

CCT 的 L^AT_EX 2_ε 中文文档类

张林波

2003 年 12 月 20 日

摘 要

`cctbase.sty`, `cctart.cls`, 和 `cctbook.cls` 是 1998–1999 年间为了一本书的排版和几个同事一起制做的宏包和文档类。由于它们是匆忙间编的, 因此当时并没有将它们正式发布, 而只是为了使用方便将它们包含在了 CCT 的发行文件中, 准备等将来有时间对它们进行规范化, 如改成 `docstrip` 格式以及对一些命令进行扩充, 然后再予以正式发布。然而之后的时间里, 除了对它们做过一些小修补外, 一直没有机会去系统地完善它们。后来邓建松等人的《L^AT_EX 2_ε 科技排版指南》一书中对这些文档类做了介绍, 使得许多人开始使用它们, 以致于逐渐变成了 CCT 事实上的标准文档。由于缺乏相关资料, 经常会有人问及有关它们的问题, 故此将当年编写这些文档类时写的一份说明草稿整理了一下, 重新用新 CCT 排版, 公布出来供大家参考。由于内容是 1999 年写的, 有些可能已经过时, 也没有介绍新版 CCT 的功能及新版 CCT 发布以来这些文档类的新改动, 因此请结合新版 CCT 手册 (<ftp://ftp.cc.ac.cn/pub/cct/README.pdf>) 阅读其中的内容。同时也希望有人帮助修改、完善本文档, 将对新版 CCT 的介绍包含进去 (本文档的排版源文件可从 <ftp://ftp.cc.ac.cn/pub/cct/CCTLaTeX.tex> 下载)。

目 录

§1 前 言	2
§2 <code>cctbase</code> 包	3
§3 <code>cctart</code> 文档类	10
§4 <code>cctbook</code> 文档类	15
§5 关于使用 L ^A T _E X 2 _ε 进行中文排版的一些建议	20

§1 前言

虽然 $\text{\LaTeX} 2_{\epsilon}$ 是为西文排版设计的, 但由于中西文排版有许多共性, 因此它的大部分功能也适用于中文排版。不过, 中西文排版的格式间毕竟存在很大的差异, 由于缺少适合中文科技排版的标准格式, 国内有关 \TeX 的书籍、资料也很少, 造成目前用户排版源文件的写法上比较混乱, 既降低了排版效率, 也大大影响了排版质量。为 $\text{\LaTeX} 2_{\epsilon}$ 设计一套较为灵活、适合于中文排版并且符合中文排版规范的文档类或包是当前亟待解决的一个问题。从事这项工作需要对中文排版的规范有全面深入地了解, 对 \TeX 排版语言和 $\text{\LaTeX} 2_{\epsilon}$ 系统的结构非常熟悉, 同时需要投入大量的时间和精力。我们目前所掌握的知识以及从精力上都还不具备完成这项工作的条件, 但我们希望能够尽快有人来完成这项工作。

本章介绍为中文排版设计的适用于 CCT 的两个文档类和一个包, 它们是分别从标准文档类 “book” 类和 “article” 类导出的 “cctbook” 类和 “cctart” 类, 以及这两个文档类中用到的 “cctbase” 包。这些包和文档类是在本书的编排过程中形成的, 目前尚不完善, 但我们希望它们能够解决部分用户在中文排版中的一些问题。介绍这些包和文档类的更重要的目的是想以此为例说明怎样修改 $\text{\LaTeX} 2_{\epsilon}$ 中的命令使之适合中文的排版习惯, 为大家提供一个使用 $\text{\LaTeX} 2_{\epsilon}$ 进行中文排版的框架, 供感兴趣的读者参考。

CCT 软件中提供过一组称为 \LaTeX 的格式文件 (包括 “carticle.sty”, “cbook.sty” 等等), 可用在 “\documentstyle” 命令中。这些文件是从 $\text{\LaTeX} 2.09$ 的格式文件 “article.sty”, “book.sty” 等修改而来的, 只适用于 $\text{\LaTeX} 2.09$ 的排版文档, 与 $\text{\LaTeX} 2_{\epsilon}$ 不兼容。此外, 这些格式中仅对有关字体大小的命令做了改动, 并且有些中西文字号大小的匹配也不是很合适, 我们建议用户不要再使用这些格式。目前的 CCT 版本中提供它们只是为了使用户能够处理以前用这些格式排版的文档, 将来它们有可能被从 CCT 系统中删除。

修改 $\text{\LaTeX} 2_{\epsilon}$ 的命令可以有两种做法。一种做法是将 $\text{\LaTeX} 2_{\epsilon}$ 的标准文件从系统目录中拷贝到自己的目录下, 然后进行修改。最好为改动后的文件另起一个文件名, 以免与原来的文件混淆。当所要修改的命令集中在一两个不太长的文件中时可以考虑这种做法 (例如上一段中所提到的 \LaTeX 中的格式文件便是如此产生的)。第二种做法是建立一个新文件, 将要修改的命令或定义汇集在该文件中, 包含新的命令、定义的文件可以根据情况做为一个包或一个新文档类来使用。任何情况下都不应该直接改动 $\text{\LaTeX} 2_{\epsilon}$ 系统中的文件, 因为这样做既违反了有关这些文件的版权声明, 又很容易造成使用上的混乱。至于如何找出哪条命令定义在 $\text{\LaTeX} 2_{\epsilon}$ 系统的哪个文件中, 在不同的操作系统下有不同的方法, 我们不在此讨论。

本章中所介绍的包和文档类的源文件均包含在 CCT 系统的最新发布版中。它们除了可以在排版中使用外, 也可在设计其它排版格式时起到参考的作用。

如果您对 \TeX 的原始排版命令有一个基本的了解, 例如阅读过 D. E. Knuth 的 “The \TeX Book” [1], 那么对本章内容的阅读会有一定的帮助。此外, [2] 也提供了许多有关编写 $\text{\LaTeX} 2_{\epsilon}$ 的文档类或包的有用资料。不过, 即使没有阅读过这些材料也没关系, 因为许多命令的含义可以从字面上猜出来。

我们在以下各节中将详细列出这些包和文档类的源文件并附以一些注释。在这些文件中行首标有 “%org” 的行以及行尾标有 “!!!” 的行用于注明我们做过修改的地方, 其中行首标有 “%org” 的行代表 $\text{\LaTeX} 2_{\epsilon}$ 的原始写法, 而行尾标有 “!!!” 的行则表示修改后的写法。

