

Einführung in das Textsatzsystem



L^AT_EX



08 – Umfangreiche Dokumente A SMALL STEP FOR THE LION ...

20. Dezember 2013

Inhalt

- 1 Grundstruktur
- 2 Header
- 3 Titelei
- 4 Verzeichnisse (TOC, LOF, LOT)
- 5 Fußnoten, Randbemerkungen
- 6 Verweise, Referenzierungen
- 7 Bibliographie
- 8 Codebeispiele
- 9 Indexgenerierung

Dokumentelemente

- Schmutztitel, Titelei
- Gliederung
- Verzeichnisse
- Kopf-/Fußzeilen
- Fußnoten, Randbemerkungen
- Formeln, Abbildungen, Tabellen etc.
- Programmcode
- (interaktive) Verweise
- Anhang: Bibliographie, Indices, Verzeichnisse
- Danksagung (meist nicht offiziell Teil des Dokumentes)

Aufteilung

- Nachteil von T_EX: lange Dokumente werden unübersichtlich

Aufteilung

- Nachteil von T_EX: lange Dokumente werden unübersichtlich
 - + Vorteil von T_EX: Teile des Dokumentes können in externe Dateien ausgelagert werden
- ⇒ geschickte Aufteilung und Verwaltung eines Dokumentes:

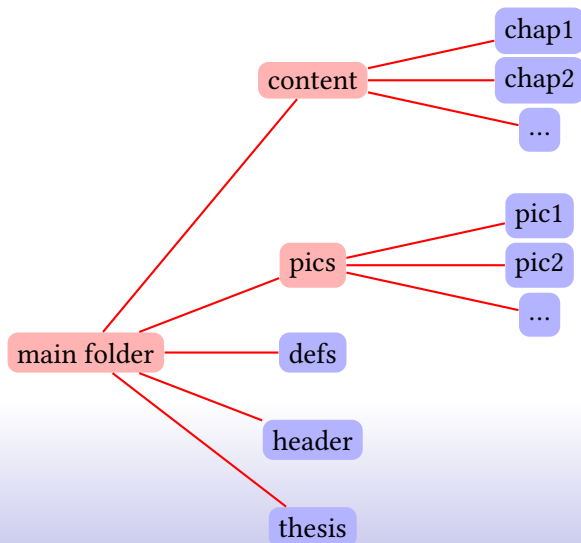
Aufteilung

- Um riesige Dateien zu vermeiden: Quellcode gemäß Inhalt aufteilen, z. B. eine Datei pro Kapitel
 - eine *Hauptdatei* als leeres Gerüst
 - eine *header*-Datei (Trennung von Inhalt und Formatierung; evtl. weitere Datei(en) für spezielle Befehlsdefinitionen)
- ⇒ saubere Trennung von Inhalt und T_EXnik.
- Inhalte in einem Unterordner nach strukturierter Anordnung
 - Abbildungen, sonstige Materialien in weiteren Unterordnern (Bilder einfügen mit `\graphicspath`)

Aufteilung

- Um riesige Dateien zu vermeiden: Quellcode gemäß Inhalt aufteilen, z. B. eine Datei pro Kapitel
 - eine *Hauptdatei* als leeres Gerüst
 - eine *header*-Datei (Trennung von Inhalt und Formatierung; evtl. weitere Datei(en) für spezielle Befehlsdefinitionen)
- ⇒ saubere Trennung von Inhalt und T_EXnik.
- Inhalte in einem Unterordner nach strukturierter Anordnung
 - Abbildungen, sonstige Materialien in weiteren Unterordnern (Bilder einfügen mit `\graphicspath`)
 - *Alles*, was man im Rahmen der Arbeit braucht, sollte innerhalb eines Ordners (plus Unterordnern) sein!
 - syncT_EX hilft, in die jeweils richtige Datei zu springen

