

Grundsätzliche Funktionsweise

L^AT_EX ist ein professionelles, auf der Basis von T_EX erstelltes Makropaket. Wie in anderen Programmiersprachen, so wird auch hier ein Programmcode zu einem Ergebnis, in diesem Fall zu einem gesetzten Text, kompiliert, d. h. umgesetzt. Im Grunde ist der Vorgang mit dem Arbeitsablauf der Entstehung professioneller Druckerzeugnisse vergleichbar. Zahlreiche Befehle in T_EX stammen daher aus der Fachsprache des Druckereiwesens.

Die folgende Liste entspricht ungefähr dem normalen Ablauf der Herstellung eines Druckerzeugnisses:

1. Der Autor verfasst einen Text und zeichnet beispielsweise bestimmte Textstellen als Überschrift aus. »Auszeichnen« ist der Fachbegriff, der die Kennzeichnung von Textstellen mit Funktionen meint. Z. B.: Dieser Satz ist eine Überschrift.
2. Der Text wird dann an weitere Instanzen der Produktion weitergegeben. Dies wäre im normalen Arbeitsablauf die Person, die den Satz und die Gestaltung plant und übernimmt. Anstelle eines professionellen Setzers kommt nun L^AT_EX zum Zuge. Durch die Umsetzung der Befehle, die dem Text hinzugefügt sind, wird bei der Kompilation ein perfekt gesetzter Text produziert, der auch höchsten typographischen Kriterien entspricht. Die eben genannte Überschrift wird als solche erkannt. Das Programm wählt nun die entsprechende Schriftgröße und die Relation der Schriftgrößen anhand professioneller Vorgaben aus.

Um einmal die Arbeitsweise zu verdeutlichen, stelle ich an dieser Stelle den Quell-Code des bisherigen Textes dar. Die % -Zeichen blenden die nachfolgenden Texte oder Befehle für den Kompilierungsvorgang aus.

```
\documentclass[a4paper, twoside, DIV10,BCOR12mm, 12pt]{scrbook}
\usepackage[T1]{fontenc}
\usepackage[applemac]{inputenc}

\usepackage{ngerman}
\usepackage{graphicx}
\usepackage{pdfpages}
\usepackage{pifont}
```

```
\usepackage{scrpage2}
\setkomafont{sectioning}{\normalfont\normalcolor\bfseries}
%\usepackage{times}
```

```
%\usepackage{mathptmx}
\usepackage{lmodern}
```

```
\usepackage{typearea}
```

```
\typearea{9}
```

```
\pagestyle{scrheadings}
\setlength{\parskip}{5pt plus 2pt minus 1pt}
\parindent0cm
\ohead{\pagemark}
\lefoot{}
\refoot{}
\lofoot{}
\rofoot{}
```

```
\begin{document}
```

```
\section*{Grundsätzliche Funktionsweise}
\thispagestyle{empty}
```

```
%\fontfamily{cmdh}
%\selectfont
```

`\LaTeX{}` ist ein professionelles, auf der Basis von `\TeX{}` erstelltes Makropaket. Wie in anderen Programmiersprachen, so wird auch hier ein Programmcode zu einem Ergebnis, in diesem Fall zu einem gesetzten Text, kompiliert, d. h. umgesetzt. Im Grunde ist der Vorgang mit dem Arbeitsablaufs der Entstehung professioneller Druckerzeugnisse vergleichbar. Zahlreiche Befehle in `\TeX {}` stammen daher aus der Fachsprache des Druckereiwesens.

