



## 第九章 错误消息

在制作长的 L<sup>A</sup>T<sub>E</sub>X 文档时很容易出现错误。这里的错误可以是各种类型，从最简单的命令名称输入错误到忘记了某些命令必须配对，或者忘记了复杂命令的语法等等种类的错误都有可能发生。

在 L<sup>A</sup>T<sub>E</sub>X 处理过程中的错误会在屏幕上显示出一长串消息，这些信息对初学者而言，是完全不可理解的。即便是高级用户在确定出错的具体地方时也会有一些困难。然而，这其中包含的关于基本结构信息，可以帮助有经验的 T<sub>E</sub>X 使用者深入了解问题本质。

另外，错误消息中也包含对初学者而言相当有用的信息。本章的目的就是解释一下这些对非程序设计者有帮助的消息。

### §9.1 错误消息的基本结构

错误消息有两个来源：一些来自于 L<sup>A</sup>T<sub>E</sub>X，另一些来自于基本的 T<sub>E</sub>X 程序。由于 L<sup>A</sup>T<sub>E</sub>X 是在更高层次上进行操作，因此 L<sup>A</sup>T<sub>E</sub>X 消息之后通常接着 T<sub>E</sub>X 消息。

#### §9.1.1 T<sub>E</sub>X 错误消息

我们从一个简单的错误例子开始。

```
\documentclass{article}
\begin{document}
The last words appear in \textbf{bold face}.
\end{document}
```

其中 `\textbf` 命令被误输入为 `\txetbf`。在处理过程中，L<sup>A</sup>T<sub>E</sub>X 认为作者想调用的是 T<sub>E</sub>X 命令 `\txetbf`。由于 L<sup>A</sup>T<sub>E</sub>X 不认识 T<sub>E</sub>X 命令，它只是把这条命令送给 T<sub>E</sub>X 处理，在这里可以确定其指令系统中并没有这条命令。那么就会在屏幕上显示出下面的错误消息：

```
! Undefined control sequence.
1.3 The last words appear in \txetbf
                                {bold face}.
?
```

程序运行到此处就停下来，等待用户给出反馈。这条消息即使是初学者也能理解。其中包含由惊叹号！开始的错误标识。这里的标识为 `! Undefined control sequence`，指的是未知的命令名称（控制序列）导致出错。接下来是两行文本，第一行前缀 `1.3`，指的是错误出现在输入文本的第 3 行。而错误本身就是在这一行最后那个符号中遇到的。而下面一行显示的是当遇到这

个错误时正在处理的输入行的其它部分，这里是单词 `{bold face}`。在继续向下处理之后，消息中最后一行的问号标志表示 `TEX` 需要用户的反应。

再输入一个 `?`，并回车，那么就会显示出如下信息：

```
Type <return> to proceed, S to scroll future error messages,
R to run without stopping, Q to run quietly,
I to insert something, E to edit your file,
1 or ... or 9 to ignore next 1 to 9 tokens of input,
H for help, X to quit
?
```

下面就是用户可能给出的反应清单：

1. **〈回车〉**：简单地输入回车键，让 `TEX` 在经过一番按照预先设计好的规则，处理这个错误后，继续向下处理。在这种出现未知命令名称的情形中，处理错误的方式就是这样来忽略它。
2. **S 滚动模式**：`TEX` 继续向下处理，当再次遇到错误时，还会在屏幕上显示出消息，但并不等用户做出反应。这就好像在所有后续错误中按的都是 **〈回车〉** 键。
3. **R 运行模式**：`TEX` 如同 **S** 一样继续向下处理，但是即使遇到类似于在 `\input` 或 `\include` 命令没有文件名这样的错误也不停下来。
4. **Q 安静模式**：同 **R** 一样，只是不会再向屏幕上显示错误消息。然而这些消息要写到 `.log` 文件中。
5. **I 插入**：通过插入正确的文本来校正错误。`TEX` 把从键盘上输入的文本行取代出错的地方，然后继续处理下去。对于这样的校正，实际上 `.tex` 文件中的原始文本并没有改变，还是需要用编辑器进行修改。输入 `I\stop` 会使得在 `.dvi` 文件的当前页上终止程序。
6. **1 ...**：输入一个小于 100 的数，从而在后续文本中删掉许多字符和命令。程序还是会停下来等用户的反应。
7. **H 帮助**：在屏幕上显示出对问题的进一步考虑，它由相对于简短错误标志要多很多的信息组成，从而可能包含改正错误的有用提示。
8. **X 退出**：在该点停止执行 `TEX` 处理过程。当前页并不出现在 `.dvi` 文件中。
9. **E 编辑**：同 **X** 一样，处理过程中止，并显示信息，说哪个文件的哪一行出现了错误。在有些实现版本中，可能自动调用编辑器，而且直接跳到出错行。

上面的反应字母既可以是写大的，也可以是小写的。只有按了 **〈回车〉** 键，反应才有作用。

在前面那个示例错误消息中，若按 **H** 或 **h** 以寻求帮助，会得到如下文本：

```
The control sequence at the end of the top line
```

of your error message was never `\def`'ed. If you have misspelled it (e.g., `'\hobx'`), type `'I'` and the correct spelling (e.g., `'I\hbox'`). Otherwise just continue and I'll forget about whatever was undefined.

?

这里更详细地描述了错误情形：在上面一行尾部的命令名称是未知的；如果这只是一个输入错误，那么就用 `I` 输入正确的文本，这里是 `I\textbf`。否则，按 `<回车>` 键，有错的命令被忽略。对这行的处理就如同是 `The last word appears in bold face`。当然，这里并不会真的得到黑体。

TeX 的错误消息结构总结如下：

每条错误消息都以错误指示开始，其第一行的开头有 `!` 号。所谓指示文本就是对问题的简要说明。接下来是一行或多行的输入文本。其中第一行的最后一个符号使得 TeX 停下来，并显示错误消息。最后一行由接下来要处理的文本或命令组成。TeX 要等待用户的反应。如果这里的反应是寻求帮助的 `H`，那么就是在屏幕上显示出更详细的说明，这可能会给出一些提示，而且 TeX 还会等待进一步的反应。

### §9.1.2 L<sup>A</sup>T<sub>E</sub>X 的错误消息

在 L<sup>A</sup>T<sub>E</sub>X 2.09 与 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 之间的一个主要差别就是它们的错误消息样子不同。在原来的版本中会显示出许多行难以理解的文本，以揭露出导致出错的深层内部代码，而新的版本只是显示出错误大概发生于其中的几行文本。在下面一节我们会讲到来自于 L<sup>A</sup>T<sub>E</sub>X 2.09 的错误消息例子，除此之外所有其它例子都是针对于 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 的，发行期为 `<1994/06/01>`。

为了给出一个有 L<sup>A</sup>T<sub>E</sub>X 错误的文本例子，我们如下输入：

```
\documentclass{article}
\begin{document}
\begin{qoute}\slshape
  Text indented at both ends
\end{qoute}
\end{document}
```

这里 `\begin{qoute}` 调用中错误输入为 `qoute`。当 L<sup>A</sup>T<sub>E</sub>X 处理这段文本时会显示如下错误消息：

```
! LaTeX Error: Environment qoute undefined
```

```
See the LaTeX manual or LaTeX Companion for explanation.
```

```
Type H <return> for immediate help.
```

...

```
1.3 \begin{quote}
```

```
\slshape
```

?

这段消息的第一行就用一个简短的错误指示，说明这是  $\text{\LaTeX}$  自己发现的一个错误，这里是 `Environment quote undefined`。所有的  $\text{\LaTeX}$  错误都是以类似这样的一行开始的，接着告诉若想得到详细解释，请看  $\text{\LaTeX}$  手册（在本书的 9.3 节也可以看到这一介绍）。文本的第三行提示利用反应 `H`（回车）可以得到其它的说明。

有三个点 ... 的那一行表示这里没有写出来的有关内部代码的许多行。在  $\text{\LaTeX} 2.09$  中会显示出这些内部代码行，即使在  $\text{\LaTeX} 2_{\epsilon}$  中，也可能有选择地显示出一些。请见 9.1.3 节。

接下来的两行显示的是当发现错误时处理过程所停住的地方。同  $\text{\TeX}$  消息一样，文本的当前输入行也是断在出错的地方，前面部分放在第一行上，这里的断点为 `\begin{quote}`，其它部分放在下面的行中。行指示符 1.3（小写字母 L，不是数字 1）表示是在输入文件的第 3 行上遇到错误的。

$\text{\LaTeX}$  现在等用户给出一个反应。输入 `H`（回车）会得到额外的信息：

```
Your command was ignored.
```

```
Type I <command> <return> to replace it with another command,
```

```
or <return> to continue without it.
```

?

这就是说以 1.3 开始的两行中上面这行的最后那条命令还没有进入处理过程中。因此可以用只适用于当前处理过程的 `I\begin{quote}`（回车）方法来校正。但是这里文件中的拼写错误仍然没改正过来，需要稍后再运行编辑器把它改过来。

然而，如果只是简单地输入（回车）键，处理过程就会继续下去，这样出错的 `\begin{quote}` 就会被忽略，好像它在文件中不存在一样。这样当遇到 `\end{quote}` 命令时就会直接导致另一个错误，因此这里没有配对的 `\begin{quote}` 命令。这第二条错误消息的内容为：

```
! LaTeX Error: \begin{document} ended by \end{quote}
```

```
See the LaTeX manual or LaTeX Companion for explanation.
```

```
Type H <return> for immediate help.
```

...

```
1.5 \end{quote}
```

?

