

第18章 结 论

现在到了这本入门教程的末尾了。你已经学习了足够的关于用 Emacs Lisp 中编程的知识，如给变量赋值、编写简单的“.emacs”文件，编写简单的功能扩充程序等。

现在是该告一段落的时候了。如果你愿意，你现在可以直接独自朝前走了——自学吧！

你已经学习了编程的一些基本知识。但是这仅仅是一部分而已。还有大量的易于使用的工具我们还没有接触到。

学习更多知识的一个途径，是阅读 GNU Emacs 源代码以及《GNU Emacs Lisp 技术手册》[Ⓐ]中提到的代码。

阅读 Emacs Lisp 源代码是一次冒险。当你阅读这些源代码并遇到一个不熟悉的函数或者表达式时，你需要弄清它的功能是什么。

去看看那本《GNU Emacs Lisp 技术手册》吧。它对 Emacs Lisp 作了完全的、且易于阅读的描述。它不仅是为专家编写的，同时也是为像你这样的读者编写的（这本技术手册随着 GNU Emacs 的标准发行版本发行，并且是一个 Texinfo 源文件，因此你能够在线阅读它，或者打印出来阅读）。[Ⓑ]

GNU Emacs 的另外一个在线帮助系统是所有函数的在线帮助文档以及帮助你阅读这些源代码的命令——find-tags。

下面是我阅读源代码的一个例子。很久以前，我总是根据文件的名称首先查看“simple.el”文件。当我阅读这个文件中的函数定义时，碰巧在这个文件中有一些函数很复杂，或者至少初看起来很复杂。例如，第一个函数看起来就很复杂。这个函数是 open-line 函数。

你可以慢慢地阅读这个函数，就像我们处理 forward-sentence 函数那样（参见12.3节）。或者你希望跳过这个函数阅读下一个函数，比如 split-line。其实，你无需阅读所有这些函数。根据 count-words-in-defun 函数，split-line 函数包含 27 个单词和符号。

虽然这个函数很短，但是它包含四个我们没有学到的函数：skip-chars-forward、indent-to、insert 和“?\n”。

先来考虑 insert 函数（这个函数曾经出现在第12章的回顾部分）。在 Emacs 中，你能够通过键入 C-h f (describe-function) 以及这个函数名，来找到关于这个函数的更多信息——打印出函数文档。你也可以用 find-tag 来阅读函数定义的源代码，这个命令绑定到 M-.(在这个例子中，这个命令并不很有用，因为这是一个用 C 语言编写的基本函数)。最后，你还可以找到技术手册中关于这个函数的说明。你可以在 Info 中键入 i (Info-index) 以及函数名来访问

Ⓐ 《GNU Emacs Lisp 技术手册》中文版由自由软件基金会中国研究院翻译。

Ⓑ 《GNU Emacs 技术手册》中文版由自由软件基金会中国研究院翻译。

技术手册中关于这个函数的内容，或者在一份打印的手册中根据索引来查阅 `insert` 这个函数。

类似地，你可以找到“`?\n`”的含义。你可以尝试用 `Info-index` 来查找相关信息。对于这个函数这样做毫无帮助，但是不要放弃。如果在索引中仅仅查询“`\n`”而不查找“`?`”，你将直接到达手册中的相关章节（参见《GNU Emacs 技术手册》中的“字符类型”一节。“`?\n`”代表换行符）。

你可能已经猜测到 `skip-chars-forward` 和 `indent-to` 函数的功能了，否则你可以仔细看一看。（顺便提一下，`describe-function` 函数本身在“`help.el`”文件中，它是那些较长但是好理解的函数中的一种。它的函数定义说明了如何不用标准的控制字符来定制 `interactive` 表达式，并且显示了如何建立一个临时缓冲区。）

其他有趣的源代码包含在“`paragraphs.el`”、“`loaddefs.el`”以及“`loadup.el`”文件中。其中“`paragraph.el`”文件中包含了短的、易于理解的函数，也有个别长的。“`loaddefs.el`”文件中包含许多标准的自动加载函数以及许多键图。我从来没有全部阅读完这个文件中的所有函数，仅仅阅读了一部分。“`loadup.el`”文件加载 Emacs 的标准部分，它告诉你如何构建 Emacs 的许多知识（详细情况参见《GNU Emacs 技术手册》）。

就像我所说的，你已经学习了一些简单的、但是却是非常重要的知识。我们几乎没有触及编程的主要部分，除了使用预先定义的 `sort` 函数之外，我没有说过任何关于信息分类的事情。我没有提到如何编写用来“编写程序”的程序。这是其他层次较深的书的主题。那是另外一种学习进阶。

阅读完本书，你所做的是学习了足够的使用 Emacs 的实际知识，你已经入门了。这本入门教程到此结束。