zhmakeindex*中文索引处理程序

刘海洋

2015 年 3 月 6 日

1 命令行

`zhmakeindex` [`-c`] [`-i`] [`-o` ⟨*ind*⟩] [`-q`] [`-r`] [`-s` ⟨*sty*⟩] [`-t` ⟨*log*⟩]
[`-enc` ⟨*enc*⟩] [`-senc` ⟨*senc*⟩] [`-strict`] [`-z` ⟨*sort*⟩]
[⟨*idx0*⟩ ⟨*idx1*⟩ ⟨*idx2*⟩ ...]

2 简介

zhmakeindex 是一个通用的中文多级索引处理程序，它从一个或多个输入文件读入索引项，将其内容按指定的方式分组、排序，然后按格式将整理好的索引输出到文件。索引项可以有 3 个级别（0, 1, 2）的嵌套。zhmakeindex 主要用于 LaTeX 索引的处理，其功能和用法与 makeindex[3] 相似，并支持中文的分组与排序。

输入与输出文件的格式由一个格式文件确定。默认的输入/输出是 `.idx`/`.ind` 格式的，即 LaTeX 格式的索引文件。

如果没有显式指定，第一个输入文件（⟨*idx0*⟩）的主文件名将用于确定输出和日志文件的主文件名。对每个输入文件名 ⟨*idx0*⟩, ⟨*idx1*⟩, ..., zhmakeindex 会首先查找这个名字的文件；如果找不到且文件名没有后缀，则加上 `.idx` 后缀查找此文件；如果还找不到文件，zhmakeindex 则会中止。

如果只有一个输入文件，并且没有显式用 `-s` 选项指定格式文件，zhmakeindex 会使用后缀为 `.mst` 的默认格式文件（如果存在的话）。

关于如何使用 makeindex 在 LaTeX 文档中处理索引，可以参考陈丕宏较早的文档 [1]，或 Lamport 的文档 [2]。zhmakeindex 的在 LaTeX 中的使用方法与 makeindex 基本一致。

3 选项说明

3.1 与 makeindex 相同的选项

-c 压缩索引项排序项前后的空格。默认情况下，排序项中的空格会被保留。

*版本 1.1-rev110(14a464d28db8)

-i 从标准输入流（stdin）读入索引项。如果使用了该选项，并且没有使用 -o 选项，则排序后的索引将输出到标准输出流（stdout）。

-o ⟨*ind*⟩ 设置输出索引文件为 ⟨*ind*⟩。如果没有指定该选项，默认的输出文件名是第一个输入文件 ⟨*idx0*⟩ 的主文件名加上 .ind 的扩展名。

-q 静默模式，不向标准错误流（stderr）显示信息。默认情况下处理过程与错误信息会同时在 stderr 与日志文件中输出。

-r 禁止隐式页码区间构造，要求页码区间必须使用显式区间符号生成。见第 4 节的说明。默认情况下，三个或三个以上连续的页码会自动合并为一个页码区间（如 1–5）。

-s ⟨*sty*⟩ 设置 ⟨*sty*⟩ 为格式文件。没有默认值。zhmakeindex 会首先在当前目录查找格式文件，如果找不到则调用 TEX 发行版的 kpathsea 库在 TEXMF 树中查找。

-t ⟨*log*⟩ 设置 ⟨*log*⟩ 为日志文件。默认情况下，会使用第一个输入文件 ⟨*idx0*⟩ 的主文件名加上 .ilg 后缀作为日志文件。

3.2 zhmakeindex 独有的选项

-enc ⟨*enc*⟩ 设置输入输出文件的编码为 ⟨*enc*⟩。可选的编码包括 UTF-8, UTF-16, GB18030, GBK 和 Big5，不区分大小写。默认使用 UTF-8 编码。

-senc ⟨*senc*⟩ 设置读入格式文件的编码为 ⟨*senc*⟩。可选的编码与 -enc 选项相同。默认使用 UTF-8 编码。

-strict 严格区分不同嵌入命令的页码。默认情况下，在页码区间处理时，会将如果页码左区间的嵌入命令与右区间不匹配，会以左区间为准（部分 LATEX 文档会生成右区间命令缺失的索引项）；而如果使用 -strict 选项，则要求左右区间的命令类型必须严格匹配。

