

用于处理多文档项目的文档类和宏包^{*}

Federico Garcia

翻译: [Marlin](#)

2018 年 2 月 12 日

摘要

使用 `subfiles` 宏包，编辑由一个主文档若干个子文档构成的多文档项目时会更方便。用户可以正常编译使用 `\input` 导入子文档的主文档，也可以直接编译子文档本身，此时子文档使用跟主文档相同的导言区而成为完整的一份 \LaTeX 文档。

1 简介

\LaTeX 的 `\include` 和 `\input` 命令可以将多个特定的子文档编辑成一个文档。这有利于编辑包含很多章节的大文档，但有时候在别的场合出于某种需要作者也会想要使用这样的便利。我经常用来处理需要大量反复试错的情况，诸如较长代码的例子，表格，插图等等¹。

在这种情况下，文档的其余部分很少用到，有时候甚至是一种干扰²。很多时候，作者只好选择新建一个文档来编辑，这意味着需要如下三个步骤：

- 新建一个文档，复制粘贴主文档的导言区；
- 在这个例子文档中，在必要的情况下，将它单独编辑编译很多次；
- 得到满意的编译结果后，删掉这个新文档的导言区（包括 `\end{document}`！）然后在主文档里使用 `\include`（很多时候用的是 `\input`）导入子文档。

理想的方式是将上述三个步骤约简为值得关注的一个：中间那一步。这就意味着每一个新建的子文档既是某个项目的一部分，也是一份完整自洽的 \LaTeX 文档，

^{*}本文档适用于 `subfiles` 宏包 (2012/05/23) 版。

¹对我来说，常常需要处理的是 Musix\TeX 音乐文档，代码往往很长，很复杂，而且可读性差。

²比如说，报错信息标识了错误的行号，甚至是错误的文档。

取决于使用 `\include` / `\input` 导入还是由 L^AT_EX 编译。这就是这份 subfiles 文档类和宏包打算做的工作。

这里主要的想法是重定义 `\documentclass` 指令和 `document` 环境。在 L^AT_EX 里这两个概念是很重要的，轻易不应改动。就我所知，subfiles 文档类和宏包对它们作了无害的改动，而且在实现需要的功能后又小心地恢复了所作的改动。这是 subfiles 的第一版，尽管我做了很多次的努力，它还是容易与别的文档类和宏包有冲突影响。

2 用法

2.1 设置

涉及到的文档有如下基本的结构：

<i>MAIN FILE</i>	<i>SUBFILE</i>
<code><some preamble></code>	<code>\documentclass[<main_file_name>]{subfiles}</code>
<code>\usepackage{subfiles}</code>	<code>\begin{document}</code>
<code><more preamble></code>	<code><text, graphics, etc.></code>
<code>\begin{document}</code>	<code>\end{document}</code>
<code><text></code>	
<code>\subfile{<subfile_name>}</code>	
<code><more text></code>	
<code>\end{document}</code>	

subfiles 宏包由 L^AT_EX 项目主文档调用，subfiles 文档类由每个子文档调用。需要注意的是 subfiles 文档类带有一个可选项（实际上是必需的）：主文档名。这个主文档名应该带上 T_EX 文档默认后缀 `.tex`，如果主文档和子文档在不同文件夹下，还需要给出主文档路径（分隔符是 `/` 而不是 `\`），路径中的空格会被忽略（至少在 Windows 下）。

2.2 结果

这样一来，L^AT_EX 编译主文档或子文档时会得到如下结果：

- 如果排版的是子文档自身，复制主文档的导言区（包括 `\documentclass`），然后正常排版。
- 如果子文档是由 `\subfile` 指令导入的，忽略子文档中 `\begin{document}` 指令及其之前的所有内容，也忽略 `\end{document}` 指令。（子文档的正文部分，实际上是被 `\input` 到主文档的。）

`\subfile` 指令更像是 `\input` 而不是 `\include`，并不会开始新的一页。这条指令允许嵌套，但是没有类似于 `\includeonly` 的抑制机制。

2.3 进一步的细节和警告

实际上，排版子文档本身时并不只是复制主文档的导言区，而是复制了主文档在`\begin {document}`和`\end {document}`之外的所有内容。这会有两个后果：a) 排版子文档本身时用户可以只读取处理到主文档导言区的那些命令；b) 用户必须很小心处理主文档中`\end {document}`之后的那些内容，任何语法错误都会毁掉子文档的编译排版。