这条消息中照样还是包含标准的 L<sup>A</sup>T<sub>E</sub>X 声明，即有错误指示的第一行，接下来提醒参考手册，以及输入 H 可以得到帮助。在这种情形中的错误指示为

```
\begin{document} ended by \end{quote}
```

这是因为当 L<sup>A</sup>T<sub>E</sub>X 遇到 `\end{quote}` 时，它会检查当前环境的名称。由于没有 `\begin{quote}`，因此匹配的 `\begin` 命令就是 `\begin{document}` 声明，从而得到一个匹配错误的 `\begin...\end`。

最后两行显示的仍然是当发现错误时，处理已进行到了哪里。这里整个当前行都已被考虑进来了，因此另一行是空的。

在这时的反应 H(回车) 会得到与第一个错误相同的消息。这里利用 I 进行改正并不会取得好效果，因为这里再不可能把不存在的 `\begin{quote}` 插入到环境文本的前面。输入 I`\begin{quote}` 只是用它代替 `\end{quote}`，这并没有解决实际问题。现在最好的反应就是输入 (回车)，这样 `\end{quote}` 命令也被忽略，处理过程继续下去。

这样到现在为止，有错的 `\begin{quote}` 和正确的 `\end{quote}` 命令都被去掉了，处理过程就如同在源文件中没有用 `quote` 环境一样。

如果是在 `\end` 处犯同样的拼写错误，而在 `\begin` 处则拼写正确，那么会得到如下错误消息：

```
! LaTeX Error: \begin{quote} on input line 3 ended by \end{qoute}
```

```
See the LaTeX manual or LaTeX companion for explanation.
```

```
Type H <return> for immediate help.
```

```
...
```

### 1.5 `\end{quote}`

?

前面的解释对读者理解这条消息的内容应该是足够了。这里的错误指示为

```
\begin{quote} on input line 3 ended by \end{qoute}
```

最后两行文件表明问题出现在第 5 行，导致麻烦的命令是 `\end{quote}`。现在明显可以采取的方法就是用 I`\end{quote}` 进行改正，H 消息也支持采取这一操作。然而，现在会在屏幕上显示出新的错误消息：

```
! Extra \endgroup
```

```
<recently read. \endgroup
```

```
1.5 \end{quote}
```

?

这里除了以 1.5 开始的最后两行外，其它的都一点儿道理也没有。错误指示 `! Extra \endgroup` 好像没有任何意义。这是一个 `TeX` 错误，而不是 `LaTeX` 错误，它对我们没有任何帮助。

此时不应责备受了挫折的用户。这里的反应是相当合理的，虽然它还是错的。只有在具备了丰富的经验后，我们才会知道此时最好的方法就是按〈回车〉。这样就会关掉 `quote` 环境，当然与 `\end{quote}` 相联系的任何特殊操作也随之去掉了。

此时的帮助消息是：

`Things are pretty mixed up, but I think the worst is over.`

这一消息至少是一个鼓励，读者不要灰心。最好的方法就是继续按〈回车〉，以结束这次运行。

这里我们给出如何选择反应的两条建议，一条非常具体，另一条很一般。具体的这条是：

如果出错的环境名位于 `\begin` 命令中，正确的改正错误的方法是 `I\begin{正确的名称}`

如果拼写错误出现在 `\end` 命令中，那么最好的处理方法就是按〈回车〉。这样就会关闭这个环境，任何局部声明或定义也会终止作用。然而，如果 `\end` 命令执行了某命令，或显示出一些文本，那么这些结果也同时消失了。

一般性建议是：

如果借助于错误消息，用户知道了如何改正错误，那么可以用 `I 修正的文本`

来进行。否则，用户可以按〈回车〉，等等看会出现什么结果。即使出现更古怪的 (`TeX`) 错误，用户也可以持续按〈回车〉键，直到处理过程被终止。那么接下来的输出结果会指出错误原因。

当然也可以不按〈回车〉键，而是先输入 `S`, `R` 或 `Q`，再按〈回车〉，以加速对错误的处理 (9.1.1 节)。在这里，如果只是按〈回车〉，错误的命令并没有被忽略。`TeX` 在尝试猜测用户此处想进行的操作，从而进行一些改正。只有这一点行不通时，`TeX` 才会完全忽略这条命令。例如，如果错误指示为

`\begin{环境} ended by \end{环境}`

这里至少有一条 `\begin` 命令中有一个非法的环境名称。这样就可以假设在 `\end` 命令中的环境名也是错误的。`LaTeX` 就会尝试利用当前环境的名称来执行这条命令。

§9.1.3 来自于 L<sup>A</sup>T<sub>E</sub>X 2.09 中的错误消息

为了进行对比，我们先把 233 页上输入例子中的 `\documentclass` 换为 `\documentstyle`，然后我们看看来自于 L<sup>A</sup>T<sub>E</sub>X 2.09 中同样错误的消息结构。

第一行消息显示为：

```
! LaTeX Error.    See LaTeX manual for explanation.
                  Type H <return> for immediate help.
! Environment quote undefined.
\@latexerr ...diate help.}\errmessage {#1}
```

```
1.3 \begin{quote}
```

?

这里主要的差别就是错误指示现在位于第三行上，而不是做为第一行上 L<sup>A</sup>T<sub>E</sub>X 错误声明的一部分。另外一个差别就是第四行上的古怪文本，开头为 `\@latexerr ...`。这实际上是生成这个消息的内部代码，只要简单地忽略它就可以了。

当没有做任何修改，只是按〈回车〉键，那么会得到第二条错误消息，它与 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 中的样子极端不一样：

```
LaTeX error.    See LaTeX manual for explanation.
                  Type H <return> for immediate help.
! \begin{document} ended by \end{quote}.
\@latexerr ...diate help.}\errmessage {#1}

\@checkend ...urrenvir \else \@badend {#1}
                                     \fi
\end ...me end#1\endcsname \@checkend {#1}
                                     \expandafter ...

1.5 \end{quote}
```

?

在 `\@latexerr...` 和 `1.5...` 之间多出来的那些行对显示的是 `\end` 命令的内部代码，这些内容对通常 L<sup>A</sup>T<sub>E</sub>X 用户来说，没有任何兴趣。它们只会令人感到混乱。如果你仍然用的是 L<sup>A</sup>T<sub>E</sub>X 2.09，而且遇到了这样的错误消息，那么你就应该把这些行都忽略过去。除此之外，错误指示、用 `1.n` 开始的输入行、帮助消息和需要的反应都与 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 中完全一样。

有时候需要深入地观察错误消息，这一点对那些知道这些消息意味着什么的有经验 T<sub>E</sub>X 专家特别有用。在这种情况下，可以利用下面的指令来在



L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>中补上去掉的代码行:

```
\setcounter{errorcontextlines}{数}
```

这里的 数 就是在错误之处宏被解码的深度层数。在默认情形中, L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>为 数 = -1, L<sup>A</sup>T<sub>E</sub>X 2.09 为 数 = 5。有经验的用户可以把它设为 5 或其它的值以获得额外的信息。

#### §9.1.4 来自于 T<sub>E</sub>X宏的错误消息

绝大多数的 T<sub>E</sub>X命令和实际用的 L<sup>A</sup>T<sub>E</sub>X命令都可以称为 T<sub>E</sub>X宏。它们就是原语命令的组合,从而具有一个新的命令名称,这样就可以整体调用它们。T<sub>E</sub>X宏在结构上类似于 L<sup>A</sup>T<sub>E</sub>X中用 `\newcommand` 命令定义的对象。可以向它传递多达 9 个的参数值,这也与 L<sup>A</sup>T<sub>E</sub>X命令一样。然而,生成宏的相应 T<sub>E</sub>X命令要比 `\newcommand` 更具一般性。

事实上,大约有 900 条 Plain T<sub>E</sub>X命令可以使用,其中只有 300 条是原语,或者称为基本命令。其余 600 条都是宏。如果在宏中出现了错误,那么它里面的其它命令也可能受牵连。

为了明白起见,这里给出一个例子。命令 `\centerline` 就是如下定义的宏:

```
\def\centerline#1{\@@line{\hss#1\hss}}
```

这里 `\@@line` 本身还是一个宏,而 `\hss` 为一个 T<sub>E</sub>X原语,它是一个橡皮长度,可以无限伸展或收缩。为了避免误导读者进入太深层的 T<sub>E</sub>X命令,这里我们只是指出上面定义的宏就基本上等价于 L<sup>A</sup>T<sub>E</sub>X命令序列:

```
\newcommand{\centerline}[1]{\makebox[\textwidth][c]{#1}}
```

下面给出示例文本:

```
\documentclass{article}
\begin{document}
\centerline{This is an \invalid command}
\end{document}
```

其中在单词 `invalid` 前面加了一个 `\`,从而生成一个错误命令 `\invalid`。

在 L<sup>A</sup>T<sub>E</sub>X处理它时,会给出如下 T<sub>E</sub>X错误消息:

```
! Undefined control sequence.
<argument> This is an \invalid
                                command
1.3 ...erline{This is an \invalid}
```

?

这条消息现在应该很容易理解的。错误指示同 9.1.1 节例子中的一样:

```
! Undefined control sequence.
```

接下来两行文本说明是在处理完 `\invalid` 命令时发现错误的，接下来要读入的文本是单词 `command`。同时，在上面那行开头部分的 `<argument>` 说明这部分文本是某个命令的参数值。

再下面两行文本类似的：错误出现在输入文本的第 3 行，而且整行文本（包括命令和参数值）都已经被读入并进行了处理。

## §9.2 一些错误样例

### §9.2.1 错误的传播

有不正确 `\begin{quote}` 环境的例子表明如果只是给出一个简单地反应（回车），尽管 `\end{quote}` 命令是正确的，也会导致第二个错误消息。这种错误地纠正导致进一步错误的现象要比想像中多很多。

现在看一些源文本：

```
\documentclass{article}
\begin{document}
\begin{itemie}
  \item This is the first point in the list
  \item And here comes the second
\end{itemize}
\end{document}
```

这部分文本中的唯一错误就是把环境名 `itemize` 错写为 `itemie`。在  $\text{\LaTeX}$  处理时，首先生成与前面不正确的 `quote` 环境中一样的错误消息：

```
! LaTeX Error: Environment itemie undefined
```

```
See the LaTeX manual or LaTeX Companion for explanation.
```

```
Type H <return> for immediate help.
```

```
...
```

```
1.3 \begin{itemie}
      \slshape
?
```

这时候，如果用户反应为 `I \begin{itemize}`，那么就会纠正这个错误，从而处理过程很顺利地结束。然而，如果输入的是（回车），那么就会得到一条新的错误消息：

```
! LaTeX Error: Lonely \item--perhaps a missing list environment.
```

```
See the LaTeX manual or LaTeX Companion for explanation.
```

Type H <return> for immediate help.

...

#### 1.4 \item T

his is the first point in the list

?