-z ⟨*sort*⟩ 设置中文分组与排序方式为 ⟨*sort*⟩。可选的中文分组排序方式包括 pinyin/reading, bihua/stroke, bushou/radical。默认值为 pinyin，即中文按拼音分组排序。有关分组与排序的详细说明见第 6 节。

3.3 未实现的选项

zhmakeindex 没有实现 makeindex 的 -g, -l, -p, -L, -T 选项。其中，选项 -p 依赖对 TEX 编译的日志文件的解析，可能在未来版本实现；另外几个语言相关的排序选项（-g 德文，-T 泰文，-L 做系统 locale 选择，-l 有关西文单词排序）对中文索引意义较小，不在 zhmakeindex 中实现。

4 索引项输入语法

zhmakeindex 输入索引项的语法与 makeindex 相同。下面以默认的格式输入设置（即在 LATEX 中使用）为例对索引项输入语法进行说明。

索引项文件是一个由多条索引项组成的文本文件，每行一条索引项，允许有空行，但索引项不得跨行。

一个简单的索引项及其输出如

```
\indexentry{简介}{15}
```

简介, 15

其语法为：

```
\indexentry{⟨条目⟩}{⟨页码⟩}
```

在 LaTeX 中，`\indexentry` 项是在编译过程中自动生成的，实际生成上面例子的 LaTeX 代码并不包括页码问题，只是在文档 15 页的源文件处输入 `\index{简介}` 即可得到这样的效果。关于 makeindex 在 LaTeX 中的使用可以参考文档 [1, 2]。

`\indexentry` 是在索引输入文件中表示索引项的关键字。可以使用 `keyword` 输入格式进行修改（表 1）。左右花括号是界定 ⟨条目⟩ 与 ⟨页码⟩ 的标记，可以使用 `arg_open`, `arg_close` 输入格式修改（表 1）。

⟨页码⟩ 是一个数字，可以使用阿拉伯数字、大小写罗马数字、大小写拉丁字母共 5 种格式。

此外，与 makeindex 类似 [3]，zhmakeindex 还支持多级页码。可以使用 - 划分不同级别的复合页码。页码分隔符可由 `page_compositor` 输入格式修改，见 表 1。例如：

```
\indexentry{方程}{5-ii-2}
\indexentry{方程}{5-ii-1}
\indexentry{方程}{3-v-9}
```

方程, 3-v-9, 5-ii-1, 5-ii-2

⟨条目⟩ 是 zhmakeindex 需要处理的正文内容。最简单的条目就是一个普通的词条串，但也可以有复杂的格式。

如果对条目排序使用的串与用来输出条目的 LaTeX 代码不一样，则可以把这两部分用符号 `@` 分开，前面是排序的键，后面是输出的值。例如：

```
\indexentry{Gamma 射线@$\Gamma$ 射线}{2}
\indexentry{粗体@\textbf{粗体}}{3}
```

粗体, 3
Γ 射线, 2

分隔符 `@` 可通过修改格式文件的 `actual` 项配置（表 1）。

索引项的条目可以分成至多三级，不同级别之间用符号 `!` 分开。例如

```
\indexentry{方程!解}{1}
\indexentry{方程!代数方程!线性方程}{2}
\indexentry{方程!代数方程!二次方程}{3}
```

方程
　代数方程
　　二次方程, 3
　　线性方程, 2
　解, 1

每一级别内部都可以分别指定排序的键与输出值，例如：

```
\indexentry{射线!Gamma 射线@$\Gamma$ 射线}{8}
\indexentry{射线!X 射线}{9}
```

射线
　Γ 射线, 8
　X 射线, 9

即分隔符 ! 的优先级低于 @。分隔符 ! 可通过修改格式文件的 level 项配置（表 1）。

索引项条目中也可以在最后指定页码的输出格式，用分隔符 | 与前面分开（用 表 1 中的 encap 项配置）。页码的输出格式由一个带单个页码参数的 TEX 命令决定，在输入索引项时这一特殊命令的反斜线 \ 和左右花括号被忽略，输出时会自动加上（用表 2 中的 encap_prefix, encap_infix, encap_suffix 项配置）。例如：

```
\indexentry{字体|emph}{2}
```
字体, *2*

