安装MIT-Scheme

1. 为什么使用Scheme

通过使用Scheme，你可以：

·编写漂亮的程序。

·享受编程的乐趣。

这就是学习Scheme的原因。在你用Scheme编写一些实用的程序的时候会遇到一些困难。然而，就如卓越的黑客喜爱Scheme，这是一门值得学习的语言。事实上，[计算机程序的构造和解释（Structure and Interpretation of Computer Programs，SICP）](http://mitpress.mit.edu/sicp/)——最好的计算机科学教科书之一——也使用Scheme来描述这些程序。[GNU](http://www.gnu.org/)也使用[Scheme](http://www.gnu.org/software/guile/guile.html)（被称作guile的实现）作为一个通用的脚本语言用于其应用[存疑]（[GNU](http://www.gnu.org/) also uses Scheme (named [guile](http://www.gnu.org/software/guile/guile.html)) for a common script language for its applications.）。guild相当于MS-Word或者Excel（原文是Excell，应该是作者的笔误，译者注）中的宏。他被用来通过简单的脚本来操作应用程序。

尽管Common Lisp更加适合构建实用应用程序，我依然推荐你首先学习Scheme，因为他：

* 1. 语言被设计得紧凑
  2. 语法简单

坊间流传着“[Scheme使你称为更棒的程序员](http://www.catb.org/~esr/faqs/hacker-howto.html" \l "skills1)”的说法。即是你很少在你的商业化实践上使用Scheme，但学习Scheme获得的良好感觉将会在你使用其他程序语言使给予良好的向导。

网络上的Scheme教程（so many as Scheme is popular language）总是或多或少的有些困难故不太适合初学者。本教程，从另一方面来说，是面向新手程序员的，他只需要读者对编程有一点了解即可。

2. 目标读者

本教程的目标读者是仅有一点编程经验的PC用户，譬如：

·因为教授使用Scheme教学而无法跟上进度的学生。

·想要学习编程的人。

Scheme的语法相当的简单，并且可以通过一个简单的方式解释出来。尽管如此，这样解释对初学者来说还是太困难了。在本教程中，我会循序渐进的解释。

Scheme代码仅由单词，括号和空格组成，这些最初可能会使你感到困惑。然而，如果你使用了一个合适的编辑器，他会为你自动展示配对的括号和缩进。因此，你不用担心括号的配对，并且你可以通过缩进来阅读代码。如果缩进看起来很奇怪，你可以用编辑器找出错误的配对。

3. 安装MIT-Scheme

这个教程面向的是Windows用户。我没有使用Macintosh的经验，因此无法提供给你相关的帮助。如果你是Unix（或者Linux）用户，请让管理员安装它（如果你无法自己安装）。使用Scheme并不依赖于操作系统。只有安装才会因不同的系统而不同。

Scheme程序设计语言中有一些规范，最近的规范在[Revised](http://www.schemers.org/Documents/Standards/R5RS/HTML/r5rs.html)[5](http://www.schemers.org/Documents/Standards/R5RS/HTML/r5rs.html)[Report on the Algorithmic Language Scheme (R](http://www.schemers.org/Documents/Standards/R5RS/HTML/r5rs.html)[5](http://www.schemers.org/Documents/Standards/R5RS/HTML/r5rs.html)[RS)](http://www.schemers.org/Documents/Standards/R5RS/HTML/r5rs.html)。

大多数的实现都是（完全或者部分的）基于R5RS的。如果你使用的是部分符合R5RS的实现，在使用时你应小心一点。在Windows系统上有很多免费的Scheme实现，比如：[ChezScheme](http://www.scheme.com/), [MzScheme](http://www.plt-scheme.org/software/mzscheme/), [DrScheme](http://www.drscheme.org/), [SCM](http://swissnet.ai.mit.edu/~jaffer/SCM.html)。在本教程中，我使用[MIT/GNU Scheme](http://www.gnu.org/software/mit-scheme/)因为他安装非常容易并且高效。MIT-Scheme的解释器十分快速，除此之外他还能够将你的程序编译为本地代码。MIT-Scheme的问题就是并不完全符合R5RS规范。稍后我会详细的讨论这点。事实上，只有MIT-Scheme和DrScheme有安装包。有人推荐DrScheme，但是它太慢了。如果你掌握有足够的知识，我推荐你 安装[Petite Chez Scheme](http://www.scheme.com/petitechezscheme.html)。这是一个运行在命令提示符（DOS Windows）下的非常棒的解释器。

[Scheme Implementations](http://www.aracnet.com/~briand/scheme_eval.html)比较了几个Scheme实现。当你习惯Scheme后，去尝试几种不同的实现将会是很好的主意。你需要一台Linux机器，因为大多数Scheme实现都是只能运行在Unix和Linux上的。

3.1 如何在Windows上安装MIT-Scheme

MIT-Scheme可以简单的通过下载并执行安装包来进行安装。

1. 访问[MIT/GNU Scheme的主页](http://www.gnu.org/software/mit-scheme/)，下载适用于Windows的二进制包： mit-scheme-N.N.N-ix86-win32.exe。

2. 双击下载好的安装包。安装包会询问一些事只需要按照默认的设置进行即可。

3. 安装完毕后，有4个快捷方式被创建出来，分别是：Scheme，Compiler，Edwin和Documentation。Scheme，Compiler和Edwin都是指向同一个程序的快捷方式，但他们调用程序的参数不同。使用Compiler，你可以把你的程序编译为本地代码，这样可使得你的程序运行时间更短。另一方面，Compiler会消耗更多的内存。

Edwin是一个Emacs-like的编辑器，用于编辑Scheme程序。你可以使用这个编辑器或者你最喜欢的编辑器。

4. 你可以通过编辑配置文件scheme.ini来自定义MIT-Scheme。

scheme.ini文件的创建路径是由环境变量HOMEPATH决定的。你可以通过在控制台（DOS Windows）中键入

1. >set HOMEPATH

来取得HOMEPATH H的值。在WinXP中，HOMEPATH被预定义为：

\Document and Setting\*username*

下面的代码给出了一个scheme.ini文件的例子：

(cd "C:\\doc\\scheme")

(define call/cc call-with-current-continuation)

第一行代表将工作目录切换到C:\doc\scheme。通过这条代码，MIT-Scheme移动工作路径切换到这个路径，你不需要再敲击程序的绝对路径来载入Scheme程序。第二行是定义call-with-current-continuation的缩略词。

4. 总结

因为安装非常容易（除了编辑scheme.ini文件），你应该会毫无问题。

下一章节是如何与MIT-Scheme前端会话。