之所以会出现这样的结果，是因为没有 `\begin` 命令，那么这就是在列表环境外面用了 `\item`，这样它没有任何意义。（实际上，它在这里是有意义的：它显示出了一条错误消息！）现在要插入不存在的 `itemize` 环境开始部分就太晚了。输入 H(回车) 可以得到如下帮助：

Try typing <return> to proceed.

If that doesn't work, type X <return> to quit.

?

遵照这条建议，按了 (回车) 键，那么就会又得到同样的错误消息，但这次是出现在第 5 行上，相应于第二条 `\item` 命令。继续按 (回车)，就会得到：

! LaTeX Error: \begin{document} ended by \end{itemize}

See the LaTeX manual or LaTeX Companion for explanation.

Type H <return> for immediate help.

...

#### 1.6 \end{itemize}

?

现在 `itemize` 环境总算结束了，但是因于其开头不正确，`LaTeX` 抱怨遇到了不匹配的 `\begin` 和 `\end` 命令。在这里最后一次按 (回车)，就会使处理继续进行下去。当然列表环境中的内容不会有正确的格式的，但是其它部分的文档不会受到影响。

在这个例子中，源文本的一个错误生成了三个其它错误消息。这并不是不常见的。有些 `LaTeX` 错误可以导致上百条后续错误。甚至有可能错误链永不会中止，处理过程不再向前进展。在这种情况下，没有别的办法了，只能终止程序。这可以在错误消息后面输入反应 `I\stop` 来做到。有可能需要给出几次这种反应，其才会发生作用。如果这还行不通，也就是说每次还是出现同样的错误消息，那么用反应 `X(回车)` 就可以马上结束程序的运行。

用 `I\stop` 要比用 `X` 结束程序好，因为这样在输出中会包含最后一页的结果。这对于要推断错误来源时是非常有用的。

本节最后要告诉你的一条经验就是：即使遇到成群的错误，也不要惊慌！

坚持按〈回车〉键，继续处理下去。

如果按的是 S〈回车〉会在屏幕上得到同样的错误消息，但是其间不会再停下来等待用户的反应（9.1.1 节）。

### §9.2.2 典型的严重错误

有的时候，用户可能忘记了 `\documentclass` 或 `\begin{document}` 命令，或者忘记了整个导言。当一个 L<sup>A</sup>T<sub>E</sub>X 文件本来是要用 `\input` 或 `\include` 读入的，但错误地直接用 L<sup>A</sup>T<sub>E</sub>X 程序调用它，后者这种错误更容易发生。如果一个文件中的文本为

```
This file has no preamble.
```

它被直接用 L<sup>A</sup>T<sub>E</sub>X 进行处理，那么就会在屏幕上显示如下错误消息：

```
! LaTeX Error: Missing \begin{document}.
```

```
See the LaTeX manual or LaTeX Companion for explanation.
```

```
Type H <return> for immediate help.
```

```
...
```

```
1.1 T
```

```
his file have no preamble.
```

```
?
```

在输入 H〈回车〉后得到的相应帮助消息为：

```
You're in trouble here. Try typing <return> to proceed.
```

```
If that doesn't work, type X <return> to quit.
```

```
?
```

从这条错误消息我们知道 L<sup>A</sup>T<sub>E</sub>X 在读入第一个字母的时候就发现了一个错误。这里的帮助消息也不怎么样。在这里持续按〈回车〉是没有什么用的。反而我们应该用 X 或 E 马上停下来，因为这里得不到任何有用的东西了。

即使上面的例子文件中包含如下环境：

```
\begin{document}
```

```
This file has no preamble.
```

```
\end{document}
```

也不可能得到一个正确的处理。现在的 T<sub>E</sub>X 错误消息为：

```
! LaTeX Error: The font size command \normalsize is not defined:
there is probably something wrong with the class file.
```

```
See the LaTeX manual or LaTeX Companion for explanation.
```

```
Type H <return> for immediate help.
```

...

### 1.1 `\begin{document}`

?

这里的错误指示相当古怪，因为在输入中从没有用 `\normalsize`。只有消息中的最后两行还好理解。它们说明是在读入第 1 行的 `\begin{document}` 命令后发现错误的。实际上从文本的第一行就可以知道这个处理不会得到任何结果，因此在它前面并不存在不可缺少的 `\documentclass` 命令。

这时候，没有别的选择，就只能用 **X** 或 **E** 终止程序，因此继续处理下去也没有任何意义。

(这里之所以出现如此奇怪的错误指示，是因为有许多格式化参数需要用 `\begin{document}` 声明进行初始化，其中就包含标准字体。因此这里是内部调用了 `\normalfont`，这条命令必须在类文件中进行定义，在这里并不存在类文件。而其它的初始化命令是在  $\text{\LaTeX}$  格式中定义的，因此它们不会导致错误。)

如果文档类的名称输入错误，例如：

```
\documentclass{fred}
```

那么就会得到如下错误消息：

```
! LaTeX Error: File 'fred.cls' not found.
```

```
Type X to quit or <RETURN> to proceed,
or enter new name. (Default extension: cls)
```

```
Enter file name:
```

这条消息应该是相当明白的：程序正在寻找一个叫 `fred.cls` 文件，但没有找到，因此它希望用户输入另外一个名称。如果新的文件具有所要求的扩展名 (`.cls`)，那么就不需要显式给出。在现在情形中，任何标准  $\text{\LaTeX}$  类的名称，如 `article`, `report`, `book` 或 `letter`，都可以做为输入，当然也可以输入其它可用类的名称。当然，文档应该是相应于该类而编写的。

只要系统找不到要读入的文件，就会显示同样的错误消息。会导致这个问题的命令有 `\input`, `\include` 或 `\usepackage`。如果这个文件确实不存在，那么就可能是因为它并不位于  $\text{\TeX}$  寻找文件的地方。通常在安装时要建立一个系统参数，告诉  $\text{\TeX}$  在哪儿找文件。如果你的文件在其它地方，那就应该输入完整的文件名，即包含目录或路径标识符。

如果  $\text{\LaTeX}$  要求你输入的文件并不存在，或者是你根本就不知道的文件，那你就应该马上终止处理或者告诉  $\text{\LaTeX}$  忽略这个烦人的文件。这时，你还有

一个问题。L<sup>A</sup>T<sub>E</sub>X 坚持要求得到一个合法的文本名。输入 `X`(回车) 只是还会得到同样的消息，这时它会说它找不到文件 `x.ext`。有些安装版本中提供了一个无内容的文本，叫 `null.tex`，而且还可以用一个扩展软件包 (D.3.4 节)，它提供了一组文件，依据标准反应字母而命名为 `x.tex`, `e.tex`, `r.tex`, `h.tex`, `s.tex`，这样来模拟处理普通错误消息时的反应。

紧急停止：有时候我们会遇到一种情形，即使用 `I\stop` 或 `X` 也不能在出现错误后令程序终止。这时，就必须借助于操作系统的程序中断了。细节请看计算机中心或 PC 机的手册。

### §9.2.3 数学错误

对数学公式命令的不正确应用，会导致多得令人吃惊的错误，这一点对没有经验的用户更是如此。更经常见到的在数学公式中的错误是无意识的错误，如忘记了一个右大括号 `}`，或者没有及时切换回文本模式。另外一种常见的错误是在文本模式中用了只能出现在数学模式中的符号。下面我们给出几个典型的例子。

如果我们想得到：‘The price is \$3.50 and the order number is type\_sample’，那么可能会输入如下源文本：

```
The price is $3.50 and the order number is type_sample.
这块文本中包含两处错误，而且第一个错误取消了第二个错误：$ 符号是在
文本中生成数学公式的数学模式切换开关 (5.1 节)。在文本中生成真正的美元
符号是输入 \$. 然而，在这段示例文本中，单独的 $ 是切换进入数学模式，
把其后的内容做为公式处理。然而，这里没有关闭的切换开关 $，当这段结束时
TEX 才会注意到这一点。
```

```
! Missing $ inderted.
```

```
<inserted text>
```

```
$
```

```
1.5
```

```
?
```

如果只是用 (回车) 做为反应，T<sub>E</sub>X 会在该点插入一个 \$。对于我们的示例文本，这就是在空行前面的文本结尾时插入的。也就是说从第一个 \$ 到这段结束之间的文本都被当做正文公式处理了：

```
The price is 3.50andtheordernumberistype_s ample.
```

从这个结果马上就可以看出用户犯的错误了。在靠近行的尾部，有一个字母 `s` 显示为下标了。这就是该例中的第二个错误。下划线字符 `_` 只能用在数学模式中，这里应该输入为 `\_`。然而，由于第一个 \$ 已 (不正确地) 切换进入数学模式，`_` 符号成为合法的了，这样其后字母 `s` 就成为下标了。

如果现在在文本中用 `\$` 代替 `$`，那么 `_` 符号就不是处于数学模式中，会生成如下错误消息：

```
! Missing $ inserted.
<inserted text>
      $
1.5 ...$3.50 and the order number is type_
                                     sample.
?
```

在这里按〈回车〉，告诉 TeX 在数学命令 `_` 前面插入 `$`，从而改正这个错误。这样处理过程就会继续下去，当到了当前段结束前的某处，TeX 就会注意到并没有闭 `$` 符号。这样就会显示与前面那种情形相同的错误消息。再按一次〈回车〉键，处理就会继续，得到如下结果：

The price is \$3.50 and the order number is type<sub>sample</sub>.