§2 cctbase 包

“cctbase”包是一个可在 L^AT_EX 2_ε 的标准文档类中使用的包,其中包含了一些与文档类无关、可供不同文档类共用的定义。cctbase 包除了被后面介绍的“cctbook”文档类和“cctart”文档类用到外,它也可以单独做为一个 L^AT_EX 2_ε 的包通过 `\usepackage` 命令在其它文档类中使用。

“cctbase”包提供了下述功能,这些功能应该对大部分 L^AT_EX 2_ε 的文档类有效:

- 改变正文中节名的组成方式,以使用户能方便地排出以“第一节”、“第二节”等方式编号的节标题。
- 保留章节中第一个段落的段头空,因为 L^AT_EX 2_ε 按照英文排版的习惯取消每个章、节中第一个段落的段头空。
- 在一些浮动体中(如由 `\caption` 命令排印的图或表的标题)插入 `\pushziti` 和 `\popziti` 命令,避免它们改变所处位置的汉字字体。
- 在 L^AT_EX 2_ε 选择字体的命令中插入汉字字体命令。例如,当用户键入命令 `\bfseries` 时汉字字体和西文字体同时变成黑体。
- 当用 L^AT_EX 2_ε 的命令改变西文的字号大小时,汉字的字号也同时改变。
- 允许用户用中文数字(即一、二、三等)排出计数器的值,可用于章节的编号中。
- 将 list 类环境中各项间的额外的纵向空设为 0,保持行距的均匀性。将 list 类环境的缩排距离改成一个或两个汉字的宽度。
- 将 `\date` 命令的缺省值改成 xxxx 年 xx 月 xx 日的形式。
- 将 `\baselinestretch` 宏的缺省值定义成 1.3 来适当放大 L^AT_EX 2_ε 中的行距。因为缺省行距对于中文排版来说略嫌小了一些。

下面我们逐段介绍 cctbase.sty 中所修改、增加的命令的作用及用法。

```
\NeedsTeXFormat{LaTeX2e}[1994/12/01]
\ProvidesPackage{cctbase}[1997/12/10 Custom definitions for CCT]
\RequirePackage{cchead}
```

cctbase 包的开始按 L^AT_EX 2_ε 的习惯给出一些声明。第一条命令表示 cctbase 包要求 1994 年 12 月 1 日以后发布的 L^AT_EX 2_ε 版本,如果系统不满足,则使用时用户会得到一个警告。第二条命令定义 cctbase 包的包名及发布日期。第三条命令则要求使用 cctbase 包时自动调入 cchead 包,即 CCT 系统的定义文件 cchead.sty,因此使用 cctbase 包的排版文档中不必再显式地调入 cchead.sty 文件。

```
\newcommand\chntoday{\the\year~年~\the\month~月~\the\day~日}
\date{\chntoday}
```

L^AT_EX 2_ε 中 `\today` 命令排出英文形式的当天日期。cctbase 包定义了一条命令 `\chntoday`，它以中文形式排出当天日期 (xxxx 年 xx 月 xx 日)，并将 `\date` 命令的缺省值改成中文形式的日期。

```
% \CS is equal to a Chinese inter-word space
\def\CS{\CC\ }
% \ccnospace suppresses spaces before and after the argument
\def\ccnospace#1{\leavevmode\unskip #1\ignorespaces}
```

命令 `\CS` 相当于两个汉字间的正常字间距，而 `\ccnospace` 命令则用于取消其参数前后的空。受 CCT 处理方式的限制，当在两个汉字中间插入其它命令时，两个字之间的距离会变大。通过适当地结合 `\ccnospace` 命令和 `\CS` 命令，可以保证汉字间的空的均匀性，在后面的一些定义中使用了这一技巧。

```
\newcommand\chnno@one[1]{%
  \ifcase#1 零\or 一\or 二\or 三\or 四\or 五\or 六\or 七\or 八\or 九\or 十%
  \else\@ctrerr\fi}

\def\chnno@two#1#2{%
  \ifnum #1=1\else\chnno@one{#1}\unskip\CS\fi 十\unskip
  \ifnum #2=0\else\CS\chnno@one{#2}\fi\unskip
}

\newcommand\chnno[1]{%
  \ifnum\arabic{#1}>99\@ctrerr\else
  \ifnum\arabic{#1}>10
    \expandafter\chnno@two\the\value{#1}\unskip
  \else
    \chnno@one{\arabic{#1}}\unskip
  \fi
\fi}
```

这里定义了一个命令 `\chnno`，它按中文方式排出所指定的计数器中的值，用法与 L^AT_EX 2_ε 的标准命令 `\arabic`，`roman` 等类似。例如，假设 `ctr` 是一个计数器并且其值为 15，则命令 `\chnno{ctr}` 的排版结果为“十五”。注意，上述定义的命令只能用于计数器的值在 0 到 99 之间的情况，但不难对上述命令进行补充使之适用于更大的数值。

```
%
% Change \@secCNTformat to use \sectionname, \subsectionname, etc.
```

```

%org \def\@secntformat#1{\csname the#1\endcsname\quad}
\renewcommand{\@secntformat}[1]{\csname #1name\endcsname\quad}%   !!!

%
% Change \@sect to use \sectionname, \subsectionname, etc., in the table of
% contents
\def\@sect#1#2#3#4#5#6[#7]#8{%
  \ifnum #2>\c@secnumdepth
    \let\@svsec\@empty
  \else
    \refstepcounter{#1}%
    \protected@edef\@svsec{\@secntformat{#1}\relax}%
  \fi
  \@tempskipa #5\relax
  \ifdim \@tempskipa>\z@
    \begingroup
      #6{%
        \@hangfrom{\hskip #3\relax\@svsec}%
        \interlinepenalty \@M #8\@par}%
      \endgroup
    \csname #1mark\endcsname{#7}%
    \addcontentsline{toc}{#1}{%
      \ifnum #2>\c@secnumdepth \else
%org    \protect\numberline{\csname the#1\endcsname}%
        \protect\numberline{\csname #1name\endcsname}%           !!!
      \fi
      #7}%
    \else
      \def\@svsechd{%
        #6{\hskip #3\relax
          \@svsec #8}%
        \csname #1mark\endcsname{#7}%
        \addcontentsline{toc}{#1}{%
          \ifnum #2>\c@secnumdepth \else
%org    \protect\numberline{\csname the#1\endcsname}%
            \protect\numberline{\csname #1name\endcsname}%           !!!
          \fi
          #7}%
        \fi
      \@xsect{#5}}

