LaTeX Paket-Verwaltung unter Unix/Linux

Stefan Kottwitz

2. Mai 2007

Zusammenfassung

Dieser Artikel befasst sich damit, wie man unter einem Unix-Betriebssystem LaTeX-Pakete komfortabler als über die „Standardmethode“ nachinstallieren kann.

Inhaltsverzeichnis

1 Die Standardmethode

In der **UK-FAQ** (und diversen Anleitungen) wird beschrieben, wie man zusätzliche packages manuell installiert. Grob umrissen umfasst die Vorgehensweise:

- Suchen der package-Datei, am besten auf einem CTAN-Server, Download auf den eigenen Rechner
- Je nach Dateiformat des packages muss dieses ggf. noch mit latex übersetzt werden.
- Unterhalb des /texmf-Verzeichnisses (Ort installationsabhängig) müssen die zum package gehörigen Dateien je nach Typ in den entsprechenden Unterordner **einsortiert** werden.
- `texhash` bzw. `mktexlsr` ausführen

Klingt kompliziert? Das kriegt man schon hin, durch genaues Lesen jener Anleitung und Kenntnis der eigenen TEX-Installation, insbesondere deren Dateistruktur.

In der Anleitung ist diese Vorgehensweise ausführlich beschrieben, hier soll das Augenmerk auf mehr Komfort liegen und ein Paket-Manager vorgestellt werden. Mit ihm lässt sich viel Mühe sparen, insbesondere wenn man eine Vielzahl von Paketen nachinstallieren möchte.

Wer Paket für Paket manuell unter Lesen der jeweiligen Doku installiert, erwirbt natürlich eine bessere Kenntnis seines Systems!

2 Der MiKTeX Package Manager

Unter Windows stellt die beliebte Distribution MiKTeX des Autors Christian Schenk eine komfortable Paketverwaltung zur Verfügung. Benötigte Pakete werden nahezu automatisch installiert: während des Übersetzens bemerkt MiKTeX fehlende packages und bietet Download sowie Installation an. Man kann den Package Manager auch direkt im MiKTeX-Ordner des Startmenüs aufrufen.

Um diesen Komfort unter Unix erhalten, hat Christian Schenk den MiKTeX Package Manager (mpm) portiert, als Zusatz zur einer vorhandenen TEX-Distribution.

Voraussetzung für die Installation ist ein Unix-System mit vorhandenem C++-Compiler. Insbesondere eignet sich der Manager also auch für Linux sowie für Mac OS X.

Ich nutze **Ubuntu Linux** in der Version 7.04 („Feisty Fawn“) mit installiertem teTeX. Aktionen, die root-Rechte benötigen, werden unter Ubuntu Linux mit `sudo` eingeleitet. Bei Ausführung mit root-Rechten kann das `sudo` natürlich weggelassen werden.

Zuallererst muss man ein wenig Aufwand betreiben, um den mpm zu installieren, dafür wird man jedoch im weiteren mit mehr Komfort belohnt. Für das gerade frisch installierte LATEX bin ich bei Bedarf weiterer Pakete daher sofort über den mpm gegangen, ohne mir jetzt und zukünftig die Mühsal des oben beschriebenen „Fußwegs“ anzutun.

2.1 Installation

Zur Installation des Paketmanagers geht man wie folgt vor:

- Download des Archivs der Quelldateien von **sourceforge**
- Entpacken des tar.gz-Files: entweder mit dem Archiv-Manager oder über die Kommandozeile:
  `tar xfz miktex-tools-xxxxxxx.tar.gz`
- Sofern sie nicht bereits installiert sind, über die Ubuntu-Paketverwaltung „Synaptic“die Pakete g++ sowie libcurl3-dev installieren

  Herausfinden, ob libCURL installiert ist: `curl-config --version` sollte eine Ausgabe liefern ähnlich: `libcurl 7.15.5`

- zum Compilieren der miktex-tools auf der Kommando-Zeile:
  - in das Verzeichnis der entpackten Dateien wechseln: `cd miktex-tools-xxxxxxx`
  - `README`-Datei lesen ;-)
  - Ausführen zur Anpassung an die lokale Systemkonfiguration: `./configure`
  - Ausführen zum Compilieren: `make`
  - Ausführen der Installation: `make install` (mit Installations- bzw. root-Rechten). Unter Ubuntu-Linux: `sudo make install`
- Nach der Installation einmaliges Initialisieren der Namens-Datenbank: `initexmf -u`

2.2 Benutzung

Nun stehen die folgenden Kommandos (auf Shell-Ebene) zur Verfügung:

Initialisieren bzw. Aktualisieren der lokalen Paket-Datenbank:
`sudo mpm --update-db`

Auflisten der verfügbaren Pakete: `mpm --list`

Informationen zu einem bestimmten Paket:
`mpm --print-package-info=`*PACKAGE*

Installation eines bestimmten Paketes:
`sudo mpm --install=`*PACKAGE*

Dokumentation bzw. Hilfe zu einem Paket:
`sudo mthelp` *PACKAGE*
Startet einen Browser, um eine html-Seite mit Dokumentation anzuzeigen.

Aktualisieren eines bestimmten Paketes:
`sudo mpm --update=`*PACKAGE*

Aktualisieren sämtlicher Pakete:
`sudo mpm --update`

Wichtig: der LaTeX-Distribution müssen neu installierte Pakete bekanntgemacht werden mittels Aufruf von:
`sudo texhash`
in teTeX[1] bzw. `sudo mktexlsr` in TeXLive.