Die folgende Liste entspricht ungefähr dem normalen Ablauf der Herstellung eines Druckerzeugnisses:

```
\begin{enumerate}
\item Der Autor verfasst einen Text und zeichnet beispielsweise bestimmte Textstellen als Überschrift aus. »Auszeichnen« ist der Fachbegriff, der die Kennzeichnung von Textstellen mit Funktionen meint. Z. B.: Dieser Satz ist eine Überschrift.
\item Der Text wird dann an weitere Instanzen der Produktion weitergegeben. Dies wäre im normalen Arbeitsablauf die Person, die den Satz und die Gestaltung plant und übernimmt. Anstelle eines professionellen Setzers kommt nun \LaTeX{} zum Zuge. Durch die Umsetzung der Befehle, die dem Text hinzugefügt sind, wird bei der Kompilation ein perfekt gesetzter Text produziert, der auch höchsten typographischen Kriterien entspricht. Die eben genannte Überschrift wird als solche erkannt. Das Programm wählt nun die entsprechende Schriftgröße und die Relation der Schriftgrößen anhand professioneller Vorgaben aus.
\end{enumerate}
```

Um einmal die Arbeitsweise zu verdeutlichen, stelle ich an dieser Stelle den Quell-Code des bisherigen Textes dar. Die \% -Zeichen blenden die nachfolgenden Texte oder Befehle für den Kompilierungsvorgang aus.

»usepackage« bedeutet die Einbindung besonderer Befehlssammlungen, die sog. Makros enthalten. Die Bedeutung ist in etwa mit einem Makro in Word vergleichbar. (Enthält jedoch keine Viren!!!!)

Jeder Befehl muss mit einem Rückstrich eingeleitet werden, damit T_EX Befehle als solche erkennt. Ähnlich funktioniert auch HTML, die Seitenbeschreibungssprache für Webseiten. Doch findet dort kein Kompilationsvorgang statt, sondern eine Interpretation, die – je nach Browser – unterschiedlich ausfallen kann.

L^AT_EX hat zahlreiche Vorteile gegenüber Word und ähnlichen Programmen:

1. Mittels $\text{T}_{\text{E}}\text{X}$ erstellte Druckerzeugnisse zeichnen sich durch ihre hervorragende typographische Qualität aus. Das ist das stärkste Argument.
2. $\text{T}_{\text{E}}\text{X}$ läuft auf allen Betriebssystemen. Ich selbst habe Versionen unter ATARI, Windows ME und Mac OS X verwendet.
3. Jede Distribution erfüllt einen Mindeststandard. Es gibt keine Inkompatibilitäten wie sie Word in den zahlreichen Versionen aufweist.
4. Das System stürzt nie ab.
5. $\text{T}_{\text{E}}\text{X}$ verteilt keine Viren.
6. Das System ist transparent. Der Code der packages (im weitesten Sinne mit CSS in HTML vergleichbar, nur viel umfangreicher) ist mit einem normalen Texteditor lesbar.
7. Der Preis verhält sich umgekehrt proportional zur Leistungsfähigkeit. $\text{T}_{\text{E}}\text{X}$ kostet nichts: Public Domain! Bei den vielen Textprogrammen ist es ähnlich: Sie leisten allerdings wenig, sind aber oft sehr teuer.
8. Das Ergebnis ist das \LaTeX -eigene DVI-Format oder PDF. Fast alle Druckereien verwenden heute PDF. Ein weiteres Format, das im Druckereiwesen Standard ist, heißt EPS, ein Postscript-Format, das die meisten, aber nicht alle Druckanlagen verstehen, jedoch im professionellen Druck üblich ist.
9. Die Aufgabe des professionellen Setzers besteht nun in der Auswahl geeigneter Schriftarten, eines geeigneten Satzspiegels (dies ist der bedruckte Bereich der Seiten und die gespiegelte Anordnung der Einzelseiten auf einem Druckbogen). Kurz: \LaTeX übernimmt alle typographischen Entscheidungen.

Ausschießen von PDF-Dateien unter \LaTeX

Gerade heute hatte ich sehr viel Freude an $\text{T}_{\text{E}}\text{X}$. Ich habe aus einer PDF-Datei eine für den Druck geeignete Verteilung auf einem Druckbogen vorgenommen. Diesen Vorgang nennt man in der Druckerfachsprache »Ausschießen«. Dies bedeutet, dass eine Datei mit vier DIN A4-Seiten wie folgt verteilt sein muss:

Außenseite auf DIN A3: Seiten 4 und 1,
Innenseiten auf DIN A3: Seiten 2 und 3.