在所有这三种情形中，都可以用 H 寻求更多的帮助，内容为

```
I've inserted a begin-math/end-math symbol since I think
you left one out. Proceed, with fingers crossed.
```

上面最后一句话就是我们所推荐的处理数学错误的方法：持续按〈回车〉或者 S，以使处理完成，这时再看打印的结果，以找到错误。

#### §9.2.4 来自于多文件的错误

如果文档被分成许多文件，每个用 `\input` 或者 `\include` 命令读入，那么错误消息中的行号相应于正在读的文件。利用反应 E〈回车〉，将会调用编辑器程序，并打开文件，跳到发现错误的指定行。利用其它反应，那就必须单独使用编辑器，而且显式地给定出错文件名称。

通过查看处理消息或者 `.log` 文件，可以知道出错时正在处理的是哪个文件。当文本被打开供读取时，TeX 会在屏幕或 `.log` 文件中写一个左小括号（以及文件的名称。当文件被关闭时，会写右小括号）。输出页码还是像通常那样显示在中括号内，这些内容都是同时显示在屏幕和写进 `.log` 文件中的。例如，如果屏幕上有如下处理消息：

```
..(sumfile.tex [1] [2] [3] (part1.tex [4] [5]) (part2.tex [6] [7]
! Undefined control sequence
1.999 \finish
?
```

那么可以有如下解释：一个叫 `sumfile.tex` 的文件正在被读取，而且在输出完第 1,2,3 页后，文本 `part1.tex` 被读取（在文件 `sumfile` 中用 `\input` 和 `\include` 命令），这时输出第 4,5 页，然后关闭 `part1.tex`，接着打开 `part2.tex` 文件做输入。在这个文件的第 999 行上发现一个错误。如果从键

盘上改正了这个错误，或者如果处理过程还是继续下去了，屏幕上会有如下消息：

```
[8] [9]) [10]
! Too many }'s
1.217 \em sample}
```

在第 9 页后面的右小括号表示 `part2.tex` 文件已被关闭。接下来在主文件 `sumfile.tex` 的第 10 页上又遇到一个错误，因为这里并没有 `)` 表示该文件已被关闭。错误是在这个文件的第 217 行上遇到。

### §9.3 L<sup>A</sup>T<sub>E</sub>X 错误消息清单

下面列出 L<sup>A</sup>T<sub>E</sub>X 错误消息，分成一般、宏包和字体错误消息三组。在每组中，按错误指示的字母顺序排序。然后给出可能导致该错误的描述和解决方法。

有许多消息是新出现在 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 中的，也有一些相比于 L<sup>A</sup>T<sub>E</sub>X 2.09 版本稍有些改变。在 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 中的所有消息都有一个前缀词 `! LaTeX Error:`，而在 2.09 版本中错误指示只是前缀惊叹号，并且位于第三行上。如果这是两者的唯一差别，那么我们就显示 2.09 版本的内容。

当不同版本有其它不同时，这里还要用通常的标志来指明。

#### §9.3.1 一般 L<sup>A</sup>T<sub>E</sub>X 错误消息

下述错误消息来源于那些没有调用字体选择或者定义，或者不涉及 L<sup>A</sup>T<sub>E</sub>X 程序类和宏包文件（附录 C）的特殊功能时所出现的问题。

```
! LaTeX Error: ... undefined.
```

`\renewcommand` 或者 `\renewenvironment` 的参数值前面还没有定义。应在前面插入相应的 `\new...` 命令。

```
! LaTeX Error: \< in mid line.
```

`tabbing` 环境中的 `\<` 命令却出现在一行的中间。这条命令只能位于一行的开头（4.6.3 节）。

```
! LaTeX Error: Bad \line or \vector argument.
```

`\line` 或 `\vector` 命令的第一个参数值指定直线和箭头的倾角。这条消息说明所选择的角度非法。见 6.4.3 和 6.4.4 节。

```
! Bad use of \\. (LATEX 2.09)
```

2.09 (这条消息在 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 中已经被 `There's no line here to end` 取代。)

```
! LaTeX Error: Bad math environment delimiter.
```



L<sup>A</sup>T<sub>E</sub>X在不正确的模式中遇到了数学切换命令：即数学模式中遇到了 \[ 或 \（，或者文本模式中遇到了 \] 或 \）。这是因为数学模式切换开关的配对不正确，或者某些大括号对 {...} 不匹配。

```
! LaTeX Error: \begin{...} on input line ... ended by \end{...}.
! \begin{...} ended by \end{...}. (LATEX2.09)
```

L<sup>A</sup>T<sub>E</sub>X遇到了一条 \end 命令，其没有与之对应的同名 \begin 命令。这可能是由于把环境名输入错了，或者前面漏掉了一个 \end 命令。避免这种错误的好方法就是一旦开始 \begin 命令，就马上输入 \end 命令，然后再在这两者之间插入实际的环境文本。对于长的嵌套环境，这一方法相当有用。这样也减少了在 \end 命令中输错环境名的机会。

```
! LaTeX Error: Can be used only in preamble.
```

有许多命令只能用在导言中。它们是：\documentclass, \usepackage, \nofiles, \includeonly, \makeindex, \makeglossary 以及其它一些。有些命令只有在类或宏包文件中才有意义，如 \ProvidesClass 和 \ProvidesPackage（C.2.1 节），还有许多 \Declare.. 和 \Set.. 命令也只能用在导言中。如果在 \begin{document} 之后调用了这些命令，就会显示上面的消息。

```
! LaTeX Error: Command ... invalid in math mode.
```

<sup>2ε</sup>在数学模式中调用了一条命令，而这条命令只有在文本模式中才有意义，如 \item 或者 \circle。L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 的字体命令 \itshape, \bfseries 等等在数学模式中也会导致这种错误，因为这时应该用的是数学字母表命令 \mathit, \mathbf 等。

```
! LaTeX Error: Command ... already defined.
```

```
! Command name ... already used. (LATEX2.09)
```

用户利用 \newenvironment, \newcommand, \newtheorem, \newsavebox, \newfont, \newlength, \newcounter 或 \DeclareMathAlphabet 等命令重定义已存在的结构。这时可以另选择一个不同的名称，或者当处理的是命令或环境时，可以用 \renew... 形式的命令。（注意当定义了一个名为 sample 的环境时，也同时创建了 \sample 和 \endsample 命令。）

```
! LaTeX Error: Command ... undefined in encoding ....
```

<sup>2ε</sup>指定的命令只是已经相应于某一种 NFSS 编码（比如说是 OT1）利用命令 \DeclareTextCommand 进行了定义，但是却在另一种编码（比如说是 T1）中使用它，因为在后一种编码中并没有相应的定义。

```
! LaTeX Error: Counter too large.
```

要以字母形式显示其值的计数器中的值大于 26。

```
! LaTeX Error: Environment ... undefined.
```

L<sup>A</sup>T<sub>E</sub>X 遇到了一条 `\begin` 命令, 它不知道该环境名称。这可能是由于输入错误造成的。只要在处理时用反应 I 跟上正确的名称就可以校正这个错误。(这并没有改变源文件, 错误仍然还在那儿。)

```
! LaTeX Error: File '...' not found
Type X to quit or <RETURN> to proceed,
or enter new name. (Default extension: ...)
Enter file name:.
```

<sup>2ε</sup> 用输入命令上载一个文件, 但是却没有找到这个文件。这里可以输入一个新的文件名, 或者退出, 或者不管它继续处理下去。如果给出了正确的文件名, 而这个文件不存在, 那么它可能是由于它不处在 L<sup>A</sup>T<sub>E</sub>X 寻找文件的地方。确保它在其中的一个正确目录中。此时用户有机会从键盘输入一个替代文件, 或者改正输入错误。输入文件名, 加上可省的扩展名, 然后按 (回车)。L<sup>A</sup>T<sub>E</sub>X 给出一个与要求相同的可省扩展名。文件名的基本部分并没有默认值。

```
! LaTeX Error: Float(s) lost.
```

`figure` 或 `table` 环境或者 `\marginpar` 命令在垂直盒子 (`\parbox` 命令或者 `minipage` 环境) 中给出, 或者这些结构出现在 L<sup>A</sup>T<sub>E</sub>X 命令中, 而这条命令内部被垂直盒子调用, 例如脚注。L<sup>A</sup>T<sub>E</sub>X 是在输出一页后才会发现这个错误, 因此实际的来源可能在指定有错文本前面好几行。这种错误的结果是有许多表格、插图或者边注丢失, 但并不是它们导致这个错误。

```
! LaTeX Error: Illegal character in array arg.
```

`tabular` 或 `array` 环境中包含一个未知的列格式字符 (见 4.8.1 节), 或者在 `\multicolumn` 命令中做为第二个参数值的格式项有错。

```
! LaTeX Error: \include cannot be nested.
```

<sup>2ε</sup> 在一个已经要用 `\include` 命令读入的文件中又用了命令 `\include`。所有的 `\include` 命令都必须是在主文件中调用 (8.1.2 节)。

```
! LaTeX Error: LaTeX2e command ... in LaTeX 2.09 document.
```

<sup>2ε</sup> 如果正处在兼容模式 (即文档中用的是 `\documentstyle` 命令, 而不是 `\documentclass` 命令), 却使用了 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 命令, 就会出现这种错误。这些命令有 `\LaTeXe`, `\usepackage`, `\ensuremath`, `lrbox` 环境, 还有相应于 `\newcommand` 和 `\newenvironment` 命令增强的一些新语法。出错的原因是在兼容模式中作者可以确保文档适用于 L<sup>A</sup>T<sub>E</sub>X 2.09, 这样可以把它送给只有原来版本的用户手中。

```
! LaTeX Error: Lonely \item--perhaps a missing list environment.
```

在列表环境 (4.3 节) 外面用了 `\item` 命令。这要么是因为把 `\begin` 中的环境名拼写错了, 并且没有从键盘上改正它, 要么是忘记了 `\begin` 命令。

! LaTeX Error: Missing @-exp in array arg.

`tabular` 或 `array` 环境的列格式化参数值包含了 `@` 符号, 但后面没有必须的放在大括号 `{ }` 内的文本 (关于 `@` 表达式见 4.8.1 节), 或者在 `\multicolumn` 命令中的第二个参数里发生同样的错误。

! LaTeX Error: Missing begin{document}.

这要么是忘记输入 `\begin{document}` 命令, 要么是在文档的导言中有可打印的文本。对于后者, 这有可能是由于一个不合语法的声明造成的, 例如命令的参数值没有放在大括号 `{ }` 内, 或者命令名前面没有 `\`。

! LaTeX Error: Missing p-arg in array arg.

`tabular` 或 `array` 环境中的列格式化参数值包含 `p` 符号, 但没有必须与其一起出现的宽度定义 (4.8.1 节), 或者在 `\multicolumn` 命令的第二个参数值中发生同样的错误。

! LaTeX Error: No counter '...' defined.