Aufteilung



input, include, includeonly, excludeonly

- `\input` und `\include` führen externe Dateien am angegebenen Ort aus
- T_EX „springt“ aus dem aktuellen Dokument, liest woanders, und springt wieder zurück

input, include, includeonly, excludeonly

- `\input` und `\include` führen externe Dateien am angegebenen Ort aus
- T_EX „springt“ aus dem aktuellen Dokument, liest woanders, und springt wieder zurück
- `\input` liest den Code einfach ein, als gehöre er ins Hauptdokument
- `\include` erstellt eigene `.aux`-Datei (sinnvoll, wenn `.aux` benötigt, s. u.)
- `\includeonly{a.tex,b.tex}` in der Präambel lässt nur die angegebenen Dateien für `\include` zu
- `\excludeonly{b.tex,c.tex}` lässt die angegebenen Dateien für `\include` *nicht* zu (benötigt Paket `excludeonly`)

Hauptdokument

Ein Beispieldokument

```
\input{header}  %% beinhaltet \documentclass, \usepackage etc.

\includeonly{chapter1}
\excludeonly{anhang} % erfordert Paket excludeonly!

\begin{document}
\include{chapter1}
\include{chapter2}
...
\include{anhang}
\end{document}
\end{document}
```

⇒ Nur chapter1 wird hier gesetzt, anhang explizit nie.

root Dokument

- nach Aufteilung muss immer das Hauptdokument kompiliert werden
⇒ ständiges Wechseln zwischen Dokumenten

root Dokument

- nach Aufteilung muss immer das Hauptdokument kompiliert werden
- ⇒ ständiges Wechseln zwischen Dokumenten
- gute Editoren nehmen die Arbeit ab:
 - Definition von Hauptdokumenten möglich
 - in T_EXworks: Setzen von `% !TeX root = Hauptdokument.tex`
in jedem Dokument verweist auf das Hauptdokument
 - Dokument im Oberverzeichnis:
`% !TeX root = ../Hauptdokument.tex`
 - Kompiliert automatisch das zugehörige Hauptdokument

Header-Dokument

- lädt Dokumentenklasse, Pakete und setzt Optionen
- bestimmt den Satzspiegel
- legt Schriften fest (Broschrift, Überschriften, ...)
- ...
- alles, was vor `\begin{document}` steht

Titelei

- enthält alles bis zur ersten Inhaltsseite
- wird vor `\begin{document}` definiert
- enthält Autor, Titel, etc.
- mit KOMA: Dokumentoption `titlepage=true/false` setzt eigene Seiten oder einen Titelpopf
- Befehl `\maketitle` setzt vordefinierte Titelei
- Angaben von `\title`, `\author`, `\extratitle` etc. nötig und möglich
- Umgebung `\begin{titlepage}` setzt eine frei gestaltbare Titelseite

Titeleibefehle im KOMA-Bundle (unvollständig)

```
\documentclass{scrbook}
\usepackage{fontspec}
\titlehead{{\Large Universität Schlauenheim}}
\subject{Diplomarbeit}
\title{Digitale Raumsimulation mit dem DSP\,56004}
\subtitle{Klein aber fein?}
\author{cand.\,stup. Uli Ungenau}
\date{30. Februar 2001}
\publishers{Betreut durch Prof.\,Dr.\,rer.\,stup. Naseweis}
\dedication{I love you to peaces, to thee I dedicate this thes
\begin{document}
\maketitle
\end{document}
```


abstract

- Umgebung `abstract` setzt eine kurze Zusammenfassung des Dokumentes
- wird ins Dokument geschrieben (nicht im header)
- mehrere Abstracts möglich (z. B. englisch / deutsch etc.)
- *nicht* verfügbar in den Buchklassen!

TOC, LOF, LOT

- Verzeichnisse fassen strukturierte Elemente zusammen
- prinzipiell kann alles in ein eigenes Verzeichnis aufgenommen werden
- üblich:
 - Inhaltsverzeichnis (`\tableofcontents`)
 - Abbildungsverzeichnis (`\listoffigures`)
 - Tabellenverzeichnis (`\listoftables`)
- Aufnahme der Verzeichnisse ins Inhaltsverzeichnis: Dokumentenoption `toc=totoc`
- möglich: Codeverzeichnis, Beispielverzeichnis, ...

