Chapter 2 Typesetting Text

Du Ang
du2ang233@gmail.com

2017 年 5 月 28 日

1 文章和语言的结构

写文章最重要的一点是要把想法、信息、知识传达给读者，而好的文章结构能够帮助读者更好地查看、感受和理解我们想传达的东西。

LaTeX 中最重要的文本单位是段（paragraph）。一段文字应该只包含一个思想或一种想法。写文章时什么时候分段？应该怎么分段呢？如果要写一个新的想法了，那就另起一段，对应在源码中空一行）；否则，可以用换行符（line breaking）来继续写原来的想法，对应在源码中使用 `\\` 或 `\newline`。

很多人都低估了合理分段的重要性。在 LaTeX 中，很多人甚至都不知道什么是分段，自己已经新起了一段都不知道。按照我的理解，一般情况下是不需要使用换行符的，在该分段的时候在代码里空一行另起一段就行了。但是在使用公式（equation）的时候需要考虑好该使用什么，这时候很容易犯上述的错误。下面是说明该换行还是该另起一段的三个正确示例：

```
% Example 1
\ldots when Einstein introduced his formula
\begin{equation}
    e = m \cdot c^2 \; ,
\end{equation}
which is at the same time the most widely known
and the least well understood physical formula.

% Example 2
\ldots from which follows Kirchhoff' s current law:
\begin{equation}
    \sum_{k=1}^{n} I_k = 0 \; .
\end{equation}

Kirchhoff' s voltage law can be derived \ldots

% Example 3
\ldots which has several advantages.
\begin{equation}
```

```
    I_D = I_F - I_R
\end{equation}
is the core of a very different transistor model. \ldots
```

2 断行和断页

2.1 合理分段

- `\\` 或 `\newline`：断行、不另起一段。`\\` 也在表格、公式等地方用于分行，而 `\newline` 只用于文本段落中。
- `\\*`：断行、不另起一段、不断页。
- `\newpage` 或 `\clearpage`：断页。二者有些细微的区别：一是在双栏排版中 `\newpage` 只起到另起一栏的作用；二是涉及到浮动体的排版上行为不同。
- `\linebreak[n]`、`\nolinebreak[n]`、`\pagebreak[n]`、`\nopagebreak[n]`：向 LaTeX 建议哪些地方适合断行、断页，哪些地方不适合断行、断页。n 是数字，代表适合/不适合的程度，取值范围为 0 到 4，默认为 4。数字越大代表程度越高。

一般 LaTeX 都会努力找到最适合断行的地方。但是有些时候——比如它不知道该如何用连字符分割单词的时候，它可能会让这一行文字从右边伸出这一段，然后报出 `overfull hbox` 的警告。这时使用 `\sloppy` 命令可以使单词之间的间距增大来避免这个问题，但是这时会报出 `underfull hbox` 的警告。大多数情况下这样排版出来的都不太好看。可以使用 `\fussy` 命令恢复 LaTeX 的默认方式。

3 连字符（Hyphenation）

对于绝大部分单词，LaTeX 都能够找到合适的断词位置，在断开的行尾加上连字符 -。如果一些单词没能自动断词，我们可以在单词内手动使用 `\-` 命令指定断词的位置。另外，也可以使用 `\hyphenation{word list}` 命令来指定使用连字符的位置，例如 `\hyphenation{FORTRAN Hy-phen-a-tion}`，其中的 `word list` 是不区分大小写的。

`\mbox{text}` 或 `\fbox{text}`：会避免 `text` 被连字符分开。`\fbox` 比 `\mbox` 多了个可见的框。

4 预定义好的字符串

- `\today`：2017 年 5 月 28 日（打印当天日期）
- `\TeX`：TeX
- `\LaTeX`：LaTeX
- `\LaTeXe`：LaTeX 2_ε

5 特殊符号

5.1 引号（Quotation Marks）

- 双引号：``...text...''
- 单引号：`...text...'