! No such counter. (L<sup>A</sup>T<sub>E</sub>X 2.09)

调用 `\setcounter` 或者 `\addtocounter` 命令, 但所引用的计数器并不存在。这很可能是名称被输错了。如果发生错误时正在读 `.aux` 文件, 而且计数器的名称肯定正确, 那么 `\newcounter` 命令就可能是在导言外面。因此我们强烈建议总是把 `\newcounter` 命令放在导言中。(如果其它的 L<sup>A</sup>T<sub>E</sub>X 计数器命令用的是一个没有定义的计数器名称, 那么就会显示出一长串有趣的 T<sub>E</sub>X 错误消息。)

! No theorem environment '...' defined. (L<sup>A</sup>T<sub>E</sub>X 2.09)

<sup>2.09</sup> (在 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 中, 这条消息被 `No counter defined.` 代替)

在 `\newtheorem` 命令中给出了可省参数值, 它指出另一个定理类型的声明, 以使两者共享同一个计数器, 但是这另一个定理结构并不存在 (4.5 节)。这要么是写错了定理的名称, 要么就是前面还没有定义。

! LaTeX Error: No \title given.

<sup>2ε</sup> 在给出 `\title` 声明之前就用了 `\maketitle` 命令。

! LaTeX Error: Not in outer par mode.

把 `figure` 或 `table` 环境或者 `\marginpar` 命令放在数学模式或者竖直盒子 (`\parbox` 命令或者 `minipage` 环境) 里面, 会导致这种错误。对于第一种情形, 通常是由于忘记了数学切换开关命令。

! LaTeX Error: Page height already too large.

<sup>2ε</sup> 用 `\enlargethispage` 命令来扩大页面的竖直尺寸, 但是 L<sup>A</sup>T<sub>E</sub>X 认为这个尺寸太大了。

! LaTeX Error: \pushtabs and \poptabs don't match.

在 tabbing 环境中 \poptabs 命令与前面已经出现的 \pushtabs 命令在数目上不一样 (4.6.4 节)。

! LaTeX Error: Something's wrong--perhaps a missing \item.

导致这个问题的原因很可能是由于列表环境 (list, enumerate 或者 description) 中的文本不是由 \item 命令开始的。如果在 thebibliography 环境中没有参数值 {样本标签} (4.3.6 节) 时也会出现这种错误。

! LaTeX Error: Suggested extra height (...) dangerously large.

<sup>2ε</sup> 当用 \enlargethispage 命令扩大页面的竖直尺寸时, 超出了 L<sup>A</sup>T<sub>E</sub>X 认为合理的范围。

! LaTeX Error: Tab overflow.

在 tabbing 环境中最后一个 \= 命令超过了在 L<sup>A</sup>T<sub>E</sub>X 可允许的制表位数目。

! LaTeX Error: There's no line here to end.

! Bad use of \\. (L<sup>A</sup>T<sub>E</sub>X 2.09)

在 \par 或者空行后面调用命令 \newline 或者 \\, 因为在这里它们没有任何意义。如果要在这一行插入额外竖直间距, 应该用 \vspace 命令。

! LaTeX Error: This may be a LaTeX bug.

这条消息说明 L<sup>A</sup>T<sub>E</sub>X 已完全糊涂了。这可能是由于前面出现错误时, 用户的反应是按了 (回车)。对于这种情形, 应该用 I\stop, X 或者 E 把处理过程终止下来, 纠正前面的错误。虽然有可能 (但可能性不大) 这确实是 L<sup>A</sup>T<sub>E</sub>X 本身的一个漏洞。如果这是处理过程中的第一条错误, 而文本看起来却令人满意, 那么这个文件应该保存起来, 交给计算中心, 以供进一步研究。

! LaTeX Error: Too deeply nested.

把列表环境 (description, itemize, enumerate 或者 list) 彼此嵌套的层次太多。可以这样嵌套的最多层数与安装版本有关, 但应至少为四层。

! LaTeX Error: Too many columns in eqnarray environment.

<sup>2ε</sup> 在 eqnarray 环境中每行只能有三列。这可能是你忘记用 \\ 开始新行了, 或者在一行里多放了 &。

! LaTeX Error: Too many unprocessed floats.

如果在一页上有太多的 \marginpar 命令时会出现这个错误。然而, 更可能是 L<sup>A</sup>T<sub>E</sub>X 维持了超过其能力的太多的 figure 和 table 浮动对象。这种情况可能发生在浮动对象被输出之前, 提交了太多对象 (6.6 节)。对于这种情况, 就应该在文本中稍后加进最后这个对象。另外一种可能的原因是插图或者

表格在正常的页面中放不下来，而它适合于放在文本结尾的特殊浮动页或者 `\clearpage` 或 `\cleardoublepage` 命令后面。由于插图和表格的输出顺序与输入顺序是相同的，因此这样的对象就会阻塞整个队列。而 `\clearpage` 或 `\cleardoublepage` 命令就可以化解这种阻塞。

! LaTeX Error: Undefined tab position.

在 `tabbing` 环境中用 `\>`, `\+`, `\-` 或者 `\<` 命令把制表位移动到了一个不存在的地方（4.6 节）。

! LaTeX Error: \verb ended by end of line.

<sup>2ε</sup>行内原文照排命令 `\verb+...+` 中的文本超过一行文本。在  $\text{\LaTeX}2.09$  中这是可以的，但在  $\text{\LaTeX}2_{\epsilon}$  中，这是禁止的，因为这样很容易捕捉一个常见的错误：少掉了结束字符。为了避免这种错误，确保在初始字符和结束字符之间的文本都在一个输入行上。

! LaTeX Error: \verb illegal in command argument.

<sup>2ε</sup>`\verb` 命令不能用在除 `\index` 和 `\glossary` 之外所有命令的参数值中。例如它就不能用在章节标题或者脚注中。

### §9.3.2 $\text{\LaTeX}$ 宏包错误

那些用来控制类和宏包文本的特殊程序设计命令（附录 C）有它们自己的错误消息集。如果这些功能出现了个严重错误，那么很少有用户能解决这个问题，只有把它告诉文件的作者了。另一方面，有些错误是源于对所提供的文件和选项不正确使用造成的。

通常类和宏包遇到其感到奇怪的文本和内容时，会有其自己的错误或警告消息。在结出这些消息时，会有类或宏包的名称。例如，宏包 `mypack` 可能显示出如下错误：

```
Package mypack Error: cannot mix options 'good'
(mypack)                and 'bad'.
```

当按了 `H` 时，应该会给出帮助消息。显然我们不可能在这里解释这些错误（警告）消息，因为它们完全与所考虑的宏包有关。

! LaTeX Error: \LoadClass in package file.

<sup>2ε</sup>在宏包文件中调用了 `\LoadClass` 命令，这是不允许的。只能从一个类文件调用另一个类文件。

! LaTeX Error: Option clash for package ....

<sup>2ε</sup>指定的宏包被重复调用，而且两次所用的选项不同。宏包文件只能被调用一次，第二次调用被忽略。因此，如果 `\usepackage` 或 `\RequirePackage` 命令调用的是同一个宏包，但是选项不同，这就会导致冲突。此时需要尽力安排一致的选项。按 `H` 可以得到帮助，它会显示出这两组冲突选项。

! LaTeX Error: \RequirePackage or \LoadClass in Options Section.

<sup>2ε</sup>这两条命令不能位于由 \DeclareOption 命令生成选项的类或宏包定义中。选项应该设置一些标志或者其它指示,以供在调用这里所考虑的命令之前进行测试。

! LaTeX Error: This file needs format ‘...’ but this is ‘...’

<sup>2ε</sup>\NeedsTeXFormat 命令指定了一个不同于正在用的格式。所谓格式,就是预先保存起来的指令集,用来确定 T<sub>E</sub>X 运行的类型。对于 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, 指定的格式名称必须是 LaTeX2<sub>ε</sub>。错误所指的意思是用当前的格式根本不能处理这个文件。

! LaTeX Error: Two \documentclass or \documentstyle commands

<sup>2ε</sup>一个文档中只能有一条 \documentclass 或者 \documentstyle 命令。如果在主文档中确实只看到一条这样的命令,那么检查一下其它上载的文件,确保其中没有与之冲突的第二条命令。

! LaTeX Error: Two \LoadClass commands.

<sup>2ε</sup>在类文件中包含多于一条的 \LoadClass 命令,而这是不允许的。所用的类文件一定在编写上有问题。

! LaTeX Error: Unknown option ‘...’ for package ‘...’.

<sup>2ε</sup>在 \usepackage 命令中指定了一个选项,但是相应的宏包中并没有定义这个选项。查看一下宏包的指令,以及有没有输入错误。

! LaTeX Error: \usepackage before \documentclass.

<sup>2ε</sup>\usepackage 命令不能出现在 \documentclass 的前面。必须在所有宏件包之前上载类文件。

### §9.3.3 L<sup>A</sup>T<sub>E</sub>X 字体错误

下面的错误可能在使用新字体选择框架(8.5 节)定义或选择字体时出现。这些消息中有些会说字体还没有正确地建立,那么这可能是因为字体描述文件已崩溃或者有错。对于这两种情形,系统管理员必须找一组正确的文件,进行重新安装。

! LaTeX Error: ... allowed only in math mode.

<sup>2ε</sup>类似于 \mathbf 这样的数学字母表命令用在了文本模式中。这可能是忘记了一个 \$。

! LaTeX Error: Command ‘...’ not defined as a math alphabet.

<sup>2ε</sup>\Set.. 或 \Declare.. 命令以数学字母表名称做为参数,但是当用这两个命令时,指定了一个不存在的字母表的名称。数学字母表名称在使用之前,必须先用 \DeclareMathVersion 命令进行声明。

! LaTeX Error: Command ... not provided in base LaTeX2e.

<sup>2ε</sup>有一些符号是基本 L<sup>A</sup>T<sub>E</sub>X 2.09 的一部分, 但 T<sub>E</sub>X 中并不存在, 那么它们不再自动包含在 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 中。这时需要利用软件包 `latexsym`。

! LaTeX Error: \DeclareTextComposite used on inappropriate  
command ....

<sup>2ε</sup>`\DeclareTextComposite` (8.5.9 节) 用来重定义特定编码中一个已存在的重音命令, 以显示单个字符。如果这个重音命令并不存在, 那就会给出这条错误。

! LaTeX Error: Encoding scheme '...' unknown.

<sup>2ε</sup>进行了一个使用不存在的编码方案的字体声明或者选择命令。这很可能是由于输入错误导致的。

! LaTeX Error: Font family '...+...' unknown.