它输出的代码实际上是

```
字体, \emph{2}
```

除了指定页码输出格式的特殊命令，还可以指定页码范围，在分隔符 | 后、特殊命令名前（如果有的话），使用开定界符 (和闭定界符)，分别表示页码范围的起始和结束。例如：

```
\indexentry{方程|(}{5}
\indexentry{方程|)}{9}
\indexentry{函数!二次函数|(emph}{11}
\indexentry{函数!二次函数|)emph}{15}
```
方程, 5–9
函数
    二次函数, *11–15*

页码范围的开闭定界符可由表 2 中的 range_open 与 range_close 项配置。

由于在索引条目中，{, }, (,), !, @, | 等多种符号都有特殊意义，因此引入引号 " 作为索引条目的转义符，引号及紧跟引号的字符都作为普通的单个后一字符看待，不作为索引条目的特殊语法符号。例如：

```
\indexentry{Aha"!}{1}
\indexentry{绝对值 $"|x"|$}{2}
```
Aha!, 1
绝对值 $|x|$, 2

引号字符 " 可以由表 1 中的 quote 项配置。

使用转义符有一个例外，就是在反斜线后的引号 \" 被解释为普通 TEX 命令 \"，这是因为 \" 经常被用于输入 Gödel（G\"odel）这样的文字。注意这一例外会影响 \| 等符号组合的输入。例如：

```
\indexentry{G\"odel}{5}
\indexentry{命令 \verb+"\"|+}{9}
```
Gödel, 5
命令 \|, 9

转义符 \ 可以由表 1 中的 escape 项配置。

5 格式文件

zhmakeindex 的格式文件与标准 makeindex 语法完全相同，内容也基本上相同，同时增加了少量控制中文分组输出的格式。可以在 zhmakeindex 中使用标准 makeindex 的格式文件。

格式文件指明了 .idx 输入文件和最终输入文件的格式。该文件在当前工作目录或在 TEXMF 树中由 kpathsea 库查找。一个格式文件由一组 ⟨关键字⟩⟨属性⟩ 的列表组成。⟨关键字⟩ 分为输出和输出两类。⟨关键字⟩⟨属性⟩ 对儿不要求以任何顺序出现。 以字符 % 开头的行是注释。⟨属性⟩ 可以有不同的类型，字符串是以任意双引号 " 或反引号 ` 界定的串，字符是以单引号 ' 界定的单个字符，数字是一个整数。 其中，双引号和单引号界定的串和字符，可以使用反斜线 \ 符号作为转义符，以得到特殊字符或引号本身；而反引号界定的串是 zhmakeindex 特有的功能，它不使用转义符。格式文件中没有指定的项目会使用其默认值。

5.1 与 makeindex 兼容的格式

表 1 与表 2 的格式是在 makeindex 中有定义，同时也在 zhmakeindex 中有支持的输入、输出格式。zhmakeindex 的这部分格式选项与 makeindex 意义相同，完全兼容。

表 1: 与 makeindex 兼容的输入格式

关键字	类型	默认值	意义
keyword	字符串	"\\indexentry"	索引关键字
arg_open	字符	'{'	参数的开定界符
arg_close	字符	'}'	参数的闭定界符
range_open	字符	'('	页码范围的开定界符
range_close	字符	')'	页码范围的闭定界符
level	字符	'!'	索引项层次分隔符
actual	字符	'@'	(区别于排序项的) 实际输出项标志符
encap	字符	'|'	页码和特殊命令指示符
quote	字符	'"'	引号 (转义符)
escape	字符	'\\'	转义 quote 的符号，在它后面的引号不作转义符解释
page_compositor	字符串	"-"	复合页码分隔符