主文档的导言区可以使用`\input`（注意不是`\include`，也不是`\subfile`）指令导入其他文档（例如包含定义和快捷命令的文档等），子文档的导言区也可以。但是必须意识到每个子文档是作为一个组`\input`导入的，因此在子文档里设置的定义等在子文档之外无效。无论是否使用 `subfiles`，一个好的经验是任何定义都放在主文档的导言区，避免混淆，也易于使用。

原则上，`\subfile`指令可以嵌套使用，在我的测试中也确实可以正常工作，只要每一个子文档在调用 `subfiles` 文档类时使用主文档名作为选项即可。然而我们知道在一些奇怪的情况下会有不可预期的结果。任何情况下，`subfiles` 都不会取代`\include`和`\input`，它们任然有效，可以正常使用。

`subfiles` 文档类和宏包需要调用 `verbatim` 宏包，该宏包的 `comment` 环境用来忽略不同文档的不同部分。这不会是一个问题，因为这个宏包是包含在 \LaTeX 2 ϵ 标准版中的。

3 实现

3.1 文档类

```

1 <*class>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesClass{subfiles}[2012/05/23 Federico Garcia]
4 \RequirePackage{verbatim}
5 \DeclareOption*{\typeout{Preamble taken from file `\"CurrentOption'}}%
6     \let\preamble@file\CurrentOption}
7 \ProcessOptions

```

首先要做的是保存常规的 \LaTeX 中的 `\document`, `\enddocument`, `\documentclass` 等定义：

```

8 \let\old@document@subfiles\document
9 \let\old@enddocument@subfiles\enddocument
10 \let\old@documentclass@subfiles\documentclass

```

现在 document 环境已经重定义为 comment。结果就是主文档的正文部分被 L^AT_EX 忽略，只有导言区被读出来。（还有主文档中 `\end {document}` 指令之后的任何内容！）对子文档已经导入的 `\documentclass` 指令，重定义为 `\LoadClass`。主文档的文档类和选项如是导入子文档。

```
11 \let\document\comment
12 \let\enddocument\endcomment
13 \let\documentclass\LoadClass\relax
```

现在可以由 `\input` 导入主文档，然后恢复原来的 `\document`, `\enddocument`, `\documentclass` 等指令的定义。备份命令被 `\undefined` 释放以节省内存。就好了。

```
14 \input{\preamble@file}
```

这里有些不太明显的东西。通常情况下，`\preamble@file` 包含一些 `\usepackage` 指令，最后使得 `@` 不再是一个字符。鉴于此，接下来需要 `\catcode` 命令，分组和 `\global's`。

```
15 {\catcode`\@=11
16 \global\let\document\old@document@subfiles
17 \global\let\enddocument\old@enddocument@subfiles
18 \global\let\documentclass\old@documentclass@subfiles
19 \global\let\old@document@subfiles\undefined
20 \global\let\old@enddocument@subfiles\undefined
21 \global\let\old@documentclass@subfiles\undefined}
22 </class>
```

3.2 宏包

任何选项都会被忽略。

```
23 <*package>
24 \NeedsTeXFormat{LaTeX2e}
25 \ProvidesPackage{subfiles}[2012/05/23 Federico Garcia]
26 \DeclareOption*{\PackageWarning{\CurrentOption ignored}}
27 \ProcessOptions
28 \RequirePackage{verbatim}
```

宏包的核心。原理是重定义 document 环境，使得子文档的 `\begin` 和 `\end {document}` 容易被导入。`\documentclass` 的重定义是类似的，只是有一个要求和一個对 `\subfile` 指令没有意义的可选参数。

`\skip@preamble`

```
29 \newcommand{\skip@preamble}{%  
30   \let\document\relax\let\enddocument\relax%  
31   \newenvironment{document}{}{}%  
32   \renewcommand{\documentclass}[2][subfiles]{} }
```

注意新的指令 `\subfile` 需要 `skip@preamble` 在一个组。在子文档被导入后，对 `document` 和 `\documentclass` 的改动需要恢复。

```
33 \newcommand\subfile[1]{\begingroup\skip@preamble\input{#1}\endgroup\subfile  
34 </package>
```