5.2 短划线（Dashes）和连字符（Hyphens）

在 LaTeX 中有下面四种横杠：

- -：-，连字符（hyphen），用于连接词语
- --：–，短破折号（en-dash），常用于连接数字表示起止范围
- ---：—，长破折号（em-dash），常用于表示意思的转换
- $-$：−，减号（minus sign）

5.3 波浪线（Tilde）

- \~{}：~
- $\sim$：∼

5.4 斜杠（Slash）

- read/write：read/write（不允许用连字符拆分）
- read\slash write：read/write（允许用连字符拆分）

5.5 度（Degree Symbol）

- $30\,^{\circ}\mathrm{C}$[1]：30 °C
- 30 \textcelsius{}：30 ℃
- 86 \textdegree{}F：86 °F

5.6 欧元符号

要想使用欧元符号，需要先在导言区通过 \usepackage{textcomp} 命令导入宏包，然后再使用 \texteuro 命令输出欧元符号。如果所用的字体不包含欧元符号或者想用别的字体的欧元符号，可以通过 \usepackage[official]{eurosym} 导入 eurosym 宏包，然后用 \euro 输出官方的欧元符号。用 gen 来替换 official 参数可以使用和当前字体匹配的欧元符号。

- \texteuro：€
- \euro：€

[1] 这里的 \, 会输出空格

5.7 省略号（Ellipsis）

LaTeX 提供了命令 `\ldots` 来生成省略号，相对于直接输入三个点的方式更为合理。`\ldots` 和 `\dots` 是两个等效的命令。

- `Apples, bananas, ...`: Apples, bananas, ...
- `Apples, bananas, \ldots`: Apples, bananas, …
- `Apples, bananas, \dots`: Apples, bananas, …

5.8 连字（Ligatures）

有些相邻的字母在排版时会连接起来，可以通过 `\mbox{}` 命令避免它们相连。

- `ffshfilfluffia`: ffshfilfluffia（相连的情况）
- `f\mbox{}fshf\mbox{}ilf\mbox{}luf\mbox{}f\mbox{}ia`: ffshfilfluffia（没有相连的情况）

5.9 重音（Accents）符号和特殊符号

示例代码如下：

```
\emph{\=a}  \emph{\'a}  \emph{\v a}  \emph{\` a}

H\^otel, na\"\i ve, \'el\`eve, \\
sm\o rrebr\o d, !'Se\~norita!, \\
Sch\"onbrunner Schlo\ss{}
Stra\ss e
```

示例输出结果如下：

$\bar{a}$ $\acute{a}$ $\check{a}$ $\grave{a}$

Hôtel, naïve, élève,
smørrebrød, ¡Señorita!,
Schönbrunner Schloß Straße

重音符号和特殊符号命令列表：

```
\`o   \'o   \^o   \~o
\=o   \.o   \"o   \c c
\u o  \v o  \H o  \c o
\d o  \b o  \t oo
\oe   \OE   \ae   \AE
\aa   \AA
\o    \O    \l    \L
\i    \j    !`    ?`
```

重音符号和特殊符号输出结果列表：

ò	ó	ô	õ
ō	ȯ	ö	ç
ǒ	ǒ	ő	ǫ
ọ	o̲	o͡o	
œ	Œ	æ	Æ
å	Å		
ø	Ø	ł	Ł
ı	ȷ	¡	¿

6 国际语言支持/中文排版支持

LaTeX 对其他很多语言提供了支持。babel 宏包可以用于对各种语言进行适配。其他语言暂时也用不到，这里就记录一下如何让 LaTeX 支持中文。

使用 LaTeX 排版中文有两种方式，一种是使用 xeCJK 宏包，另一种是使用 CTeX 宏包和文档类，推荐使用后者。CTeX 宏包和文档类是对 CJK 和 xeCJK 等宏包的进一步封装。文档类包括 ctexart、ctexrep、ctexbook，分别是对 LaTeX 的三个标准文档类 article、report、book 的封装，对 LaTeX 的排版样式做了许多调整，以切合中文排版风格。最新版本的 CTeX 宏包/文档类甚至支持自动配置字体。

6.1 CTeX 的安装

CTeX 宏集依赖的宏包和宏集已被最常见的 TeX 发行版 TeXLive 和 MiKTeX 所收录。如果本地安装的 TeXLive 或 MiKTeX 不是完全版本，就需要通过这两个发行版提供的宏包管理器来安装宏包。