表 2: 与 makeindex 兼容的输出格式

关键字	类型	默认值	意义
preamble	字符串	"\\begin{theindex}\n "	索引导言代码
postamble	字符串	"\n \n \\end{theindex}\n "	索引末尾代码
group_skip	字符串	"\n \n \\indexspace\n "	组间垂直间距
headings_flag	数字	0	控制显示分组名标题的旗标
heading_prefix	字符串	""	分组名标题的前缀
heading_suffix	字符串	""	分组名标题的后缀
symhead_positive	字符串	"Symbols"	当旗标 headings_flag 为正数时，符号的标题
symhead_negative	字符串	"symbols"	当旗标 headings_flag 为负数时，符号的标题
numhead_positive	字符串	"Numbers"	当旗标 headings_flag 为正数时，数字的标题
numhead_negative	字符串	"numbers"	当旗标 headings_flag 为负数时，数字的标题
item_0	字符串	"\n \\item "	第 0 级条目间分隔
item_1	字符串	"\n \\subitem "	第 1 级条目间分隔
item_2	字符串	"\n \\subsubitem "	第 2 级条目间分隔
item_01	字符串	"\n \\subitem "	第 0/1 级条目间分隔

关键字	类型	默认值	意义
item_x1	字符串	"\n \\subitem "	第 0/1 级条目间分隔，其中第 0 级无页码
item_12	字符串	"\n \\subsubitem "	第 1/2 级条目间分隔
item_x2	字符串	"\n \\subsubitem "	第 1/2 级条目间分隔，其中第 1 级无页码
delim_0	字符串	", "	第 0 级条目与页码分隔
delim_1	字符串	", "	第 1 级条目与页码分隔
delim_2	字符串	", "	第 2 级条目与页码分隔
delim_n	字符串	", "	多个页码的分隔
delim_r	字符串	"--"	页码范围符号
delim_t	字符串	""	将插入到在页码列表末尾。如果页码列表为空，此项无效果。
encap_prefix	字符串	"\\"	页码特殊指令前缀
encap_infix	字符串	"{"	页码特殊指令中缀
encap_suffix	字符串	"}"	页码特殊指令后缀
page_precedence	字符串	"rnaRA"	不同类型页码的次序，默认值表示小写罗马、阿拉伯数字、小写字母、大写罗马、大写字母
suffix_2p	字符串	""	在 2 页的页码范围中代替 delim_r 和第二个页码
suffix_3p	字符串	""	在 3 页的页码范围中代替 delim_r 和后面的页码
suffix_mp	字符串	""	在更多页的页码范围中代替 delim_r 和后面的页码

5.2 zhmakeindex 特有的格式

zhmakeindex 定义了新的输入格式（表 3），以支持索引输入的行注释。

表 3: zhmakeindex 特有的输入格式

关键字	类型	默认值	意义
comment	字符	'%'	行注释的开始符

zhmakeindex 定义了一些新的输出格式（表 4），用来控制按笔画数或部首分组时，分组名的输出格式。

表 4: zhmakeindex 特有的输出格式

关键字	类型	默认值	意义
stroke_prefix	字符串	""	笔画数前缀
stroke_suffix	字符串	" 画"	笔画数后缀
radical_prefix	字符串	""	部首前缀
radical_suffix	字符串	"部"	部首后缀
radical_simplified_flag	数字	1	是否输出简化部首的标志
radical_simplified_prefix	字符串	"（"	简化部首前缀
radical_simplified_suffix	字符串	"）"	简化部首后缀

5.3 未实现的 makeindex 格式

表 5 中给出的是在标准的 makeindex 中定义，但在 zhmakeindex 中没有实现对应功能的格式。格式文件中如果出现这些格式，zhmakeindex 将会忽略它们。

表 5: 未实现的输出格式

关键字	类型	默认值	意义
setpage_prefix	字符串	"\n \\setcounter{page}{"	页码设置前缀
setpage_suffix	字符串	"}\n "	页码设置后缀
line_max	数字	72	最大行长度，超出长度会自动折行
indent_space	字符串	"\t \t "	自动折行的缩进
indent_length	数字	16	indent_space 的长度

5.4 格式文件示例

合法的 makeindex 格式文件都是合法的 zhmakeindex 格式文件。LaTeX 2_ε 的 doc 包附带的 gind.ist 和 gglo.ist 就是两个实际的例子。此外，makeindex 手册页 [3] 也给出了几个有用的例子。