<sup>2ε</sup>对字体编码和族组合调用了 `\DeclareFontShape` 命令, 但是这种组合之前并没有用 `\DeclareFontFamily` 命令进行声明 (8.5.8 节)。

! LaTeX Error: Font ... not found.

<sup>2ε</sup>找不到具有指定属性的字体, 也无法进行适当的替换。这时会由 `\DeclareErrorFont` 定义的字体。

! LaTeX Error: Math alphabet identifier ... is undefined  
in math version '...'.

<sup>2ε</sup>调用了一种对当前数学变体还没有定义的数学字母表 (8.5.5 节)。这就是说已经用 `\DeclareMathAlphabet` 创建了字母表, 其形状属性为空, 但没有用 `\SetMathAlphabet` 声明把它定义成可以在选定数学变体中可用。

! LaTeX Error: Math version '...' is not defined.

<sup>2ε</sup>`\Set...` 或 `\Declare...` 命令以数学变体名称做为参数, 但是当用这两个命令时, 指定的一个不存在的字母变体名称。数学变体名称在使用之前, 必须先用 `\DeclareMathVersion` 命令进行声明。

! LaTeX Error: \*\*\* NFSS release 1 command ... found  
\*\*\* Recovery not possible. Use ....

<sup>2ε</sup>用了第一版 NFSS 中的一条命令, 而这条命令现在不再可用了。用建议的替换命令。

! LaTeX Error: Not a command name: '...'.

<sup>2ε</sup>`\DeclareMathAccent` 命令的第一个参数值必须是一个命令名称, 如 `\acute`。这时很可能是忘记输入反斜杠 `\` 了。

! LaTeX Error: Symbol font '...' not defined.

<sup>2ε</sup>在`\Set..` 或`\Declare..` 命令中指定了一个不存在的符号字体名称, 这两条命令需要符号字体名称做为参数值。字体名称在做为符号字体使用之前, 必须用`\DeclareSymbolFont` 进行声明。

! LaTeX Error: The font size command `\normalsize` is not defined:

there is probably something wrong with class file.

<sup>2ε</sup>在类文件中必须定义`\normalsize` 命令, 它表示文档中文本的标准尺寸。这是根据 L<sup>A</sup>T<sub>E</sub>X 2.09 中的经验进行的一个修改。如果类文件中没有进行这个定义, 而只是定义了`\@normalsize`, 那就必须进行修正。如果没有`\documentclass` 命令, 也会出现这条消息, 因为空的类文件当然是错误的。

! LaTeX Error: This NFSS system isn't set up properly.

<sup>2ε</sup>在`.fd` 文件中的字体描述有错误, 或者没有由`\DeclareErrorFont` 声明的合法字体。这是一个严重的错误, 应向系统管理员报告这一信息。

! LaTeX Error: Too many math alphabets used in version ....

<sup>2ε</sup>只可能有多达 16 个的数学字母表, 这是由 T<sub>E</sub>X 本身设定的限度。其它数学字母表定义都会被忽略。

! LaTeX Error: Unknown symbol font '`...`'.

<sup>2ε</sup>用一个不存在的符号字体名称做为了`\DeclareSymbolFontAlphabet` 命令的参数值。这里的字体名称必须在使用前用`\DeclareSymbolFont` 进行声明。

## §9.4 T<sub>E</sub>X 错误消息

本节列出一些比较常见的 T<sub>E</sub>X 错误消息, 按错误指示进行字母排序, 而且同时给出一个导致该错误的简短描述。这里每一个前面只是以惊叹号开始。由于这些消息是由 T<sub>E</sub>X 生成的, 由此在 L<sup>A</sup>T<sub>E</sub>X 2.09 和 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 之间没有任何差别。

! Counter too large.

做为 T<sub>E</sub>X 错误, 这个消息是说当脚注标记用的是字母或符号时, 计数器的值超过了 26 或 9。如果在标题上有很多`\thanks` 命令时也可能出现这个错误。

! Double subscript.

在数学公式中一个变量后面有两个下标, 例如, `x_{2_3}` 或 `x_{2}_{3}`。而要得到  $x_{2_3}$  的正确方法是 `x_{2_3}` 或者 `x_{2_{3}}` (5.2.2 节)。

! Double supscript.





! Misplaced alignment tab character &.

不能把单个字符命令 `&` 放在 `tabular` 或 `array` 环境外面的普通文本中。可能的动机是本来想显示 `&`，那么就应该输入为 `\&`。在响应错误消息时，可以用 `I\&` 来订正这一点。

! Missing control sequence inserted.

这个错误很可能是由于在 `\newcommand`, `\renewcommand`, `\length`，或 `\newsavebox` 命令的第一个参数值前面漏掉了反斜杠 `\`。只要用 `<回车>` 做为反应，就会正常地结束处理过程，因为 T<sub>E</sub>X 假定你是少掉了反斜杠，从而替你补上了。

! Missing number, treated as zero.

这个错误很可能来源于需要一个数值或长度做为参数值的命令，而却少掉了这个参数值。或者也有可能是一条命令有可省参数值，而其后所接的文本开头就是 `[`，这也可能造成这种错误。最后一种可能是在长度或者 `\value` 命令前面有 `\protect` 命令。

! Missing { inserted.

! Missing } inserted.

当显示这两条消息之一的时候，就表明 T<sub>E</sub>X 已完全糊涂了。这时显示出来的行号往往并不是出现错误的地方，这是由于少掉个左大括号或右大括号造成的。如果错误不是很明显，那就按 `<回车>` 让处理继续下去，从打印出来的结果中推断错误的地方。

! Missing \$ inserted.

很可能是在普通文本中用了一个只能出现在数学模式中的符号或命令而造成这种错误。查看一下在第 5 章中列出的那些除非有特别说明，否则只能用在数学模式中的命令。如果在数学公式中插入了一个 `\mbox` 命令，那么其参数值就暂时从数学模式切换到文本模式。如果在数学公式中出现一个表示新段的空行，或者在公式的尾部没有必要的 `$` 符号也会出现这种错误。

! Not a letter.

`\hyphenation` 命令的单词列表中包含一个不被认为是字母的字符，例如 `\'e` 这样的重音命令。这样的单词可以通过显式插入可选连字符来给出建议断点（3.6.1 和 3.6.2 节）。

! Paragraph ended before ... was complete.

命令的参数值中包含空行或者 `\par` 命令，而这是不允许的。很可能是漏掉了右大括号 `}`。

! \scriptfont ... is undefined (character ...).

! \scriptscriptfont ... is undefined (character ...).

`! \textfont ... is undefined (character ...).`

当在  $\text{\LaTeX}2.09$  中数学公式使用了一个符号，其字体并不是针对于数学模式而设计的，如 `\sf(sans serif)`，就会显示这种消息。这样的字符字体必须用 `\load{尺寸}{样式}` 命令才可以使用，如 `\load{\normalsize}{\sf}`。如果这样的公式出现在脚注中，就必须事先给出 `\load{\footnotesize}{\sf}` 命令。在  $\text{\LaTeX}2_{\epsilon}$  中，只要用的是 `\mathsf` 这样的数学字母表命令，就不会再出现这种问题。

`! TeX capacity exceeded, sorry [...].`

$\text{\TeX}$ 在进行文本处理时，会在计算机内存中建立起各种存贮缓冲区。当这些缓冲区有的已满了，从而不能再用时，就会显示这条消息。缓冲区的名称及其最大尺寸显示在错误指示的中括号内。显示过这条消息， $\text{\TeX}$ 的处理被终止。这个问题的来源几乎从不会是内存不足，无论处理的文本有多复杂，而多是文本自身的错误。在 9.6 节描述的方法可以用来探索真正的错误原因。

下面描述一下各种缓冲区，来帮助你确定是否真的是因为  $\text{\TeX}$ 分配缓冲区能力太小，以及应该怎样校正这个错误。

**buffer size** 这个问题可能是由于章节命令、`\caption`、`\addcontentsline` 或 `\addtocontents` 命令中的参数值太长造成的。这条消息通常只有当 `\end{document}` 命令被处理时才会出现，但是在遇到 `\tableofcontents`、`\listoffigures` 或 `\listoftables` 时也有可能出现。避免这种错误的方法就是对太长的标题文本，利用表示短标题的可省参数值 (3.3.3 和 6.6.4 节)。因为这样的太长的项出现在目录表也是很烦人的，总是要把它缩短。一旦在源文本中进行了这种修正，必须删除已有的  $\text{\LaTeX}$  .aux 文件。

在 PC 机上如果用的是字处理程序，而不是文本编辑器来生成源文本，有可能出现这个问题。因为有的这样程序把整个段落放在单独一行上，即使显示在屏幕上是分成几行的。

**exception dictionary** 用 `\hyphenation` 命令输入的连字符例外清单太大了。那些使用频率很小的单词应该去掉，而用 - 进行显式地指定单词断点。

**hash size** 在源文件中包含了太多的命令定义或者交叉引用标志。这并不是说源文件需要所有这些命令，而可能是由于用户开发出一个很大的自己专用的命令集，并把它存贮在一个文件中，读入到每个文档中，而不管用不用到这些命令。

**input stack size** 这个缓冲区的溢出可能是命令定义中出现了错误。例如，如果用下面这样的命令定义：

```
\newcommand{\com}{One more \com}
```

就会得到 `One more {One more {...One more \com}...}`，永远这样持续下去，因为它自己调用了自己。实际上它并不会永远持续下去，直至

缓冲区变满。

**main memory size** 这个缓冲区包含当前正在处理页面中的文本。如果调用了递归定义的命令，它也会溢出。然而，更常见的原因是：(1) 在一页上定义了很多相当复杂的命令；(2) 在一页上有太多的 `\index` 或 `\glossary` 命令；(3) 页面自身太复杂，以致于充满分配的缓冲区空间。

对于前两种情形的解决方法是非常简单的：减少该页的命令定义和 / 或 `\index` 与 `\glossary` 命令。对于第三种情形，原因可能是由于有太长的 `tabbing`, `texttttabular`, `array` 或 `picture` 环境，或者等待输出的浮动对象 (插图或表格) 发生了堵塞。

