Wie ein LaTeX Dokument entsteht

Carl Wenninger

LIT 2009

1 Ein druckfertiges Dokument entsteht

1.1 Der Quelltext

Diese LaTeX-Quelldatei heißt `blatt.tex`. Sie kann mit einem beliebigen Text-Editor erstellt werden. Linux-Profis verwenden selbstredend den `vi`.

1.2 Das `latex` Kommando

Das Kommando `latex blatt.tex` ruft den LaTeX-Compiler auf. Syntaxfehler, die gerade bei Einsteigern unvermeidlich sind, werden vom Compiler gnadenlos aufgedeckt. Tipp: Mit der Option `-interaction=nonstopmode` vermeiden Sie heutzutage meist unerwünschte Nachfragen des Compilers im Fehlerfall.

1.3 Was ist eine dvi-Datei?

Wenn alle Fehler ausgemerzt sind, so erzeugt LaTeX eine Datei `blatt.dvi`.[1] Die Endung `dvi` steht für »device independent«.[2] Mit einem zu Ihrem Drucker passenden DVI-Druckertreiber können Sie diese Datei im Prinzip ausdrucken. Zumeist geht man aber heute einen anderen Weg, indem man Postscript bzw. PDF-Dateien generieren lässt.

[1] Als Nebeneffekte entstehen Dateien wie `blatt.log` oder `blatt.toc` oder `blatt.aux`. Auf diese kann hier nicht eingegangen werden.

[2] Vgl. `http://de.wikipedia.org/wiki/Device_independent_file_format`

1.4 Die Konverter `dvips` und `dvipdf`

Ein Aufruf von `dvips blatt.dvi` bzw. `dvipdf blatt.dvi` erzeugt Dateien `blatt.ps` bzw. `blatt.pdf`. In diesem Schritt werden auch eventuell verwendete Grafiken im encapsulated postscript Format (Endung meist `.eps`) mit ins Dokument eingebunden.

1.5 Die Alternative `pdflatex`

`pdflatex` ist ein LaTeX-Compiler, der unmittelbar PDF-Dokumente erzeugt. Eine DVI-Datei wird hier nicht benötigt. Es besteht aber ein gravierender Unterschied bei der Einbindung von Grafiken: Während `latex` ausschließlich `.eps`-Grafiken verlangt, arbeitet `dvipdf` sowohl mit `.jpg` wie auch `.png`, jedoch nicht mit `.eps`-Dateien zusammen. Da METAPOST, mit dem wir uns ja hauptsächlich beschäftigen wollen, emcapsulated postscript erzeugt, ist `pdflatex` für unsere Zwecke leider nicht geeignet.

2 Ein `Makefile` erleichtert den Build!

Programmierer lieben `make`. Mit einer passenden Steuerdatei, dem sogenannten `Makefile`, werden auch komplizierte Build-Prozesse zum Kinderspiel! In diesem Verzeichnis befindet sich neben `blatt.tex` auch so ein `Makefile`. Ein Aufruf von `make` genügt und es werden alle für `blatt.ps` benötigten Zwischenschritte ausgeführt. Allerdings nur, falls diese notwendig sind.

2.1 Ein Beispiel

Im Makefile finden sich die Zeilen[3]

```
{Leerzeile}
blatt.dvi: blatt.tex
        latex -interaction=nonstopmode blatt.tex
{Leerzeile}
```

Dies bedeutet im Klartext: Falls `blatt.dvi` nicht existiert oder älter ist als `blatt.tex`, so baue mir `blatt.dvi` durch Aufruf des nachfolgenden latex-Kommandos. Eine etwas ausführlichere Beschreibung zu `make` findet sich unter `man make` bzw. `http://de.wikipedia.org/wiki/Make`.

[3]Neben den beiden Leerzeilen ist hierbei entscheidend, dass die Kommandozeile mit dem latex-Aufruf *mit einem Tabulator-Zeichen* (ASCII-Zeichen Nummer 8) beginnt!

2.2 Aufruf von `make`

Auf der Shell geben wir einfach `make` ein. Alternativ liefert `:make` im vi-Kommandomodus das selbe Ergebnis — nur viel komfortabler.

Unser Makefile enthält auch ein Target namens clean. Es löscht alle überflüssigen Dateien (Achtung: Wildcards!), die durch einen erneuten Aufruf von make hoffentlich wieder erstellt werden können.

Als Default-Target (das erste Target im Makefile) wird `blatt.ps` erzeugt. Ich verwende während der Dokumenterstellung `gv` (ghostview) zur Visualisierung des Dokuments. Wenn man im State-Menu von gv die Option »watch file« aktiviert hat, genügt ein »:make« im vi und schon sieht man das Endergebnis!