下面是本文档使用的格式文件，注意其中反引号格式的串是 zhmakeindex 特有的：

```
% 此索引格式用来编译 zhmakeindex 的手册

preamble
`\begin{theindex}
  \def\seename{见}
  \def\alsoname{又见}
  \providecommand*\indexgroup[1]{%
    \indexspace
    \item \textbf{#1}\nopagebreak}
`
```

```
postamble "\n\n\\end{theindex}\n"

group_skip "\n\n  \\indexspace\n\n"

headings_flag 1
heading_prefix "  %\n  \\indexgroup{"
heading_suffix "}\n  %\n"

numhead_positive "数字"
numhead_negative "数字"
symhead_positive "符号"
symhead_negative "符号"
```

6 排序细节

6.1 索引项分组

zhmakeindex 与 makeindex 类似，主要是按第一级索引项的排序项的首个字符分组的。对最普通的场景，西文是按单词的首字母分组，中文则根据 `-z` 选项（3.2 节）的不同，选择对应的分组，数字和其他符号单独分组。

zhmakeindex 的前 28 个分组是固定的，分别是符号、数字，以及 A–Z 的 26 个拉丁字母。按笔画排序时，后面是按总笔画数分组的汉字，共 64 组；按部首笔画排序时，后面是按康熙字典部首分组的汉字，共 214 组。详细情况见表 6。

表 6: zhmakeindex 支持的分组方式

`-z` 选项	别名	前 28 个分组	后面的分组
`pinyin`	`reading`	符号、数字、A, ..., Z	（无）
`bihua`	`stroke`	符号、数字、A, ..., Z	1 画、2 画、……、64 画（共 64 组）
`bushou`	`radical`	符号、数字、A, ..., Z	一部、丨部、……、龠部（共 214 组）

zhmakeindex 主要是面向中文排版的索引处理，因此对于西文条目，只对没有重音等符号的拉丁字母能进行正确的分组，对带重音符号的拉丁字母（如 é, ç）、非拉丁字母（如 Σ, Δ, Ж, Я 等）等则会一律按符号分组。

如果索引项的第一级排序项全部由数字组成，则该项会被分入数字分组；但如果条目的首个字符是数字，后面还有其他字符，则条目会被计入符号分组。除了阿拉伯数字，zhmakeindex 还将其他 Unicode 数字符号（如罗马数字“Ⅳ”、带圈数字“⑧”）也当作数字处理，但汉字数码“〇”被看作汉字处理。

当索引项第一级排序项的首个字符不是拉丁字母或汉字，排序项本身也不是纯数字时，则此索引项就会计入符号分组。

当输出格式变量 `headings_flag` 非零时（参见表 2），将输出分组名称。`headings_flag` 是正数时，以大写字母显示分组名称；`headings_flag` 是负数时，以小写字母显示分组名称。

当使用汉字特有的分组时，分组名称可以使用格式文件定制（参见表 4）。按笔画分组时，分组名为

⟨*stroke_prefix*⟩⟨笔画数⟩⟨*stroke_suffix*⟩

按康熙字典部首分组时，如果变量 `radical_simplified_flag` 非零（默认情况），则分组名由原康熙字典部首与简化字部首组合而成：

⟨*radical_prefix*⟩⟨部首⟩⟨*radical_suffix*⟩
⟨*radical_simplified_prefix*⟩⟨简化字部首⟩⟨*radical_simplified_suffix*⟩

如果变量 `radical_simplified_flag` 是零，则分组名没有简化字部首：

⟨*radical_prefix*⟩⟨部首⟩⟨*radical_suffix*⟩

6.2 索引项排序

大体上，zhmakeindex 逐字符按字典序对索引项的排序项进行排序，汉字与其他 Unicode 字符一样，按单个字符比较。排序时使用的字符串比较有一些特殊规则：