为了验证内存溢出是否确实是由于复杂的页面造成的，那么在出现溢出的地方前面加上一条 `\clearpage` 命令。如果在接下来的处理运行中不再出现这种错误，那么就可以确定这真的是由于该页面对 T<sub>E</sub>X 而言确实太复杂了。然而，如果仍然有溢出，那么就可能是由于源文本中的错误造成的。必要的话，可以用 9.6 节给出的方法确定错误的位置。

如果确实是由于对 T<sub>E</sub>X 处理而言，页面太复杂了，那就必须想法进行简化。然而要注意到在输出页面之前要把最近的那整个段落处理完，即使分页点可能出现在这个段的开始部分。利用 `\newpage` 命令，可能会解决这个问题，因此应该在进行烦人的文本重组织之前，尝试一下这条命令。如果错误是由于浮动对象阻塞造成的，那么把浮动对象移到后面文本中，或者改变一下定位参数值 (6.6.1 节)，可能会解决这个问题。如果整个文本还没有结束，那么现在可以尝试用 `\clearpage`，以清除被阻塞的浮动对象，然后在完成文本后给出浮动对象的一个更合理的顺序。

**pool size** 很可能是由于有太多的命令定义和 / 或标签，或者是其名称太长。

缩短一下名称长度看看。如果在诸如 `\setcounter`、`\newenvironment` 或 `\newtheorem` 等命令中的记数器命令参数值漏掉了右大括号 `}` 时也会出现这种错误。

**save size** 当命令、环境和声明的范围嵌套层数太多时，这个缓冲区会溢出。

例如，`\multiput` 命令的参数值中包含 `picture` 环境，而这个环境中又在另一个 `\multiput` 命令中用了 `\footnotesize` 声明，以此类推。这样的嵌套必须简化，除非实际导致问题的原因是因为忘记了右大括号 `}`，而使得结构变得相当复杂。

**! Text line contains an invalid character.**

在源文本中包含一个特殊符号，T<sub>E</sub>X 无法识别。这可能是编辑器本身的问题，是它插入了这个多余的字符。如果检查源文本时无法看到这个字符，那只有寻求计算中心的帮助了。

`! Undefined control sequence.`

每个 T<sub>E</sub>X 用户都会在经常遇到这条错误消息。它通常是由于不正确输入命令名称造成的。可以在处理过程中用 I 及正确的命令名称，再加上 (回车) 进行更正。如果正确地输入了一个 L<sup>A</sup>T<sub>E</sub>X 命令，那就可能是由于在不正确的环境中调用造成的，因为在这个环境中这条命令是不允许使用的 (即还没有定义)。

`! Use of ... doesn't match its definition.`

如果 ‘...’ 是一个 L<sup>A</sup>T<sub>E</sub>X 命令的名称，那它就很可能是 6.3 和 6.4 节中的图形环境中的命令调用时参数值语法不对。如果名称是 `\@array`，那么在 `tabular` 或 `array` 环境中有一个有错的 `@-` 表达式 (4.8.1 节)。这可能是由于在 `@-` 表达式中有一个脆弱命令，但没有前缀 `\protect` 命令。

`! You can't use 'macro parameter #' in ... mode.`

在普通文本中用了特殊符号 #。这可以是需要用 `\#` 以生成 # 自身。可以在处理过程用 I`\#` 加 (回车) 纠正这个错误。

## §9.5 警告

T<sub>E</sub>X 和 L<sup>A</sup>T<sub>E</sub>X 错误都会使得处理过程暂时停下来，并等待用户的反应，或者程序完全中止。而另一方面，警告只是提示用户输出结果中可能有些不对头，需要进一步修正。警告消息是连同其所在的页码一起显示在屏幕上，但不是会使处理过程停下来。当然这些消息也会同时写到 `.log` 文件中，这样处理结束后可以查看该文件，甚至打印出来。T<sub>E</sub>X 或者 L<sup>A</sup>T<sub>E</sub>X 都有可能导致警告。

### §9.5.1 普通 L<sup>A</sup>T<sub>E</sub>X 警告

L<sup>A</sup>T<sub>E</sub>X 的警告是以 ‘LaTeX Warning:’ 为标志的，它位于消息的开头，后接警告消息。

LaTeX Warning: Citation ‘...’ on page ... undefined on  
input line ....

在 `\cite` 命令中的关键词并没有对应的 `\bibitem` 命令进行定义 (4.3.6 和 8.3.2 节)。

LaTeX Warning: Citation ‘...’ undefined on input line ....

<sup>2ε</sup> 在 `\nocite` 命令中的关键词还没有定义。

LaTeX Warning: Command ... has changed.

Check if current package is valid.

<sup>2ε</sup>用 `\CheckCommand` 语句来测试一个命令是否已经具有特定的定义。如果这个测试失败, 就会给出这条警告消息。这可以用来确认给定宏包的设计者到底想做什么。

LaTeX Warning: Float too large for page by ..pt on input line ...

<sup>2ε</sup>浮动对象 ( `figure` 或 `table` 环境 ) 太大, 在一页上放不下来。当然浮动对象总是会被显示出来的, 但是会超出正常页面的边界。

LaTeX Warning: 'h' float specifier changed to 'ht'.

<sup>2ε</sup>浮动对象的定位指定 'h' 表示 '这里' ( 6.6.1 节 ), 并不一定能精确地放在浮动对象所处的位置上。只有在当前页面有足够的空间时, 才会把它放在 '这里', 否则它会出现于下一可用页面的顶部。这个消息就是对这一事实的提醒。

LaTeX Warning: inputting '...' instead of obsolete '...'.

<sup>2ε</sup>有些专门为  $\text{\LaTeX}2.09$  编写的宏包显示要求输入特定的文件, 例如 `article.sty` 已经有一个新的等价文件, 但名称不同 (此时为 `article.cls`)。那么原文件名称就对应一个空文件, 这里给出这个消息, 要求输入正确的文件。

LaTeX Warning: Label '...' multiply defined.

两条 `\label` 或 `\bibitem` 命令定义的标记有相同的名称 (8.3.1 和 8.3.2 节)。即使进行了改正, 这条消息还会再显示一次, 因为这次能从上次运行生成的 `.aux` 文件中找到重名标记。当第三次运行时, 它应该没有了。

LaTeX Warning: Label(s) may have changed.

Return to get cross-references right.

从 `\ref`, `\pageref` 和 `\cite` 命令中得到的输出可能不正确, 因为在处理中它们的值已发生了变化。这就必须再运行一次  $\text{\LaTeX}$ , 以使用正确的值。

LaTeX Warning: Marginpar on page ... moved.

在指定页上的边注已经被向下移动了, 以防边注之间离得太近。这就是说这个边注并不是从 `\marginpar` 命令实际所位于的那行开始显示。

LaTeX Warning: No \author given.

<sup>2ε</sup>调用了 `\maketitle` 命令, 但是前面却没有 `\author` 命令。与没有 `\title` 命令不同的是, 这并不是一个错误, 只是它觉得有点儿奇怪。只所以显示这个消息, 就是以防你忘记输入作者信息。

LaTeX Warning: Optional argument of \twocolumn too tall on page..

<sup>2ε</sup>`\twocolumn` 命令 (3.2.5 节) 开始新页, 并切换进入两列格式。在可省参数值中的文本以单列方式显示在两列文本上面的。如果这部分文本太长, 在一页中放不下来, 就会显示出这条警告。

LaTeX Warning: Oval too small on input line ....

在 `\oval` 命令中指定的尺寸太小， $\text{\LaTeX}$  无法找到一个相应的四分之一圆周。

LaTeX Warning: Reference ‘...’ on page ... undefined on  
input line ....

在 `\ref` 或 `\pageref` 命令中的标志名在前一次运行时还没有用 `\label` 命令进行定义 (8.3.1 节)。如果在下一次运行中还会出现这条消息，那就是少掉了相应的 `\label` 命令。

LaTeX Warning: Text page ... contains only floats.

<sup>2ε</sup> 这条消息指出在指定页上，浮动对象样式参数已经把所有其它正常文本排除在外了。这并不是不好，但通常你需要自己看看这一页面样子。

LaTeX Warning: There were multiply-defined labels.

<sup>2ε</sup> 如果有 `\label` 或 `\bibitem` 命令多次使用同名的标志，就会在  $\text{\LaTeX}$  运行结束时显示这条消息。在运行到每个重复标志附近时，也会显示一条警告消息。

LaTeX Warning: There were undefined references.

<sup>2ε</sup> 如果在前一次运行中，有的 `\pageref` 命令中的标志还没有定义，就会在  $\text{\LaTeX}$  运行结束时显示这条消息。在每次用到一个还没有定义的标志时也会显示一条警告消息。

### §9.5.2 $\text{\LaTeX}$ 宏包警告

类和宏包警告主要就是当要求上载的文件版本和名称与实际不符，或者选项使用不当，或者不知道 `filecontents` 环境已经向一个文件中写入了某些文本时显示的消息。这里所提及的命令在 C.2.9 节有介绍。

类和宏包也可以有自己独特的警告消息。这里不可能列出这些消息，因为它们是与类和宏包密切相关的。

LaTeX Warning: File ‘...’ already exists on the system.

Not generating it from this source.

<sup>2ε</sup> 由于 `filecontents` 环境发现在系统中已存在一个同名的文件，因此它并没有从主文件中提取文本。

LaTeX Warning: Unused global option(s):....

<sup>2ε</sup> 在 `\documentclass` 声明中指定的选项是全局性的，即它要应用于所有的类和后面调用的宏包。然而，如果对某一个选项，所有的类或宏包都不能识别，就会显示出这条警告消息，然后列出没有用的选项清单。

LaTeX Warning: Writing file ‘...’

<sup>2ε</sup> `filecontents` 环境 (C.2.9 节) 正从主文档中提取信息, 并写到一个给定名称的文件中。

LaTeX Warning: You have requested class/package ‘...’,  
but the class/package provides ‘...’.

<sup>2ε</sup> 由内部识别命令 `\ProvidesClass` 或 `\ProvidesPackage` 给出的类或宏包名称并不与 `\usepackage` 或 `\RequirePackage` 命令中的名称一致。

LaTeX Warning: You have requested, on input line ..., version  
‘...’ of class/package ...,  
but only version ‘...’ is available.