Ab nun reicht es meistens, den Namen des Paketes zu kennen und `sudo mpm --install=`*PACKAGE* sowie `sudo texhash` aufzurufen, statt wie eingangs angedeutet die Pakete einzeln zu suchen, herunterzuladen und manuell in die Verzeichnisstruktur einzupflegen.

[1] Die **teTeX**-Distribution wird nicht mehr weiterentwickelt, deren Autor empfiehlt einen Umstieg auf **TeXLive**.

Bei reinen Info-Befehlen kann, wie oben angegeben, auf einleitendes `sudo` verzichtet werden, in mancher Doku wird es trotzdem verwendet. Weitere Informationen bieten die Links im abschließenden Abschnitt „Referenzen“, kurze Info erhält man am Prompt mit `mpm --help` sowie `mpm --usage`.

Compilieren und Installieren des Paketmanagers liefen auf meinem System reibungslos und schnell durch. Das Installieren neuer Pakete funktionierte auf Anhieb, sie waren sofort verwendbar.

3 Shellscripte für Routine-Jobs

Noch etwas „Spielerei“ zum Schluss: um mir die Nachinstallation von Paketen ein wenig zu erleichtern, schrieb ich zwecks Ersparnis von Schreibarbeit zunächst ein kurzes Shell-Script `mpm-once`, mit dem Ziel: bei Aufruf von `sudo mpm-once package1 package2 package3` etc. sollen genau all diese Pakete installiert werden, die ich als Parameter benenne, zudem soll anschließend automatisch die ls-R Datenbank aktualisiert werden:

```
#!/bin/sh
MPM_ONCE_TEMP=/tmp/mpm-once.tmp
echo $* | tr "[:blank:]" "\n" >$MPM_ONCE_TEMP
mpm --install-some=$MPM_ONCE_TEMP
rm $MPM_ONCE_TEMP
texhash
```

Hierin werden die angegebenen Paketnamen zeilenweise in eine temporäre Datei geschrieben, die mpm übergeben wird und hinterher gelöscht wird. `tr` ersetzt dabei die Leerzeichen zwischen den Paketnamen durch Zeilenumbrüche. Als Script-Eigentümer setzte ich root: `chown` oder mit root-Rechten editieren bzw. `sudo` (`sudo gedit mpm-once`). Wo der mpm liegt, läßt sich mit `which` herausfinden:

```
which mpm
```

Ausgabe bei mir: `/usr/local/bin/mpm`. Da kann man es natürlich gleich mit kompletter Pfadangabe editieren (`sudo gedit /usr/local/bin/mpm-once`) oder man legt dann das script `mpm-once` in denselben Ordner:

```
sudo cp mpm-once /usr/local/bin/mpm-once
```

Damit das Script ausführbar ist, muss man noch das executable-Flag setzen:

```
sudo chmod u+x /usr/local/bin/mpm-once
```

Unter Verwendung dieses Scriptes kann man nun mit einem einzigen Aufruf mehrere Pakete installieren und anschließend sofort verwenden, z.B:

```
sudo mpm-once listings xypic microtype tocloft
```

Als nächstes entwarf ich ein Script `mpm-all`, das sämtliche (!) verfügbaren Pakete über den mpm nachinstalliert. Damit erhält der MiKTeX Paketmanager die maximale Kontrolle, und ein `mpm --update` aktualisiert jeweils diesen gesamten Paket-Bestand. Wieder wird `tr` verwendet, um aufeinanderfolgende Leerzeichen durch ein einzelnes zu ersetzen sowie `cut`, um die Spalte mit den Paketnamen zu selektieren:

```
#!/bin/sh
MPM_ALL_TEMP=/tmp/mpm-all.tmp
mpm --list|grep -v "^i␣" | tr -s "[:blank:]" "␣" \
| cut -d"␣" -f4 >$MPM_ALL_TEMP
mpm --install-some=$MPM_ALL_TEMP
rm $MPM_ALL_TEMP
texhash
```

Damit würde der mpm mächtig viel installieren... Bei mir trat allerdings die Fehlermeldung auf: „`mpm: Unknown archive file size.`“ – vermutlich geht nicht soviel auf einmal! Als workaround ergänzte ich das Script um eine Beschränkung auf „nur“ 100 Pakete auf einmal:

```
mpm --list|grep -v "^i␣" | tr -s "[:blank:]" "␣" \
| cut -d"␣" -f4|head -n 100 >$MPM_ALL_TEMP
```

was problemlos funktionierte. Ob es nun sinnvoll ist, *alles* zu installieren, muss man für sich selbst entscheiden.

Je nach Betriebssystem kann der Einsatz bzw. die Installation dieser Scripte etwas variieren, dieser Abschnitt richtet sich daher an Nutzer, die sich bereits etwas auskennen.

Stefan Kottwitz

4 Referenzen

miktex.org Die Homepage des MiKTeX-Projekts

miktex.org/unx Unterrubrik für den mpm auf miktex.org

mpm auf sourceforge Die Quelldateien des unix-mpm zum Download auf dem Sourceforge-Server

How to use the package manager on Unix-like operating systems Dokumentation von Christian Schenk zum Paketmanager, englischsprachig

MiKTeX_Package_Manager-HowTo Deutschsprachige Anleitung zum MiKTeX Package Manager unter Ubuntu Linux

MiKTeX Package Manager - Ubuntu help Englischsprachige Kurz-Anleitung zum MiKTeX Package Manager auf ubuntu.com

MiKTeX Package Manager Manual Englischsprachige Anleitung

UK TeX FAQ Antwort auf häufig gestellte (La)TeX-Fragen (u.a. zur package-Installation), englischsprachig