- 如果字符串只包含阿拉伯数字，则首先按数字大小比较，数字相等时再按字典序比较。
- 以符号开头的字符串总是先于排在以数字开头的串（即使符号的 Unicode 码在数字之后），而这又先于纯数字的排序项和以字母开头的串。
- 在比较两个字符串时，zhmakeindex 首先忽略字母大小写进行比较，如果此时结果相等，再按区分大小写进行比较，将大写字母排在小写字母之前。

makeindex 有一个 `-l` 选项忽略条目中的空格，按所有非空白符比较所有条目（3.3 节）。zhmakeindex 未提供此功能。

按 `-z` 选项（3.2 节）的不同，汉字可以使用不同的排序方式，如表 7 所示。

表 7: zhmakeindex 支持的中文排序方式

`-z` 选项	别名	意义
`pinyin`	`reading`	汉字按常用读音的拼音排序。
`bihua`	`stroke`	汉字按笔画数和笔顺排序。
`bushou`	`radical`	汉字按康熙字典部首和除部首笔画数排序。

使用读音排序时，没有读音数据的字符（包括生僻字）一律排在有读音的字符之前，汉字按其最常用读音比较，读音相同汉字的按 Unicode 编码排序。使用笔画数和笔顺排序时，笔画数小的排在前面，笔画数相同的，按横、竖、撇、点（捺）、折的顺序逐笔画比较，仍然相同的按 Unicode 编码排序；生僻汉字没有笔顺信息的，排在同笔画数有笔顺的字后面。使用部首和除部首笔画数排序时，部首按康熙字典 214 部首顺序排列，部首和笔画数相同的按 Unicode 编码排序。

6.3 页码排序与合并

内容完全相同而页码不同的索引项，会在排序时合并为一项。同一索引项的所有页码会按前后次序排序、去重，并按要求合并为页码区间。

zhmakeindex 能识别不同类型的页码格式，包括阿拉伯数字（1, 2, 3, ...）、大小写罗马数字（I, II, III, ...）、大小写拉丁字母（a, b, c, ...）。不同类型的页码按照 `page_precedence` 变量的设置（表 2）区分前后次序。默认情况下，不同类型的页码次序是 `rnaRA`，即小写罗马数字的页码排在最前，然后依次是阿拉伯数字、小写拉丁字母、大写罗马数字、大写拉丁字母的页码。

目前，页码的类型判别比较粗糙，仅以页码的第一个字符判断类型。当页码以字母 `i`, `v`, `x`, `l`, `c`, `d`, `m` 起始时，即被识别为罗马数字，其他字母被识别为拉丁字母页码。

zhmakeindex 支持页码区间，在索引项中显式指定区间，如对输入

```
\indexentry{foo|(}{5}
\indexentry{foo|)}{10}
```

则 `foo` 项会输出页码 5–10。连续的区间、区间内的零散页码会被合并为一个大的区间，如对输入

```
\indexentry{foo|(}{5}
\indexentry{foo}{6}
\indexentry{foo|)}{10}
\indexentry{foo|(}{11}
\indexentry{foo|)}{13}
```

则 `foo` 项会输出页码 5–13。

如果没有使用 `-r` 选项（3.1 节），零散的页码也会与区间一起合并，3 页或以上连续的页码也会被合并为区间。例如输入

```
\indexentry{foo}{i}
\indexentry{foo}{ii}
\indexentry{foo}{iii}
\indexentry{foo|(}{5}
\indexentry{foo|)}{7}
\indexentry{foo}{8}
```

则 `foo` 项会输出页码 i–iii，5–8。使用 `-r` 选项将禁止这种自动合并。

在排序合并页码时，zhmakeindex 会区分不同数字类型的页码；如果页码还有特殊命令修饰，则还会区分不同的修饰命令。例如输入

```
\indexentry{foo}{1}
\indexentry{foo|textit}{5}
\indexentry{foo}{3}
\indexentry{foo|textit}{7}
```

则 foo 项会分别以不同格式输出页码 1, 3, *5*, 7。

在页码区间中如果出现不同类型的特殊命令修饰，则按不同的页码格式分别输出。但是，为了适应部分 LaTeX 代码不精确的页码输出，如果页码区间头与页码区间尾使用的特殊命令不同，仍把它们看做相同的格式，按页码区间头的格式输出。例如输入

```
\indexentry{foo|(textit}{1}
\indexentry{foo|textbf}{5}
\indexentry{foo|textit}{2}
\indexentry{foo|)}{4}
```