```

上述命令对 $\text{\LaTeX} 2_{\epsilon}$ 中节单元 (节、小节等) 标题的排版格式进行修改。

$\text{\LaTeX} 2_{\epsilon}$ 的标准文档类采用与节单元名称同名的计数器来为节单元编号, 如用 “section” 为节编号, 用 “subsection” 为小节编号等。在排版节单元的标题以及构造书眉、目录中的内容时, $\text{\LaTeX} 2_{\epsilon}$ 调用相应的 “\the...” 形式命令来排出节单元的编号。以节为例, 标准文档类通常将命令 “\section{Introduction}” 排版为 “\thesection\quad Introduction”, 因为 “\thesection” 的定义为 “\arabic{section}”, 因此当计数器 “section” 的值为 2 时, 排版结果为 “**2 Introduction**”, 这样排出的节标题符合英文排版习惯。

在中文排版中, 节号往往带有节单元的名称或某些特殊符号, 而不是仅由数字组成。中文排版中常见的节号的排法有: “第二节”, “§2”, “二、” 等等。当使用 $\text{\LaTeX} 2_{\epsilon}$ 的标准文档类

时, 我们可以通过重新定义 `\the...` 命令来改变节单元编号的排版格式使之符合中文排版的习惯。如将 `\thesection` 命令的定义改为 “`\S\arabic{section}`” 来排出形如 “§1”, “§2” 形式的节号。

上一段中的做法虽然可以控制节单元编号的形式, 但因为 $\text{\LaTeX} 2_{\epsilon}$ 在其它许多地方也使用 `\the...` 命令, 因而它可能在一些意想不到的地方造成排版格式的破坏。例如, 一些文档类中将排公式号的命令 `\theequation` 定义成 `\thesection.\arabic{equation}`, 如果 `\thesection` 命令的定义为 `\arabic{section}`, 则第一节中的公式号为 “1.1”, “1.2” 等, 第二节中的公式号为 “2.1”, “2.2”, 但如果 `\thesection` 命令的定义为 `\S\arabic{section}`, 则公式号将被排成 “§1.1”, “§1.2” 等等, 显然不符合通常习惯。

为了能够方便地排版出各种形式的中文节单元编号, 又尽可能避免破坏节单元编号在其它地方的排版格式, `cctbase` 包中对节单元的标题的构造方式进行了重新组织, 即在构造节单元的标题时调用 `\...name` 形式的命令 (而不用 `\the...` 形式的命令) 来排出节单元的编号, 其中 “...” 为节单元计数器的名称 (如 `section`, `subsection` 等)。这样用户可以通过改变 `\...name` 形式的命令 (如 `\sectionname`) 来控制节单元编号的排版格式而不必改动 `\the...` 形式的命令 (`\thesection`), 以避免破坏其它引用节单元编号的地方的排版格式。例如, 使用下述命令:

```
\renewcommand{\sectionname}{第\ccnospace{\CS\chnno{section}\CS}节}
\section{中文排版的特点}}
```

排出的节标题为 (假定 `section` 的值为 2): “第二节 中文排版的特点”。

这里所介绍的采用 `\sectionname` 命令控制节标题排版的做法类似于 $\text{\LaTeX} 2_{\epsilon}$ 中对章的编号形式的处理, 但也有区别。 $\text{\LaTeX} 2_{\epsilon}$ 使用命令 `\chaptername` 来定义章名, 并同时调用 `\chaptername` 和 `\thechapter` 命令来构造章标题, 通常形式为 “`\chaptername~\thechapter`”, 当 `chaptername` 命令被定义成 “Chapter” 时, 得到的章号形为 “Chapter 1”, “Chapter 2” 等。而 `cctbase` 要求 `\sectionname` 命令同时给出节名与节号, `cctbase` 在构造节标题时只调用 `\sectionname` 命令, 而不再调用 `\thesection` 命令。

除此之外, 在文档类 `cctart` 和 `cctbook` 中还提供了 `\...format` 形式的命令用来控制节单元的标题的排版位置 (居中或顶左), 请参看后续节中的说明。

```
%
% Change \afterheading to leave \parindent after heading
\def\@afterheading{%
  \@nobraektrue
  \everypar{%
    \if@nobraek
      \@nobraekfalse
      \clubpenalty \@M
%org \if@afterindent \else
%org   {\setbox\z@\lastbox}%
%org \fi
    \else
      \clubpenalty \@clubpenalty
```

```
\everypar{}%
\fi}}
```

这里修改了 L^AT_EX 2_ε 的内部命令 `\@afterheading`, L^AT_EX 2_ε 在章节命令中调用该命令来取消每章节第一段的段头空。修改后的命令将保留第一段的段头空。

```
% Change default font size and style in \caption.
% Insert \pushziti and \popziti to avoid changing style of Chinese chars.
% (based on article/book classes).
\long\def\@makecaption#1#2{%
  \vskip\abovecaptionskip
%org \sbox\@tempboxa{#1: #2}%
  \sbox\@tempboxa{\pushziti\small #1: #2\popziti}%      !!!
  \ifdim \wd\@tempboxa >\hsize
%org #1: #2\par
    {\pushziti\small #1: #2\popziti\par}%              !!!
  \else
    \global \@minipagefalse
    \hbox to\hsize{\hfil\box\@tempboxa\hfil}%
  \fi
  \vskip\belowcaptionskip}
```

上述命令将 `\caption` 中的缺省字号改成小 5 号字, 同时插入 CCT 的 `\pushziti` 和 `\popziti` 命令, 以避免 `\caption` 命令改变其周围的中文字体。

```
%
% Match Chinese typefaces to LaTeX typefaces
\newcommand\@matchccfont[3]{%
  \let#2#1\def#1{#2#3}
}

% Use \heiti for \bfseries
\@matchccfont{\bfseries}{\obfseries}{\heiti}

% Use \songti for \rmfamily, \mdseries, \upshape
\@matchccfont{\rmfamily}{\ormfamily}{\songti}
\@matchccfont{\mdseries}{\omdseries}{\songti}
\@matchccfont{\upshape}{\oupshape}{\songti}

% Use \kaishu for \itshape
\@matchccfont{\itshape}{\oitshape}{\kaishu}

% Use \fangsong for \slshape, \sffamily
\@matchccfont{\slshape}{\oslshape}{\fangsong}
\@matchccfont{\sffamily}{\osffamily}{\fangsong}
```