Fußnoten, Randbemerkungen

- zusätzlicher Text, der nicht ins Hauptdokument / in den Textfluss passt
- `\footnote{}`
- `\marginpar[left]{right}`: gleitende Randnotiz
- `\marginnote[left]{right}`: nichtgleitende Randnotiz (erfordert Paket `marginnote`)

Verweise

- Nummerierte Elemente können mittels `\label{}` bezeichnet werden
- mögliche Elemente sind Überschriften (sections etc.), `table`, `figure`, Formeln, Fußnoten, ...
- Referenzierung im Text mit `\ref{}`
- Pakete liefern sehr vielfältige Referenzierungen:
`fancyref`, `varioref`, `cleveref`
(Auf Kompatibilität achten, z. B. mit `hyperref`)
- geschicktes Benennen:
- `\label{fig:elephant}`
⇒ Pakete können erkennen, dass es eine Abbildung ist
- `hyperref` erkennt mit `\autoref` automatisch, welches Objekt vorliegt!
- *aber*: `\autoref` liefert nur Ausgabe in Großbuchstaben, was im englischen Satz stören kann!

Verweise anpassen

- im fertigen (pdf-)Dokument sollten alle Verweise als Hyperlinks ausgeführt sein
- Standardeinstellungen von `hyperref` können fragwürdig sein
- Anpassungen mittels `\hypersetup{}` möglich
- vor allem `colorlinks=true,linkcolor,urlcolor,citecolor=blue` u. ä.

Dokumentinformationen

- pdf-Format sieht Metainformationen über das Dokument (Autor, Herausgeber etc.) vor
- ebenfalls mittels `\hypersetup{}` einstellbar
- z. B. `pdftitle`, `pdfauthor` (automatisch gesetzt)
- Optionen zum Öffnen des Dokuments im Viewer, z. B. `pdfstartpage`, `pdftoolbar` etc.

Erstellen von Bibliographien

- Verweise auf Literatur sollten weitestgehend automatisiert werden
⇒ vermeidet Fehler

Erstellen von Bibliographien

- Verweise auf Literatur sollten weitestgehend automatisiert werden
⇒ vermeidet Fehler
- je nach Umfang ist manuelles Anordnen und Sortieren sinnvoll oder Zuhilfenahme externer Programme und Pakete
- externe Programme basieren auf (selbst erstellter) Datenbank
- BibT_EX bzw. biber *sortieren* Einträge
- Pakete wie biblatex oder natbib *formatieren* die Literaturangaben und -verzeichnisse

Literaturverweise ohne externe Hilfe

- `\cite[text]{cite key list}` setzt (automatisierten)
Literaturverweis
Beispiel: `text = Chapter 2`, `cite key=jon90`, `label` (automatisch erzeugt)
= „Jones et al. (1990)“
⇒ Jones et al. (1990, Chapter 2)
- `label` wird aus einem Bibliographieeintrag erzeugt, entweder extern
(mittels BibT_EX) oder intern mittels `thebibliography`:

```
\begin{thebibliography}{widest entry}  
\bibitem[Thor90]{thor} A. U. Thor, \textit{Elephants}, 1990  
\bibitem[Ele91]{phant} L. Ephant , \textit{My Thor and I}, 1990  
\end{thebibliography}
```

Bibliographie

- Aussehen der Bibliographie wird von der Dokumentenklasse bzw. Paketen verwaltet und beeinflusst
- manuelles Erstellen (Sortieren etc.) im Dokument möglich, aber umständlich
- Einträge nicht wiederverwertbar!

Bibliographie

- Aussehen der Bibliographie wird von der Dokumentenklasse bzw. Paketen verwaltet und beeinflusst
 - manuelles Erstellen (Sortieren etc.) im Dokument möglich, aber umständlich
 - Einträge nicht wiederverwertbar!
- ⇒ Programm BibT_EX übernimmt Sortierung und Verwaltung der Einträge
- biber als neuste BibT_EX-Weiterentwicklung ist utf8-fähig und sehr flexibel einsetzbar – allerdings nur mit Paket biblatex

Bibliographie mit BibT_EX

- Bibliographieeinträge liegen als externe Textdatei (.bib) in fest vorgegebener Syntax vor
- Referenz im Dokument mit `\cite{mittelbach2004}`
- Art der Referenz vielfältig einstellbar
- Zugriff auf große Menge an verfügbaren Referenzen
- Beispieleintrag:

```
@Book{Danzer1972,  
  author    = {Danzer, Klaus},  
  title     = {{Robert W. Bunsen und Gustav R. Kirchhoff}},  
  publisher = {B. G. Teubner},  
  address   = {Leipzig},  
  year      = {1972}}
```