<sup>2ε</sup> 由内部识别命令 `\ProvidesClass` 或 `\ProvidesPackage` 给定的类或宏包日期, 早于 `\usepackage` 或 `\RequirePackage` 命令要求的日期。此时类或宏包可能不具备输入文件所要求的全部功能。

LaTeX Warning: You have requested releases ‘...’ of LaTeX,  
but only release ‘...’ is available.

<sup>2ε</sup> 你拥有的 L<sup>A</sup>T<sub>E</sub>X 版本比某些输入文件中 `\NeedsTeXFormat` 命令要求的早 (C.2.1 节), 因此它可能不会提供该文件要求的所有功能。

### §9.5.3 L<sup>A</sup>T<sub>E</sub>X 字体警告

字体警告是在调用 NFSS 命令 (8.5 节) 时出现的。其标志为 LaTeX Font Warning: 后接警告文本。

LaTeX Font Warning: Command ... invalid in math mode.

<sup>2ε</sup> 只能用在文本模式中的命令, 被用在数学模式中。这条命令就会被忽略。`\boldmath`, `\unboldmath` 和 `\em` 都会导致这条消息。也有一些命令会产生同样内容的错误消息。

LaTeX Font Warning: Command `\tracingfonts` not provided.

(Font) Use the ‘tracefnt’ package.

(Font) Command found: on input line ....

<sup>2ε</sup> 字体跟踪诊断工具 `\tracingfonts` 只有当上载了 `tracefnt` 宏包 (217 页) 时才可以使用。否则就会忽略该命令。

LaTeX Font Warning: Encoding ‘...’ has changed to ‘...’ for

(Font) symbol font ‘...’ in the math version ‘...’.

<sup>2ε</sup> 要想在给定的数学变体中应用某一特定的符号字体, 需要暂时修改字体编码。

LaTeX Font Warning: Font shape ‘...’ in size <...> not available

(Font) size <...> substituted.



<sup>2ε</sup>没有为要求的尺寸和形状定义字体，因此用了一个替换尺寸。

LaTeX Font Warning: Font shape ‘...’ undefined

(Font) using ‘...’ instead.

<sup>2ε</sup>要求的形状属性是未知的，或者还没有定义，因此使用一个替换形状。

LaTeX Font Warning: \*\*\* NFSS release 1 command ... found

(Font) \*\*\* Update by using release 2 command ....

(Font) \*\*\* Recovery is probably possible.

<sup>2ε</sup>使用了一条来自于 NFSS 第一版本的命令，而这条命令现在是不可用的。但是 L<sup>A</sup>T<sub>E</sub>X 会尝试插入新的等价命令，并继续处理下去。

#### §9.5.4 T<sub>E</sub>X 警告消息

T<sub>E</sub>X 警告消息的标志为它并不是一个错误消息 (没有前缀 !)，处理过程也不会停下来。最常见的 T<sub>E</sub>X 警告消息是：

Overfull \hbox ....

T<sub>E</sub>X 无法以一种合理的方式断开该行，从而该行有部分内容伸展进右边界。消息中的其它内容可以给你一些帮助。

例如，下面是一条完整的警告消息：

Overfull \hbox (17.2122pt too wide) in paragraph at lines 4--6

[ ]\OT1/cmr/m/n/10 If T[]X can-not find an ap-pro-pri-ate

spot to di-vide a word at the end of the line, as right

here aaaaaaaaaaaaaaaaaaaaaaaaaaaaa

从这条消息我们可以知道当前行长度约超过了 17.2 pt(6 mm)，这也就是进入右页边界的长度。这一行位于从 4-6 行所在的段落中。所用字体的属性标志为 \OT1/cmr/m/n/10。有问题行的文本是 If ... .. right here aaaaaaaaaaaaaaaaaaaaaaaaaaaaa。供选择的断词点用连字符表示，如 di-vide。最后那个单词无法断开，也就是它导致当前的问题。

解决这个问题的一种方法是利用建议断词点，即在单词内部插入 \-，例如 aaaaaaaaa\ -aaaaaaaaaaaaaaaaaaaaa。同样，在有问题单词前面插入 \linebreak 命令，或者把整个段落放在 sloppypar 环境中，也会去掉这条警告消息。

如果一个断词不是很好的行只稍稍长一点儿，比如说是 1 pt 或更少，在绝大多数情形中很少会注意到它，从而我们就不必在意这一点。

Overfull \vbox ....

很少会出现这条消息。它表示 T<sub>E</sub>X 不能很好的分页，文本伸展到了页面底部。T<sub>E</sub>X 通常宁可在一页上少放些内容，也不会放太多的东西。因此只有当一页上有一个太大 (超过 \textheight 的当前值) 的竖直盒子，例如一个长表格时，才会显示出这条警告消息。

Underfull \hbox ....

这是与 `Overfull \hbox` 警告相对的一条消息。当  $\text{\TeX}$  对一行进行了填充, 以使得左右对齐, 但是这时的单词间距太大, 它认为结果不太令人满意时就会显示这条消息。这通常是 `sloppypar` 环境, `\sloppy` 声明或者 `\linebreak` 命令的后果。也有可能是对 `\` 或 `\newline` 的不恰当应用造成的, 例如连续给出了两条 `\` 命令。警告中的其它消息由格式化结果不好的文本以及对单词间距糟糕程度的估计组成的。

如果对于上面的 `Overfull \hbox` 例子, 在 ... 插入命令 `\linebreak`, 即 `as right\linebreak here`, 那么警告消息为

```
Underfull \hbox (badness 5504) in paragraph at lines 4--6
[]\OT1/cmr/m/n/10 If T[]X can-not find an ap-pro-pri-ate
spot to di-vide a word at the end of the line, as right
```

这条消息说明在第 4 行到第 6 行上的段落包含了一个输出行, 其单词间距太宽, 不能令人满意。这行文本的内容为 `If ... ..., as right`, 当前字体为 `\OT1/cmr/m/n/10`。估计值 `badness 5504` 是对间距糟糕程度的度量, 这个值越小, 结果就越好。

为了对这个 `badness` 有一个大致的印象, 那就必须知道它到底是如何计算出来的。所有的单词间距都有一个基本尺寸, 以及它可以伸展或收缩的最佳限度。对于每个输出行, 就会求出达到理想伸展和收缩后的实际单词间距。`badness` 的值为

$$\text{badness} = 100 \times (\text{实际的收缩量} / \text{最佳的收缩量})^3 \text{ 或者}$$

$$\text{badness} = 100 \times (\text{实际的伸展量} / \text{最佳的伸展量})^3$$

这里‘实际的收缩量’和‘实际的伸展量’就是在每行两端加上或减去的距离。

通常  $\text{\TeX}$  容忍把单词间距伸展到 `badness=200`, 即真正的伸展是最佳伸展的 1.26 倍。在 `sloppypar` 环境或者 `\sloppy` 声明后, 文本行是可以无限伸展的。当 `badness` 超过 1000 时就会显示出 `Underfull \hbox` 警告消息, 这意味着伸展值为最佳值的 2.15 倍。如果上述公式结果超过 10000, 就简单地取 `badness` 为 10000。

在实际应用过程中, 可以容忍 `badness < 2000`, 即使这时不理想的单词间距是很容易注意到的。应该打印出结果看看到底怎样。

Underfull \vbox ....

已进行了分页, 并对顶部与底部进行了调整, 但  $\text{\TeX}$  认为段落间距可能不太令人满意。这里的 `badness` 数相应于 `Underfull \hbox` 警告消息中同名数量。

## §9.6 搜索顽固错误

有的时候，你会遇到一个错误，怎么也找不到出错的地方。对于这种可恶的错误，我们推荐如下的搜索策略：

1. 把正在处理的文档复制两份，一份做为旧版本，一份做为工作版本（原来的版本仍保留住，在下面的搜索过程中不要改动它）。
2. 在工作版本文档中，找到出错地方的最外层环境，去掉一个或多个内层环境。如果其没有内层环境，就缩短余下的文本。利用 L<sup>A</sup>T<sub>E</sub>X 再次处理这个文件。
3. 如果仍然有那个错语，就把缩短后的工作版本复制到旧版本中，重复第 2 步。如果第 2 步中外层环境是 `\begin{document} ... \end{document}`，那么就可以通过在某些点简单地插入 `\end{document}` 来缩短文本。
4. 如果在缩短后的工作版本中不再有那个错误，那就把旧版本内容复制到工作版本中，这样错误仍然在这个版本中。这次比上次少去掉一些文本，重复 2 到 4 步。
5. 如果按这种方法，发现错误是在下一个最外层环境中，那就重复从 2 到 4 的同样操作。

利用这种方法，就可以把错误局限在一条命令或者只剩下很少一部分结构的环境中。如果尽管错误已被准确定位，当仍然不知道为什么错了，那就需要得到有经验的团体或计算中心的帮助了。然而，通常只要一找到出错的地方，就很容易知道为什么错了。

有时也确实会即使已改正了错误，在下次运行 L<sup>A</sup>T<sub>E</sub>X 时还会出现这条错误消息。这主要是因为 L<sup>A</sup>T<sub>E</sub>X 通过辅助文件进行信息转递，而这些文件的更新是在当前处理过程结束。例如，若某一个章节命令中有错误，那么即使已修正了这个错误，在 `.toc` 文件中仍旧是那个有错误的条目。如果文档中包含 `\tableofcontents` 命令，那么 L<sup>A</sup>T<sub>E</sub>X 就会在下次运行时读入这个 `.toc` 文件，从而仍然给出那条错语消息。只有当成功处理完一次文件后才会得到新的 `.toc` 文件。

对于这种情况，应当编辑 `.toc` 文件，去掉那个错误。如果没法这样做，那就删掉这个 `.toc` 文件，并用 L<sup>A</sup>T<sub>E</sub>X 编译修正后的文档两遍。如果错误是在 `\caption`, `\addcontentsline` 或者 `\addtocontents` 命令中，也要对 `.lof` 或 `.lot` 文件做同样的处理。

有时我们也必须删掉 `.aux` 文件，以杜绝虽然已改正了 `.tex` 文件中的错误，但仍然显示同样的错误。这时如果导言中有命令 `\nofiles` 时用户必须更加小心，因为此时再运行一次 L<sup>A</sup>T<sub>E</sub>X 也不会生成正确的 `.aux` 文件。