上面一段命令在一些英文字体命令中加入相应的改变中文字体的命令,使它们在改变英文字体的同时也相应地改变中文字体。如 `\rmfamily` 命令将中文字体变成宋体, `\bfseries` 命令将中文字体变成黑体,等等。

```
%
% Map Chinese and LaTeX2e font sizes by redefining
% \set@fontsize{baselinestretch}{point size}{baselineskip}
%
% We also set \parindent=2\ccwd if current parindent is not zero.
\let\oset@fontsize=\set@fontsize
\def\set@fontsize#1#2#3{%
  \oset@fontsize{#1}{#2}{#3}%
  \ifdim #2pt < 6.25pt \zihao{7}\else
    \ifdim #2pt < 8pt \zihao{6}\else
      \ifdim #2pt < 9.2pt \zihao{-5}\else
        \ifdim #2pt < 10.61pt \zihao{5}\else
          \ifdim #2pt < 12.16pt \zihao{-4}\else
            \ifdim #2pt < 14pt \zihao{4}\else
              \ifdim #2pt < 17.5pt \zihao{3}\else
                \ifdim #2pt < 23pt \zihao{2}\else
                  \ifdim #2pt < 30pt \zihao{1}\else
                    \zihao{0}\relax
  \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
  \ifdim\parindent=0pt\else\parindent 2\ccwd\fi
}
```

这里我们对 `\set@fontsize` 命令进行修改,使它在设定英文字号大小的同时也选择一个与之相匹配的中文字号。由于 \LaTeX 2_ϵ 中所有改变字号的命令包括 `\large`, `\small`, `\normalsize` 等内部都是调用 `\set@fontsize` 命令来完成的,因此,对它的修改实际上对 \LaTeX e 中的所有字号命令都起作用。

```
% Change definition of \numberline to allow wider section/chapter labels in
% \tableofcontents
%org \def\numberline#1{\hb@xt@\@tempdima{#1\hfil}}
\renewcommand\numberline[1]{\ifx\relax#1\relax\else#1\quad\fi}% !!!
```

\LaTeX 2_ϵ 在排印目录时为章节的编号预留出固定的排版空间,我们发现当章节的编号超宽时(这在中文排版中经常会碰到,而在西文排版中却不多见,因为西文的章节编号形式与中文相比较为简单)排版的结果不对。我们在这里所做的修改是将章节编号的排版宽度改成可变的,而在编号与章节的标题间留出固定的距离。我们不清楚这样做是否符合中文排版规范,不过很容易为目录中的章节编号定义其它的排版形式。

```
%===== suppress extra inter-line spacing in \list environments
```



```

\setlength\partopsep{0pt}
\renewcommand\@listI{\leftmargin\leftmarginii
  \parsep 0pt
  \topsep \parsep
  \itemsep \parsep}

\@listI
\renewcommand\@listii {\leftmargin\leftmarginii
  \labelwidth\leftmarginii
  \advance\labelwidth-\labelsep
  \parsep \z@ \@plus\z@ \@minus\z@
  \topsep \parsep
  \itemsep \parsep}

\renewcommand\@listiii{\leftmargin\leftmarginiii
  \labelwidth\leftmarginiii
  \advance\labelwidth-\labelsep
  \parsep \z@ \@plus\z@ \@minus\z@
  \topsep \parsep
  \itemsep \parsep}

\renewcommand\@listiv{\leftmargin\leftmarginiv
  \labelwidth\leftmarginiv
  \advance\labelwidth-\labelsep
  \parsep \z@ \@plus\z@ \@minus\z@
  \topsep \parsep
  \itemsep \parsep}

\renewcommand\@listv{\leftmargin\leftmarginv
  \labelwidth\leftmarginv
  \advance\labelwidth-\labelsep
  \parsep \z@ \@plus\z@ \@minus\z@
  \topsep \parsep
  \itemsep \parsep}

\renewcommand\@listvi{\leftmargin\leftmarginvi
  \labelwidth\leftmarginvi
  \advance\labelwidth-\labelsep
  \parsep \z@ \@plus\z@ \@minus\z@
  \topsep \parsep
  \itemsep \parsep}

```

$\text{\LaTeX} 2_{\epsilon}$ 在 `list` 类环境的前后以及各项之间会留出一定的空，而我们遇到的中文编辑则往往要求我们保持行距的均匀性。这里所列出的命令将 `list` 类环境的前后以及各项之间的额外的空全部设置成 0。如果您不喜欢这种排版方式，可将这一段命令变成说明，从而保留 $\text{\LaTeX} 2_{\epsilon}$ 原来的排法。

```

%
% Change default margins for \list environments
\setlength\leftmargini {2\ccwd}
\setlength\leftmarginii {\leftmargini}
\setlength\leftmarginiii {\leftmargini}
\setlength\leftmarginiv {\leftmargini}
\setlength\leftmarginv {\ccwd}
\setlength\leftmarginvi {\ccwd}
\setlength\leftmargin {\leftmargini}

```

```

\setlength\labelsep      {.5\ccwd}
\setlength\labelwidth    {\leftmargini}

%
\setlength\listparindent{2\ccwd}
% Change \listparindent to 2\ccwd in \list
\renewcommand\list[2]{\ifnum \@listdepth >5\relax \@toodeep
  \else \global\advance\@listdepth\@ne \fi
  \rightmargin \z@ \listparindent2\ccwd \itemindent \z@
  \csname @list\romannumeral\the\@listdepth\endcsname
  \def\@itemlabel{#1}\let\makelabel\@mklab \@nmbolistfalse #2\relax
  \@trivlist
  \parskip\parsep \parindent\listparindent
  \advance\linewidth -\rightmargin \advance\linewidth -\leftmargin
  \advance\@totalleftmargin \leftmargin
  \parshape \@ne \@totalleftmargin \linewidth
  \ignorespaces}

