<<实用Common Lisp编程>>

图书基本信息

书名：<<实用Common Lisp编程>>

13位ISBN编号：9787115263742

10位ISBN编号：7115263744

出版时间：2011-10

出版时间：人民邮电

作者：Peter Seibel

译者：田春

前言

译者序　　很荣幸，我被授权翻译Practical Common Lisp一书。
本书是自1994年Common Lisp语言标准化以来，国内出版的第一本Common Lisp的中文教材。
Lisp语言家族最早诞生于1959年，它是人类历史上第二个高级程序设计语言（第一个是Fortran）。那一年，人工智能（AI）专家John McCarthy发表了具有重大历史意义的第一篇LISP论文“Recursive Functions of Symbolic Expressions and their Computation by Machine, Part I”，其中介绍了一种运行在古老的IBM 704计算机上的列表处理语言LISP（LISt Processing，列表处理），借助它可以轻松描述当时人工智能领域用到的各种算法。
从此，Lisp语言在包括AI领域在内的所有主流计算机分支上，都获得了长足的发展。
Lisp平台不但在IBM PC出现之前的几乎所有计算机硬件体系上均有移植，甚至在20世纪80年代还出现过专门用来运行Lisp程序的硬件——Lisp机。
1994年，ANSI标准化的Common Lisp语言将之前历史上的所有现存Lisp厂商的各种语言和平台特性做了一次伟大的总结，从此语言核心不再变化，不但标准化以前的历史遗留代码只通过少量修改就可以兼容现代Lisp平台，而且标准化以后写出的所有新代码也都几乎不经任何调整就可能运行在任何一种Common Lisp平台上，无论是带有原生或是字节码编译器的，还是间接转译成C语言的，或是运行在JVM上的。
目前至少有13种不同的Common Lisp语言平台可以运行在现代计算机上，其中10种还在广泛使用中，远超过它们所在的操作系统上C和其他语言编译器的数量。
可以说，Lisp语言家族长达50年的发展史就是整个计算机发展史的缩影。
我从2003年大学三年级时开始学习Common Lisp语言，至今已有八个年头。
当时学习它的动机基本上是出于对人工智能（传统的逻辑和推理、知识表示等方向）的个人兴趣。
不过随后很快就发现，Common Lisp是一门通用的编程语言，如果不考虑其历史渊源而只从语言本身的特性来观察的话，可以说它跟人工智能毫无关系。
在Practical Common Lisp一书中，作者Peter Seibel也谈到这个问题。
当今有太多的人对Lisp语言存在类似的误解，包括相当多学过早期Lisp语言的人还停留在列表（List）是Lisp语言的唯一复合数据类型的认识上。
如果读者从头到尾学完了这本书，就会发现Common Lisp是一门特性丰富的大型编程语言，不但提供了现代编程语言普遍支持的各种数据类型（包括各种数值类型、字符串、数组、结构体和哈希表在内），还支持几乎所有的编程范式（面向过程的、函数式的以及面向对象的），尤其带有一套特性丰富且思想独到的面向对象编程接口CLOS（Common Lisp Object System）和OO扩展接口MOP（Meta-Object Protocol）。
如果要用一句话来描述Common Lisp中的OO与C++/Java/SmallTalk等语言的OO有何不同，那就是Common Lisp对象系统完全不是基于消息传递的，而是基于广义函数的。
有兴趣的读者应当仔细阅读本书的第16章和第17章，其中介绍了CLOS的一些入门内容。
不过Lisp语言最吸引人的地方还在于其与众不同的程序运行方式。
从C语言一路学过来的人往往把一门语言的语法及其标准函数库视为语言的全部，因为一旦程序写好，编译器就会将整个代码编译成一个可执行程序或者被其他可执行程序使用的库。
接下来语言本身是什么就不重要了，重要的是程序员写出了什么功能，甚至连编译器本身是什么都不重要，因为它只是一个黑箱，除了简单的优化开关之外几乎无法调整其行为。
各种Lisp语言则采用完全不同的方式来运行Lisp程序：Lisp平台本身是一个交互式的环境，它在很大程度上就是用其本身写成的。
用户的Lisp代码以编译或解释的形式加载到Lisp环境中，然后跟Lisp语言或平台本身的代码直接融合在一起。
换句话说，每一个Lisp程序都是对Lisp语言本身的某种形式的扩展。
然后通过一个启动函数，整个程序得以运行。
听到这里，读者似乎看到了Python或者Ruby的影子，但Lisp环境还有更绝的地方：几乎所有Lisp平台都

允许用户将加载了用户代码的整个环境从内存中导出（dump）为一个磁盘文件。
通过直接加载这个文件而不是默认的那个只含有Lisp本身的文件，可以迅速地重建导出前的Lisp环境，从而达到增量开发或者哪怕是快速加载已有Lisp程序的目的。
最后，和其他语言很不同的一点是，Lisp语言规范（至少Common Lisp是这样的）不但包括了如何定义某个程序组成部分（指的是变量、函数和类这些东西）的能力，还定义了从Lisp环境中清除任何程序组成部分以及就地修改它们的能力，并在语义和功能上确保了这些操作不会破坏运行中的Lisp代码。
这导致了Lisp语言的另一个重要应用：通过加载补丁，Lisp系统可以在运行中被任意修改，这对24×7的服务器端程序的平滑升级尤为有利。
顺便说一句，Lisp也是最早引入垃圾收集（GC）机制的编程语言，Lisp环境中的任何对象，一旦失去了来自其他对象的引用，就会在某个时刻被GC系统从内存中清除掉。