则 foo 项会把页码 1, 2, 4 都归入区间 1–4，输出页码 *1–4*, **5**。这也是 makeindex 默认效果。如果使用了 -strict 选项（3.2 节），则不再允许页码区间有不同的特殊命令修饰，而要求格式严格匹配。-strict 选项适合用于不同格式页码区间相互交错的复杂的页码格式，例如输入

```
\indexentry{foo|(textit}{1}
\indexentry{foo|(textbf}{5}
\indexentry{foo|)textit}{8}
\indexentry{foo|)textbf}{9}
```

则 foo 项会输出页码 **5–9**, *1–8*。但如果不使用 -strict 选项，或者使用 makeindex，则无法正确识别这类页码区间。

7 与 makeindex 的比较

- ✓ zhmakeindex 支持 Unicode，所有字符按 Unicode 字符而非字节流处理。同时支持输入输出文件、格式文件使用不同的中文编码（3.2 节）。
- ✓ zhmakeindex 支持中文特有的分组与排序方式（第 6 节）。
- ✓ zhmakeindex 与 makeindex 的页码合并算法不同（6.3 节），在页码区间有嵌套和交错，前后分界的格式不统一时，zhmakeindex 与 makeindex 可能会产生不完全相同的效果。如果使用 -strict 选项，zhmakeindex 会使用更复杂的页码区间合并算法，能正确处理不同格式多层嵌套和互相交错的页码区间，也能识别输入不正确的前后区间分界，而 makeindex 将给出错误的结果。
- ✓ zhmakeindex 的格式文件支持使用反引号界定的串，在其中禁用转义符（第 5 节）。
- ✓ zhmakeindex 对输入的索引文件支持单个注释符开始的行注释。
- ✗ zhmakeindex 暂不支持 -p 选项自动获取 TeX 编译日志文件中的页码，然后输出对应的页码设置语句（3.3 节、表 5）。
- ✗ zhmakeindex 不支持 -g, -l, -L, -T 等与中文索引无关的语言选项。
- ✗ zhmakeindex 不支持输出折行（表 5）。

8 已知问题

汉字按拼音分组排序时，多音字可能会被分到错误的分组或排在错误的位置。例如，“长度”的“长”有 zhǎng 与 cháng 两个常用读音，由于读音 zhǎng 的使用频率比 cháng 略高一点，“长”字就会按 zhǎng 的读音分到 Z 组，排序也较为靠后。目前 zhmakeindex 缺少一种通用的方式控制多音字按拼音分组与排序，只能暂时使用同音字处理此问题。另外，部分汉字因为旧字形等问题，笔画数和笔顺也可能有多种选择，造成分组的分歧。

9 版权与许可

以及编译源文件得到的二进制文件 zhmakeindex.exe 或 zhmakeindex、PDF 文档 zhmakeindex.pdf 组成。

大部分汉字数据来自 Unicode 项目（http://www.unicode.org/）：

maketables/CJKRadicals.txt
maketables/Unihan_DictionaryLikeData.txt
maketables/Unihan_RadicalStrokeCounts.txt
maketables/Unihan_Readings.txt

部分字形数据来自海峰五笔项目（http://okuc.net/sunwb/）：

maketables/sunwb_strokeorder.txt

参考文档

[1] Pehong Chen and Michael A. Harrison. Index preparation and processing. *Software—Practice and Experience*, 19(9):897–915, September 1988. ISSN 0038-0644. URL CTAN://indexing/makeindex/paper/ind.pdf. The LaTeX text of this paper is included in the makeindex software distribution.

[2] Leslie Lamport. *MakeIndex: An Index Processor For LaTeX*, February 1987. URL CTAN://indexing/makeindex/doc/makeindex.pdf.

[3] Rick P. C. Rodgers. *Makeindex(1)—Manual Page*. UCSF School of Pharmacy, December 1991. URL CTAN://indexing/makeindex/doc/manpages.dvi.

索引

Z