```

上述命令将 list 类环境中左右缩排的距离改成一个或两个汉字的宽度, 以便与正文中其它部分更好地对齐。

```

%
% Use \bfseries in the title
\let\oldtitle=\title
\def\title#1{\oldtitle{\bfseries\biaosong #1}}

```

将标题 (`\title` 命令) 中的西文字体改成黑体, 中文字体改成标宋。排中文标题时这样排版似乎比 L^AT_EX 2_ε 缺省使用的罗马体/宋体更好看一些。这仅仅只是一个人口味问题。

```

\renewcommand{\baselinestretch}{1.3}\normalsize

\endinput

```

cctbase 包的最后将 `\baselinestretch` 宏定义成 1.3, 即将行距设成 L^AT_EX 2_ε 的标准行距的 1.3 倍, 适合于中文排版。

§3 cctart 文档类

cctart 是从 article 文档类导出的一个文档类, 它的使用方法与 article 类完全一样, 适合于文章的排版。与 article 类相比, cctart 类主要在排版格式上针对中文排版的特点进行了相应的修改, 使排版结果更接近中文排版的习惯。cctart 文档类的源文件名为 “cctart.cls”。

```

%
% 'cctart' class derived from the 'article' class.

\NeedsTeXFormat{LaTeX2e}[1994/12/01]
\ProvidesClass{cctart}[1997/12/10 Derived from the 'article' class]
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
\ProcessOptions
\LoadClass[a4paper]{article}[1996/10/31]
\RequirePackage{cctbase}[1997/12/10]

```

按照 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 文档类的标准写法, `cctart` 类的开头指明所需要的 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 格式文件的发布日期, 定义文档类的名称, 调入标准文档类 “`article`” 并将所有可选项传递给它 (注意我们在调入 `article` 类时使用了 “`a4paper`” 选项, 并调入 `cctbase` 包。

```

%
% Redefine some names
\renewcommand\contentsname{目 录}
\renewcommand\listfigurename{插图目录}
\renewcommand\listtablename{表格目录}
\renewcommand\refname{参考文献}
\renewcommand\indexname{索 引}
\renewcommand\figurename{\rm 图}
\renewcommand\tablename{\bf 表}
\renewcommand\abstractname{摘 要}
\renewcommand\partname{第\ccnospace{\CS\chnno{part}}\CS}部分}
\renewcommand\appendixname{附录\ccnospace{\CS\chnno{chapter}}}

```

这几条命令修改部分、图表等的编号方式以及目录、索引等的标题。 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 浮动体中的 `\caption` 命令用 “`\figurename~\thefigure`” 的形式排出图号, 用 “`\tablename~\thetable`” 的形式排出表号, 缺省情况下, `\figurename` 定义成 “Figure”, `\tablename` 定义成 “Table”, 因此用 `\caption` 命令得到的图号形式为 “Figure 1”, 而表号形式则为 “Table 1”. 我们将 `\figurename` 和 `\tablename` 的定义分别改成 “图” 和 “表”, 这样便可产生 “图 1” 和 “表 1” 形式的图表编号。类似地, 我们将目录和索引的标题分别定义成 “目 录” 和 “索 引”, 等等。

```

% headings
\if@twoside
  \def\ps@headings{%
    \let\@oddfoot\@empty\let\@evenfoot\@empty
%org \def\@evenhead{\thepage\hfil\slshape\leftmark}%
%org \def\@oddhead{\slshape\rightmark\hfil\thepage}%
    \def\@evenhead{\pushziti\thepage\hfil\slshape\leftmark\popziti}% !!!
    \def\@oddhead{\pushziti\slshape\rightmark\hfil\thepage\popziti}% !!!
    \let\@mkboth\markboth
    \def\sectionmark##1{%
%org \markboth {\MakeUppercase{%
    \markboth {%

```

```

        \ifnum \c@secnumdepth >\z@
%org      \thesection\quad
          \sectionname\ \ %          !!!
        \fi
%org      ##1}}{}}%
          ##1}}{}}%          !!!
        \def\subsectionmark##1{%
          \markright {%
            \ifnum \c@secnumdepth >\@ne
%org      \thesubsection\quad
              \subsectionname\ \ %          !!!
            \fi
            ##1}}}}
\else
  \def\ps@headings{%
    \let\@oddfont\@empty
%org  \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
    \def\@oddhead{\pushziti{\slshape\rightmark}\hfil\thepage\popziti}% !!!
    \let\@mkboth\markboth
    \def\sectionmark##1{%
%org  \markright {\MakeUppercase{%
      \markright {%          !!!
        \ifnum \c@secnumdepth >\m@ne
%org      \thesection\quad
          \sectionname\ \ %          !!!
        \fi
%org      ##1}}}}
          ##1}}}}          !!!
    \fi
    \ps@headings%          !!!

```

这里对页眉的排法做了三方面的修改, 第一是在构造页眉的内容时使用 `\...name` 形式的命令以便与节单元的标题的构造一致, 第二是在页眉中插入 `\pushziti` 和 `\popziti` 命令以保护紧跟在页眉后面的正文的字体, 第三则是在页眉构造中去掉了 `\MakeUppercase` 命令, 因为对包含汉字的部分进行字母大小写转换会破坏其中的汉字。

```

\def\@part[#1]#2{%
  \ifnum \c@secnumdepth >\m@ne
    \refstepcounter{part}%
%org  \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
    \addcontentsline{toc}{part}{\partname\hspace{1em}#1}%          !!!
  \else
    \addcontentsline{toc}{part}{#1}%
  \fi
  {\parindent \z@ \raggedright
   \interlinepenalty \@M
   \normalfont
   \ifnum \c@secnumdepth >\m@ne
%org  \Large\bfseries \partname~\thepart
     \Large\bfseries \partname \quad%          !!!
   \fi

```

```

%org \huge \bfseries #2%
  \Large \bfseries #2%          !!!
  \markboth{}{\}\par}%
  \nobreak
  \vskip 3ex
  \@afterheading}%
\def\@spart#1{%
  {\parindent \z@ \raggedright
  \interlinepenalty \@M
  \normalfont
%org \huge \bfseries #1\par}%
  \Large \bfseries #1\par}%    !!!
  \nobreak
  \vskip 3ex
  \@afterheading}%