TeXLive 的宏包管理器是 tlmgr。在 Linux 系统上，一般需要 sudo 权限才能正确地执行 tlmgr 的功能。

直接使用 sudo tlmgr [arg] 时，可能会提示找不到 tlmgr 或没有这个命令。乍一看，情况比较尴尬：不加 sudo 没有权限，加了 sudo 反而找不到命令了。经过上网搜索，在一个帖子[2] 里找到了解决方法。原来，sudo 有一种内置的保护机制，只会使用安全的环境变量 PATH。如果 TeXLive 的路径不在 sudo 的安全环境变量内，它就找不到相关的命令。可以在终端执行 sudo gedit /etc/sudoers，然后将 TeXLive 的路径添加到 sudo 的 secure_path 中。在我的 Ubuntu 16.04 上，添加后结果如下（后面的原有路径省略，不同路径用 : 隔开）：

```
Defaults secure_path="/usr/local/texlive/2016/bin/x86_64-linux:/usr/local/sbin:..."
```

不能用 sudo 执行 tlmgr 的问题解决后，在终端中依次执行以下命令，以更新 tlmgr 宏包管理器、已安装的所有宏包、安装 CTeX 宏集。

```
sudo tlmgr update --self
sudo tlmgr update --all
sudo tlmgr install ctex
```

[2] *sudo does not find tlmgr*, https://tex.stackexchange.com/questions/203874/sudo-does-not-find-tlmgr

6.2 使用 CTEX 文档类

CTEX 宏集提供了四个中文文档类：ctexart、ctexrep、ctexbook 和 ctexbeamer，分别对应 LATEX 的标准文档类 article、report、book 和 beamer。使用它们的时候，需要将涉及到的所有源文件使用 UTF-8 编码保存。

下面是使用 ctexart 文档类编写的一个例子：

```
\documentclass[UTF8]{ctexart}
\begin{document}
中文文档类测试。你需要将所有源文件保存为 UTF-8 编码。
你可以使用 XeLaTeX、LuaLaTeX 或 upLaTeX 编译，也可以使用 (pdf)LaTeX 编译。
推荐使用 XeLaTeX 或 LuaLaTeX 编译。
\end{document}
```

CTEX 预定义的字库中的中文字体已经基本够用，包括宋体（\songti）、**黑体**（\heiti）、楷书（\kaishu）、仿宋（\fangsong）等。更多 CTEX 的使用参考《CTEX 宏集手册》[3]。

7 单词之间的空格

为了使输出更美观、更具可读性，LATEX 可能会在不同单词之间或句子末尾插入更多空格。LATEX 默认句子以句点（periods）、问号（question marks）或者感叹号（exclamation marks）结尾。但是如果句点跟在一个大写字母后面，它不会认为这是句子结尾，因为大写字母后面跟句点往往是缩略词。

用户可以通过具体的命令来改变上面的默认设定。一个斜杠跟一个空格会产生一个不会被扩大的空格；一个波浪线（~）会产生一个既不能被扩大、也不能从这里断行的空格；在句点前使用 \@ 命令，不管这个句点是不是跟在大写字母后面，都会指定这个句子到句点就结束。使用 \frenchspacing 命令可以强制不在一个句子后面插入多余的空格。如果使用 \frenchspacing 命令就没必要再用 \@ 了。

代码示例：

```
Mr.~Smith was happy to see her\\
cf.~Fig.~5\\
I like BASIC\@. What about you?
```

示例输出：

Mr. Smith was happy to see her
cf. Fig. 5
I like BASIC. What about you?

[3] 《CTEX 宏集手册》，http://mirror.unl.edu/ctan/language/chinese/ctex/ctex.pdf

8 标题、章、节

文档类 article 中有以下几种分层次结构的命令：

```
\section{...}
\subsection{...}
\subsubsection{...}
\paragraph{...}
\subparagraph{...}
```

\part{...} 命令也可以把文档分为多个部分，但它不会影响 section 和 chapter 的编号。

和 article 文档类相比，在 report 和 book 中，可以使用 \chapter{...}。

由于 article 文档类中不包含 chapter，所以可以很方便地把 article 作为 chapter 插入 book 文档类。章节空隙、编号等由 LaTeX 自动完成。

下面是两个比较特殊的情况：

- \part 命令不会影响 chapter 或 section 的编号
- \appendix 命令没有任何参数，会把 chapter（对于 report、book）或 section（对于 article）的数字编号转换成字母编号。