So können im DIN A4-Format zwei DIN A3 Bögen bedruckt werden. Diese werden dann gefalzt und mittels Heftklammern gebunden.

Hier ist der Quelltext:

```
\documentclass[12pt]{scrbook}
\usepackage[T1]{fontenc}
\usepackage[applemac]{inputenc}

\usepackage{geometry}
\geometry{verbose, a3paper, twoside, nofoot, includeheadfoot,%
pdftex}
\usepackage{pdfpages}

\begin{document}

\includepdf[pages={4,1,2,3}, landscape, nup=1x2, noautoscale]
{WeihnachtsliederUmschlag.pdf}

\end{document}
```

Der Kompilationsvorgang dauert auf meinem iMac gerade einmal drei Sekunden! Doch Distributionen für Windows bringen vergleichbare Geschwindigkeiten zustande.

HTML ... XML und \LaTeX

\LaTeX ist leicht zu erlernen. Es gibt zwar mehr Befehle als in HTML. Diese sind jedoch sehr logisch aufgebaut. Fehler werden während der Kompilation kenntlich gemacht. Fehlermeldungen enthalten Hinweise auf die jeweilige Lösung.

Benutzer, die schon einmal mit HTML-, oder XML-Code zu tun hatten, finden leicht Zugang zu \LaTeX . Die Begriffe sind verwandt. So heißen Befehle in \LaTeX Befehle oder commands, in HTML spricht man von Tags oder tags. Befehle

enthalten in \LaTeX Argumente oder in HTML Attribute. Sie können sich auf eine Umgebung beziehen: Dazu ist in \LaTeX keine Syntax notwendig, die eine Umgebung definiert:

```
\begin{Argument}
Text
\end{Argument}
```

HTML kennt dazu Anfangs- und Endtags.

Eine Gegenüberstellung der Dokumentenstruktur beider Seitenbeschreibungssprachen zeigt dies deutlich:

In HTML lautet die Grundstruktur eines Dokuments:

```
<html>
<head>
</head>
<body>
</body>
</html>
```

Jeder Befehle ist in Klammern gefasst, die den enthaltenen Text als Befehl kennzeichnen. Zu einem Befehl gehört auch ein schließender Befehl, der an dem vorangestellten Schrägstrich zu erkennen ist.

In \LaTeX ist die Grundstruktur der in HTML sehr ähnlich:

Der Quelltext enthält eine Präambel, die Dokumentenstile, den Dokumententyp (Artikel, Bericht, Buch ...) definiert. Dies entspricht dem `<head>`-Tag in HTML.

```
\documentstyle{hier stehen verschiedene Stile}
\usepackage[T1]{fontenc} Dies betrifft die Codierung,
in HTML z. B.: charset="UTF-8"
\usepackage[applemac]{inputenc}
\usepackage{ngerman}
```

```
\usepackage{graphicx}
\usepackage{pdfpages}
```

Mit `\usepackage` lassen sich entsprechende Makropakete in das Dokument einbinden. Diese gelten für das gesamte Dokument.

```
\begin{document}
```

Dies entspricht `<body> </body>` in HTML. Hier steht der eigentliche Text.

Grundsätzlich entspricht dem Paar `<>` und `</>`, also dem Anfangs- und Endtag die sogenannte Umgebung.

```
\end{document}
```

Auch andere Befehle sind verwandt:

Überschriften:

In HTML:

```
<h1> </h1>
```

```
<h2> </h2>
```

```
<h3> </h3>
```

usw.