```

$\text{\LaTeX} 2_{\epsilon}$ 中“部分”是一个比较特殊的章节单元。上述命令对“部分”的编号及其标题的排法做了一些改动，以便于排出符合中文习惯的标题。 $\text{\LaTeX} 2_{\epsilon}$ 中原来可以用 `\partname` 命令定义“部分”的名称，但 `\partname` 中只包含“部分”的名称，不包含部分的编号， $\text{\LaTeX} 2_{\epsilon}$ 在构造“部分”的标题时对 `\partname` 与 `\thepart` 进行组合来为“部分”编号（如“Part I”，“Part II”等）。经改动后的 `\part` 命令在构造“部分”的标题时假定 `\partname` 中既包含了“部分”的名称，也包含了“部分”的编号，而不再调用 `\thepart` 命令，这样便与 `cctbase` 中节单元标题的构造方法统一起来，而且便于定义各种中文习惯的编号方式，如前面所介绍的命令中定义的 `\partname`，它使得“部分”的编号采用“第一部分”的形式。

```

\renewcommand\section{\@startsection
{section}{1}{\z@}% {name}{depth}{
  {3.5ex \@plus1ex \@minus .2ex}% leave parindent
  {2.3ex \@plus.2ex}%
  {\reset@font\large\bfseries\sectionformat}}
\newcommand\sectionname{\S\arabic{section}}
\newcommand\sectionformat{\centering}

\renewcommand\subsection{\@startsection
{subsection}{2}{\z@}% {name}{depth}{
  {1.75ex \@plus.5ex \@minus .1ex}% leave parindent
  {1.15ex \@plus.1ex}%
  {\reset@font\normalsize\bfseries\subsectionformat}}
\newcommand\subsectionname{\S\arabic{section}.\arabic{subsection}}
\newcommand\subsectionformat{}

\renewcommand\subsubsection{\@startsection
{subsubsection}{3}{\z@}% {name}{depth}{
  {1.75ex \@plus 0.5ex \@minus .1ex}% positive->leave parindent
  {1.15ex \@plus.1ex}%
  {\reset@font\normalsize\bfseries\subsubsectionformat}}
\newcommand\subsubsectionname{\S\arabic{section}.\arabic{subsection}.\%
\arabic{subsubsection}}
\newcommand\subsubsectionformat{}

```

上述命令重新定义了 `cctart` 文档类中的节单元以及它们的编号方式。值得指出的是节单元命令中引用了 “`\...format`” 形式的命令, 如 “`\sectionformat`”, “`\subsectionformat`” 等, 这是 $\text{\LaTeX} 2_{\epsilon}$ 原来没有的。引用 “`\...format`” 形式命令的目的是为了方便用户控制节单元的标题的排版格式。当 “`\...format`” 为空命令时, 相应的节单元的标题项左排版, 要使节单元标题居中排版, 只需将对应的 “`\...format`” 命令定义成 “`\centering`” 即可, 如:

```
\renewcommand{\subsectionformat}{\centering}
```

即可使得小节的标题居中排版。

```
% abstract
\renewenvironment{abstract}{%
  \leftskip=2\ccwd \rightskip=2\ccwd
  \small\parindent2\ccwd
  \bigskip
  \begin{center}%
    {\bfseries \abstractname\vspace{\z@}}%
  \end{center}%
  \par\nopagebreak\medskip\nopagebreak}
{\par\bigskip}
```

上述命令重新定义了摘要的排版格式。 $\text{\LaTeX} 2_{\epsilon}$ 采用 `quote` 环境来排版摘要, 因而无法保留摘要中的段头空。这里进行修改的目的—是为了保留段头空, 二是为了选择小 5 号字及适当的左右缩排距离。

```
%
% Change definition of 'thebibliography' environment
\renewenvironment{thebibliography}[1]
  {\section*{\refname
%org      \@mkboth{\MakeUppercase\refname}{\MakeUppercase\refname}}%
    \@mkboth{\refname}{\refname}}%          !!!
  \list{\@biblabel{\@arabic\c@enumiv}}%
    {\settowidth\labelwidth{\@biblabel{#1}}%
    \leftmargin\labelwidth
    \advance\leftmargin\labelsep
    \@openbib@code
    \usecounter{enumiv}%
    \let\p@enumiv\@empty
    \renewcommand\theenumiv{\@arabic\c@enumiv}}%
  \sloppy
  \small%          !!!
  \clubpenalty4000
  \@clubpenalty \clubpenalty
  \widowpenalty4000%
  \sfcode'\.\@m}
{\def\@noitemerr
  {\@latex@warning{Empty 'thebibliography' environment}}}
```

```

\endlist}

% set default \pagestyle to 'plain'
\pagestyle{plain}

\endinput

```

cctart 文档类的最后对 thebibliography 环境进行了修改, 去掉了构造页眉的命令中的 “\MakeUppercase” 命令, 将环境中的缺省字号改成小 5 号字 (\small), 并将缺省的 \pagestyle 设成 “plain”。

§4 cctbook 文档类

cctbook 是从 book 文档类导出的一个文档类, 它的使用方法与 book 类完全一样, 适合于书的排版。与 cctart 类类似, cctbook 文档类的源文件名为 “cctbook.cls”。

```

%
% 'cctbook' class derived from the 'book' class.

\NeedsTeXFormat{LaTeX2e}[1994/12/01]
\ProvidesClass{cctbook}[1997/12/10 Derived from the 'book' class]
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{book}}
\ProcessOptions
\LoadClass[a4paper]{book}[1996/10/31]
\RequirePackage{cctbase}[1997/12/10]

```

这是 L^AT_EX 2_ε 文档类开头的标准书写格式, 其含义可参看上节中对 cctart 类的有关说明。

```

%
% Redefine names+
\renewcommand\contentsname{目 录}
\renewcommand\listfigurename{插图目录}
\renewcommand\listtablename{表格目录}
\renewcommand\bibname{参考文献}
\renewcommand\indexname{索 引}
\renewcommand\figurename{\rm 图}
\renewcommand\tablename{\bf 表}
\renewcommand\partname{第}\ccnospace{\CS\chnno{part}}\CS}部分
\renewcommand\chaptername{第}\ccnospace{\CS\chnno{chapter}}\CS}章
\renewcommand\appendixname{附录}\ccnospace{\CS\chnno{chapter}}

