderlich, um ein RPM Paket zu erstellen. Sie enthalten eine Beschreibung der Software, Shellscrippte zu deren Übersetzung, sowie eine Dateiliste aller Binaries, die installiert werden. Sie müssen je nach Distribution und Architektur angepasst werden. Sie sind untergliedert in Sektionen mit festlegter Bedeutung.

Die Präambel: Für Benutzer relevante Informationen

Die Prep-Sektion: Befehle die den Quellcode-Dateibaum aufbauen

Die Build-Sektion: Befehle um den Quellcode zu kompilieren

Die Install-Sektion: Befehle um die Dateien an die entsprechenden Positionen zu kopieren.

Die Files-Sektion: Kein Shellscrip; Hier werden Dateien angegeben, die in das RPM-Paket übernommen werden.

3.2 Verzeichnisstruktur für rpmbuild

BUILD ist das Verzeichnis, in dem die Übersetzung durch den RPM stattfindet. Es ist nicht notwendig, die Testpakete in einem speziellen Verzeichnis erstellen, aber dies ist der Platz, wo der RPM seine Übersetzungen durchführt. SOURCES hier sollten alle tar-Archive mit den originalen Quellen und die Patches stehen. Hier sucht der RPM per Default nach den Quelldateien.

SPECS In dieses Verzeichnis gehören die Spec-Dateien.

RPMS das Verzeichnis für die fertig übersetzten und gepackten rpm's.

SRPMS das Verzeichnis für die Quell-rpm's.

3.3 Beispiel Kernel als RPM

An dieser Stelle verweise ich auf die dem Vortrag beiliegenden Bilder. Das Beispiel habe ich ausgesucht, da es für manche Benutzer auch in der Praxis Vorteile hat. Verzeichnisstruktur anlegen, und Datei `~/rpmmacros` erstellen:

```
rpmdev-setuptree
```

Kernelsource downloaden:

```
yumdownloader -source kernel
```

Benötigte Programme um Kernel zu kompilieren installieren:

```
yum-builddep kernel-2.6.24.4-64.fc8.src.rpm
```

Kernelsource nach `~/rpmbuild/SOURCES` und `.spec` Datei nach `~/rpmbuild/SPECS` kopieren:

```
rpm -Uvh kernel-2.6.24.4-64.fc8.src.rpm
```

ins SPEC Verzeichnis wechseln

```
cd ~/rpmbuild/SPECS
```

Source vorbereiten, in dem die Prepare Sektion der `.spec` Datei durchlaufen wird:

```
rpmbuild -bp -target='uname -m' kernel.spec
```

in das Verzeichnis der Kernelsource wechseln:

```
cd ~/rpmbuild/BUILD/kernel-2.6.24/linux-2.6.24.i686
```

Kernel konfigurieren:

```
make menuconfig
```

`.config` Datei kopieren:

```
cp .config ~/rpmbuild/SOURCES/config-i686
```

in das Verzeichnis der `.spec` Datei wechseln:

```
cd ~/rpmbuild/SPECS
```

Öffne die `kernel.spec` datei, und ändere die Zeile: `define buildid .local` in `define buildid .irgendwas`

Falls man einen Patch anwenden will, diesen auch in die `.spec` Datei eintragen.

Schließlich : `rpmbuild -bb -target='uname -m' kernel.spec` um den Kernel zu bauen.

4 Quellen

4.1 ManPages

Manpages (Abkürzung für Manual Page, deutsch: Handbuch-Seite) sind Hilfe- und Dokumentationsseiten unter UNIX und verwandten Betriebssystemen. Manpages sind durch POSIX standardisiert und waren ursprünglich in acht Bereiche unterteilt.

4.2 Webseiten

- docs.fedoraproject.org/
- de.opensuse.org/
- de.wikipedia.org/wiki/RPM_Package_Manager
- www.linuxhaven.de/dlhp/HOWTO-test/DE-RPM-HOWTO-8.html
- www.rpm.org/
- www.linux-praxis.de/