BibT_EX

- Verwendung unintuitiv
- graphische Oberflächen erleichtern das Leben
- z. B. jabref, citavi, etc.
- direkte online-Suche z. B. beim Abstract Service des ADS:
http://adsabs.harvard.edu/abstract_service.html
oder bei google scholar:
<http://scholar.google.de/>

Pakete

- Gestaltung der Bibliographie mittels Paketen
- Am einfachsten: `biblatex`
- ältere, verbreitete Pakete: `natbib`, `jurabib`

Erstellung: biblatex

- sämtliche Layouteinstellungen sind über reine L^AT_EX-Makros definiert
 - andere Syntax als die „alten“ Pakete (verwenden BibT_EX-Code)
 - Einstellungen über Paketoptionen (siehe [biblatex](#)-Dokumentation, Abschnitt 3 User guide)
 - erst ab T_EX Live 2010 verfügbar
 - kann statt BibT_EX auch den Nachfolger biber verwenden
- ⇒ utf8-fähig!

Verwenden von biblatex

- Im Header:
`\usepackage[backend=biber]{biblatex}` (stellt auf biber als backend um, statt BibT_EX zu verwenden)
`\bibliography{mybib}` (Bibliographiedatei mybib.bib muss schon im Header angegeben werden!)
- Im Dokument:
`\cite{test}` wie gewohnt
- Zum Setzen der Bibliographie:
`\printbibliography`

Verwenden von natbib

```
\usepackage[optionen]{natbib}
\begin{document}
\bibliographystyle{plainnat} % oder andere ...
Text ... \cite{quelle} \citet{quelle} \citep{quelle}
\bibliography{Bibquellen}
\end{document}
```

Setzen von Code

- für kurze Sequenzen: `\verb~\befehl~`
- für längerer Sequenzen:
`\begin{verbatim} \befehle \end{verbatim}`
- beide bieten *-Version für Anzeigen von Leerzeichen:
- für Setzen von Programmcode: Paket `listings`
- für Setzen von L^AT_EX-Beispielcode: Paket `showexpl`

Indexerstellung

- Indexerstellung ist immens aufwändiges Unterfangen:
- sämtliche (sinnvollen!) Erscheinungen von
Namen / Ereignissen / Sachthemen müssen registriert werden
nicht jede Nennung eines Namens soll im Index erwähnt werden!
(Volltextsuche ist in digitalen Dokumenten ohnehin schneller.)
- sinnvolle Seitenangabe: 1, 2–4, 17 statt 1, 2, 3, 4, 17

Indexerstellung

- dank logischer Struktur leichte Erstellung in T_EX:
- Definieren von Befehlen erleichtert die Eingabe:
`\kirchhoff` statt `Kirchhoff` `\index{Kirchhoff}`
- mit L^AT_EX dreistufiger Prozess:
- im L^AT_EX-Lauf wird Hilfsdatei erstellt
- Verarbeitung mittels Programm `makeindex` (Sortierung, Seitenangaben etc.)
- Einbettung im nächsten L^AT_EX-Lauf

Indexerstellung

makeidx

im Dokument

```
\usepackage{makeidx}  
\makeindex{stichwoerter} % VOR \begin{document}!!  
\index{Stichwort} %% IM Dokument!  
\printindex{stichwoerter} % druckt das Verzeichnis hier
```

in der Kommandozeile

Aufruf von `makeindex` `hauptdocument` im Ordner des Hauptdokumentes

Indexerstellung

multind

multind ermöglicht Erstellung mehrerer Indizes – Unterscheidung mit zusätzlichem Attribut:

im Dokument

```
\usepackage{multind}  
\makeindex{stichwoerter}\makeindex{Personen}  
\index{Stichwoerter}{Stichwort}\index{Personen}{Kirchhoff}  
\printindex{stichwoerter}{Index der Stichwörter} \printindex{p
```

in der Kommandozeile

```
makeindex personen makeindex stichwoerter
```