```

将一些名称的定义改成中文。请参看上节中对 cctart 类的有关说明。

```

% Change format of the running heads. Insert \pushziti and \popziti
\if@twoside
  \def\ps@headings{%
    \let\@oddfoot\@empty\let\@evenfoot\@empty
%org \def\@evenhead{\thepage\hfil\slshape\leftmark}%
%org \def\@oddhead{\slshape\rightmark}\hfil\thepage}%
    \def\@evenhead{\pushziti\thepage\hfil\slshape\leftmark\popziti}% !!!
    \def\@oddhead{\pushziti\slshape\rightmark}\hfil\thepage\popziti}% !!!
    \let\@mkboth\markboth
  \def\chaptermark##1{%
%org \markboth {\MakeUppercase{%
  \markboth {%
    \ifnum \c@secnumdepth >\m@ne
    \if@mainmatter
%org \@chapapp\ \thechapter. \ %
    \chaptername \ \ %
    \fi
    \fi
%org ##1}}{}}%
    ##1}}{}}%
  \def\sectionmark##1{%
%org \markright {\MakeUppercase{%
  \markright {%
    \ifnum \c@secnumdepth >\z@
%org \thesection. \ %
    \sectionname \ \ %
    \fi
%org ##1}}}}
    ##1}}}}%
  \else
    \renewcommand\ps@headings{%
      \let\@oddfoot\@empty
%org \def\@oddhead{\slshape\rightmark}\hfil\thepage}%
      \def\@oddhead{\pushziti\slshape\rightmark}\hfil\thepage\popziti}% !!!
      \let\@mkboth\markboth
      \def\chaptermark##1{%
%org \markright {\MakeUppercase{%
        \markright {%
          \ifnum \c@secnumdepth >\m@ne
            \if@mainmatter
%org \@chapapp\ \thechapter. \ %
            \chaptername \ \ %
            \fi
            \fi
%org ##1}}}}
          ##1}}}}%
        \fi
      \ps@headings

```

改变页眉的构造方式。请参看上节中对 cctart 类的有关说明。

```
% Define part
```



```

\def\@part[#1]#2{%
  \thispagestyle{empty}%
  \ifnum \c@secnumdepth >-2\relax
    \refstepcounter{part}%
%org \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
    \addcontentsline{toc}{part}{\partname\hspace{1em}#1}% !!!
  \else
    \addcontentsline{toc}{part}{#1}%
  \fi
  \markboth{}{}%
  {\centering
  \interlinepenalty \@M
  \normalfont
  \ifnum \c@secnumdepth >-2\relax
%org \huge\bfseries \partname~\thepart
    \Huge\bfseries \partname% !!!
  \par
  \vskip 20\p@
  \fi
  \Huge\bfseries #2\par}%
\@endpart}

```

重新定义“部分”的排版风格。请参看上节中对 `cctart` 类的有关说明。

```

% Define chapter
\def\@chapter[#1]#2{\ifnum \c@secnumdepth >\m@ne
  \if@mainmatter
    \refstepcounter{chapter}%
    \typeout{\@chapapp \space \thechapter.}%
    \addcontentsline{toc}{chapter}%
%org
    {\protect\numberline{\thechapter}#1}%
    {\protect\numberline{\chaptername}#1}% !!!
  \else
    \addcontentsline{toc}{chapter}{#1}%
  \fi
  \else
    \addcontentsline{toc}{chapter}{#1}%
  \fi
  \chaptermark{#1}%
  \addtocontents{lof}{\protect\addvspace{10\p@}}%
  \addtocontents{lot}{\protect\addvspace{10\p@}}%
  \if@twocolumn
    \@topnewpage[\@makechapterhead{#2}]%
  \else
    \@makechapterhead{#2}%
    \@afterheading
  \fi}
\def\@chapapp{Chapter}%
\def\@makechapterhead#1{%
%org \vspace*{50\p@}%
%org {\parindent \z@ \raggedright \normalfont
%org \ifnum \c@secnumdepth >\m@ne

```

```

%org      \if@mainmatter
%org      \huge\bfseries \@chapapp\space \thechapter
%org      \par\nobreak
%org      \vskip 20\p@
%org      \fi
%org      \fi
%org      \interlinepenalty\@M
%org      \Huge \bfseries #1\par\nobreak
%org      \vskip 40\p@
      \vspace*{-\headsep}\vspace*{-\headheight}\vspace*{15\p@}%      !!!
      { \normalfont \centering%      !!!
        \ifnum \c@secnumdepth >\m@ne%      !!!
          \if@mainmatter%      !!!
            \Large\bfseries \chaptername \quad #1 \par\nobreak%      !!!
          \else%      !!!
            \Large\bfseries #1 \par\nobreak%      !!!
          \fi%      !!!
        \fi%      !!!
        \vskip 15\p@%      !!!
      }}
\def\@makeschapterhead#1{%
%org \vspace*{50\p@}%
%org {\parindent \z@ \raggedright
%org \normalfont
%org \interlinepenalty\@M
%org \Huge \bfseries #1\par\nobreak
%org \vskip 40\p@
      \vspace*{-\headsep}\vspace*{-\headheight}\vspace*{15\p@}%      !!!
      { \normalfont \centering%      !!!
        \interlinepenalty\@M%      !!!
        \Large \bfseries #1\par\nobreak%      !!!
        \vskip 15\p@%      !!!
      }}
}

```

修改“章”命令。这里所做的修改与对“部分”命令所做的修改完全类似，即将“章”的标题的构成改成只依赖于“\chaptername”，不依赖于“\thechapter”，以便与其它节单元的名称的处理方式统一。

```

% Change definition of section
\renewcommand\section{\@startsection
{section}{2}{\z@}% {name}{depth}{
  {3.5ex \@plus 1ex \@minus .2ex}% positive->leave parindent
  {2.3ex \@plus .2ex}%
  {\reset@font\large\bfseries\sectionformat}}
\newcommand\sectionname{\S\thechapter.\arabic{section}}
\newcommand\sectionformat{\centering}

% Change definition of \subsection
\renewcommand\subsection{\@startsection
{subsection}{3}{\z@}% {name}{depth}{
  {1.75ex \@plus 0.5ex \@minus .1ex}% positive->leave parindent
}