In \LaTeX :

```
\part{Überschrift}
```

```
\section{Überschrift}
```

```
\subsection{Überschrift}
```

Listen:

In HTML:

```
<ul> Ungeordnete Liste
```

```
<li> Listenelement
```

```
</ul> Ende der Liste
```

oder

```
<ul> Geordnete Liste
<li> Listenelement
</ul> Ende der Liste
```

```
\begin{itemize} Unnummerierte Liste
\item Listenelement
\end{itemize}
```

oder:

```
\begin{enumerate}
\item
\end{enumerate}
```

Das alles ist sehr einfach zu bedienen. Vor allem aber ist das System nicht nur äußerst zuverlässig: Das Verfahren erzeugt Textsätze bester Qualität.

Übrigens existieren Konvertierungsprogramme, die \LaTeX -Code in HTML oder XML konvertieren. Auf der Apple-Seite kann »SimpleTeX4Ht« (Freeware) direkt herunter geladen werden kann. Nach der Installation, können die Quelldateien direkt in diesem Programm in HTML oder XML konvertiert werden.

Im Unterschied zu ähnlichen Programmen für Windows können auch relativ komplizierte Strukturen konvertiert werden, Bilder lassen sich ebenfalls in den Text integrieren (nur keine PDF-Dateien, eher PNG- oder JPG-Graphik), sogar das \LaTeX -Logo wird korrekt umgesetzt. Komplizierte Abstandskorrekturen, für die in \LaTeX entsprechend anspruchsvolle Berechnungsgrundlagen eingesetzt werden, führen in HTML zu einer Vielzahl von Korrekturen und einem komplizierten Code, der jedoch notwendig ist, um die Typographie unter HTML zu simulieren.

Schlussbemerkung

Leider ist \LaTeX in den Geisteswissenschaften und in der Kunst nicht besonders verbreitet. Gleichzeitig verwenden Anwender dieser Disziplinen oft Mac OS.

Die Anwendung von \LaTeX ist dank des hervorragenden Formelsatzes unter Mathematikern sehr hoch (nahezu 100%). Akademische Mathematiker schreiben fast ausschließlich in \LaTeX . Sie arbeiten meist unter Linux oder Windows.

Daraus folgt leider auch, dass Mathematiker in der Regel hochwertigere Druckerzeugnisse vorlegen als beispielsweise Germanisten. Window vs. Mac OS ist dann vielleicht nicht die wichtigste Frage, sondern eher Word oder \LaTeX . Wahrscheinlich ist jedem Leser der Glaubenskrieg zwischen Mac OS und Windows bekannt. Aus meiner Sicht ist Mac OS sicher das bessere und zuverlässigere Betriebssystem und auch das bessere Konzept eines Computers. Warum aber die meisten Mac OS-Apologeten selbst nur Word und Excel benutzen, ist mir jedoch unverständlich. Word ist und bleibt auch unter Mac OS ein schwaches Programm.

Die Benutzervereinigung Dante stellt im Internet ausführliche Anleitungen zur Verfügung. \LaTeX ist ein Textsatzprogramm für wissenschaftliche Arbeiten. Es gibt jedoch auch Anpassungen für Briefe oder sogar für juristische Texte. Sogar Schachdokumentationen und Noten sind möglich. Alle als PDF-Dokumente gespeicherten Texte sind mit \LaTeX gesetzt und geben so einen Eindruck von dem Ergebnis.

Die Titel und Texte meiner Notenausgaben sind ebenfalls mit \LaTeX geschrieben. Im reinen Notensatz hat sich \LaTeX allerdings nicht bewährt. Notensatz erfordert die Korrektur des Augenmaßes. Hier versucht Lilypond mit einer ähnlichen, von Benutzeroberflächen unbelasteten Struktur die Anforderungen der Notation von Musik zu meistern.

Weitere Informationen, professionelle Darstellungen und Einführungen findet man in allen wissenschaftlichen Instituten und Universitäten und unter der Internetseite der Benutzervereinigung: www.dante.de.

18. Dezember 2007

Michael Sieberichs-Nau