```

```

        {1.15ex \@plus.1ex}%
        {\reset@font\normalsize\bfseries\subsectionformat}}
\newcommand\subsectionname{\S\thechapter.\arabic{section}.\arabic{subsection}}
\newcommand\subsectionformat{}

% Change definition of \subsubsection
\renewcommand\subsubsection{\@startsection
{subsubsection}{4}{\z@}%   {name}{depth}{}
{1.75ex \@plus 0.5ex \@minus .1ex}% positive->leave parindent
{1.15ex \@plus.1ex}%
{\reset@font\normalsize\bfseries\subsubsectionformat}}
\newcommand\subsubsectionname{\S\thechapter.\arabic{section}.\%
\arabic{subsection}.\arabic{subsubsection}}
\newcommand\subsubsectionformat{}

% Set maxi. depth of numbered sections to 4 (\subsubsection).
\setcounter{secnumdepth}{4}

```

重新定义各级节单元。参看上节关于 `cctart` 文档类的说明。

```

%
% Change definition of \appendix
\renewcommand\appendix{\par
\setcounter{chapter}{0}%
\setcounter{section}{0}%
\renewcommand\@chapapp{Appendix}
\renewcommand\thechapter{\Alph{chapter}}%
\renewcommand\chaptername{\appendixname}}%

```

修改 `\appendix` 命令, 使得它之后的章编号变成“附录一”, “附录二”, 等等, 而公式、图表等的编号则变成“A.1”, “A.2”, 等等。

```

% Rewrite theindex environment (removing \MakeUppercase, use 'multicol' )
\RequirePackage{multicol}
\renewenvironment{theindex}
    {\chapter{\indexname}
     \vskip 15pt
     \begin{multicols}{2}%
     \let\item\@idxitem
     \end{multicols}}

```

这里我们重写了 `theindex` 环境, 重写后的环境利用 ‘`multicol`’ 包来排出双栏的索引。用 `multicol` 包进行双栏排版时, 可以在同一页中进行单栏和多栏间的切换, 同时当双栏内容排不满一页时两栏被切成一样高。

```

% Change definition of \thebibliography environment to use smaller font.
\renewenvironment{thebibliography}[1]
%org    {\chapter*{\bibname
%org    \mkboth{\MakeUppercase\bibname}{\MakeUppercase\bibname}}%
{\def\chaptername{\chapter{\bibname}}%          !!!
\list{\@biblabel{\@arabic\c@enumiv}}%
{\settowidth\labelwidth{\@biblabel{#1}}%
\leftmargin\labelwidth
\advance\leftmargin\labelsep
\@openbib@code
\usecounter{enumiv}%
\let\p@enumiv\@empty
\renewcommand\theenumiv{\@arabic\c@enumiv}}%
\small%          !!!
\sloppy
\clubpenalty4000
\@clubpenalty \clubpenalty
\widowpenalty4000%
\sffcode'\.\@m}
{\def\@noitemerr
{\@latex@warning{Empty 'thebibliography' environment}}%
\endlist}

\endinput

```

该处对 thebibliography 环境所做的改动与 cctart 文档类中完全类似。可参看前一节中的有关说明。

§5 关于使用 L^AT_EX 2_ε 进行中文排版的一些建议

T_EX 系统提供了强大的宏命令功能, 这些功能有助于使用者根据需要建立自己的宏命令库, 这样可以简化排版过程, 也使得排版结果更为规范、一致。L^AT_EX 2_ε 系统就是一个庞大的宏命令库, 它为一般科技文献的排版提供了一个很好的框架。

目前科技书刊的出版、发行要兼顾到两个方面, 一方面是书刊的排版、印刷, 另一方面则是科技文献的检索、查询。当前流行的可用于科技排版的系统也可分成两大类。第一类以排版为出发点, 它们注重文档的版面格式, 如 MS Word, T_EX 等都属于这一类。第二类以文献的入库、检索为出发点, 其代表是 SGML 语言以及由它所导出的目前广泛应用于 WWW 网的 HTML 语言。L^AT_EX 2_ε 虽然是从 T_EX 导出的排版语言, 但由于它的命令注重于文档的逻辑结构, 因此其排版源文件亦提供了很好的文献功能。例如, 通过重新定义 L^AT_EX 2_ε 的一些命令、环境 (如 \title 命令, \author 命令, abstract 环境等) 或编写一个简单的转换程序, 不难自动从一个 L^AT_EX 2_ε 的排版源文件中抽取文章的标题、作者、摘要等信息。

L^AT_EX 2_ε 用于西文排版在国际上的使用非常广泛。国外许多专业出版部门都为自己的书刊设计了专门的宏命令集并将其提供给作者。作者只要按这些命令格式排版, 排版输出的格式自然满足出版部门的要求, 在为出版部门节省大量人力同时, 还简化了作者的排版过程。相比较而言, 在国内使用 T_EX 系统排版的编辑部、出版社中, 排版语言的使用则比较混乱。原因之

一是国内目前 $\text{T}_{\text{E}}\text{X}$ 方面的资料较少, 另一个原因则是中文 $\text{T}_{\text{E}}\text{X}$ 的排版尚未形成一套规范, 缺少适合于中文排版的标准宏命令集。

我们的建议是, 当使用 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 排版中文文档时, 在排版源文件的书写格式上尽量采用 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 的标准命令 (包括 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 的标准文档类及包)。一个好的、规范的排版源文件中不应该出现直接控制排版格式的命令 (如选择字体、字号, 留空等), 而只包含指明排版内容的逻辑含义的命令 (标题、作者、关键字、章节命令等)。当排版结果不符合要求时, 首先想到的应该 是如何修改 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 的命令及定义, 而不是自己的排版源文件。这样做的好处一方面是可以保证排版源文件的简洁、规范, 另一方面当需要改变排版格式时, 只需要使用一套不同的命令及定义而可以不改动排版源文件。编辑部门则可根据自己刊物的特点定义出自己的 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 的文档类或包供作者使用。作者、编辑部门基于一套同样的标准命令可以避免稿件的排版格式的重新调整, 提高稿件编辑出版的效率。

参考文献

- [1] Donald E. Knuth, “The $\text{T}_{\text{E}}\text{X}$ book”, Addison-Wesley, 1984
- [2] $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ ’s Guide for Class and package writers, ???