读者可能已经注意到了我在不停地混用Lisp和Common Lisp两个概念。
这有两层含义：首先，存在Common Lisp之外的Lisp语言，更准确地说是Lisp方言（dialect），至少包括了Emacs Lisp、AutoLISP、Scheme、Racket（前身是PLT Scheme）和Clojure，其中最后一个是高速发展中的新兴Lisp方言；其次，所有Lisp家族的语言都有很多共性，除了上面描述中带有Lisp而非Common Lisp字样的部分以外，还有最大的也是初学者最容易看到的一点，那就是所有Lisp方言都使用前缀表达式和用小括号表示的列表，例如 1+1 在Lisp中将写成（+ 1 1）。
很多初学者一开始都不适应前缀表达式，但我认为前缀表达式是有很多优点的：首先，它彻底消除了运算符结合性问题，令表达式毫无歧义可言；其次，它让语言处理器更加简单高效，避免了语法分析的困难。
当然，一旦习惯了也就感觉没什么了。

学习本书对更好地使用其他Lisp方言无疑是大有帮助的。
在翻阅书店里关于AutoLISP（AutoCAD计算机辅助设计软件的扩展语言）的各种书籍时，我经常痛心疾首地发现这些图书的作者虽然精通AutoCAD所提供的Lisp编程接口，但写出的AutoLISP代码要么极为难看，要么缺乏效率、滥用内存。
AutoLISP在语法上跟Common Lisp非常接近，本书的大部分内容都适用于AutoLISP。
因此我强烈推荐所有AutoLISP程序员阅读本书以加强自身的Lisp素养。
同样的问题对于Emacs Lisp（GNU Emacs文本编辑器的扩展语言）来说也是一样的。
Scheme系的Lisp方言区别相对大一些，如果连基本的变量和函数定义都在形式上完全不同的话（当然，思想上是没什么本质区别的），我恐怕初学者从本书中学得Scheme编程思想的机会不大，这种情况下还是推荐《计算机程序的构造和解释》、Lisp in Small Pieces和The Little Schemer等书籍比较好。

本书可以作为其他Common Lisp语言教材的学习基础。
在本书的最后一章里，作者给出了很多后续的教材，在此就不一一重复了。
需要特别指出的是，另一本著名的Common Lisp教材On Lisp（作者Paul Graham，也就是《黑客与画家》一书的作者）多年前已经被我和我的几位朋友共同翻译成中文版，细心的读者可以从网上轻易地找到它。
On Lisp主要介绍Common Lisp的宏编程，这是Common Lisp区别于其他语言甚至其他Lisp方言的最重要特性。
我相信一旦读者掌握了本书中关于宏的章节以后就可以阅读On Lisp中的进阶内容，从而将自身对编程语言的认识上升到一个新的高度，不过更加符合实用原则的思路还是先把本书读完。

Common Lisp绝不是一门过时的编程语言，整个Common Lisp社区一直都在高速的发展之中，近几年的发展尤为迅速。
在我学习Common Lisp的这些年里，我亲眼目睹了几个Common Lisp平台从无到有（ECL、ABCL）或者发展壮大（SBCL、Clozure CL）的过程。
经典平台（CMUCL、MCL）也得到了良好的维护并始终跟进操作系统的自然发展。
随着计算机硬件的高速发展，即便相对保守的Common Lisp商业平台也开始或即将开始支持对称多处理器（SMP），其中LispWorks和Scieneer CL都以SMP支持作为主要卖点。
第三方软件包长足发展，虽然尚未达到Perl社区CPAN的水平，但常用的工具包一应俱全，其中不乏高

质量的大型项目。
近年来最新的成果Quicklisp包管理平台，更是将Common Lisp第三方软件包的安装过程提升到了前所未有的便捷程度。
免费平台越来越好，商业平台依然昂贵，开源工具蓬勃发展，所有这些都暗示着Common Lisp语言还保持着旺盛的生命力，唯一的问题是如何让更多的国内计算机领域爱好者了解它。
这就是我翻译本书的目的所在。
过去8年里，我一直活跃在国内和国际Common Lisp社区的前沿。
我在大学本科的最后两年学完了Common Lisp语言语法的主要部分，读完了包括本书在内的几本最经典的Lisp书籍，并已经能够在当时最常见的CMUCL平台（CMU Common Lisp）上编写一些简单的程序。
后来在网易工作的5年里，我在工作之余从头研究了一遍Lisp语言的发展史，亲身体会了包括Lisp机在内的十几种不同的Common Lisp平台或实现，并自费购买了价值数千美元的商业开发环境LispWorks，拥用三种主流操作系统上的License。
在网易从事Linux系统管理工作期间，我用Common Lisp从头实现了一万行源代码规模的SNMP简单网络管理协议工具包，它可以为任何服务器端Common Lisp程序添加通过SNMP协议进行远程监控的能力，也可以作为基于Common Lisp的网络监控系统的基础。
我还在过去3年里参与维护了Common Lisp社区两个最重要的可移植网络库之一：usocket，并由于SNMP库的需要将其从原本只支持TCP扩展到了同时支持UDP，其中对于LispWorks的UDP支持代码是完全从头写的，因为官方并不支持。
2009年，我向国际Lisp会议的投稿被接受，并作为会议论文集的一部分出版。
我是长期担任水木社区函数型编程板块的板主之一，专门负责Lisp方向的讨论和技术分享。
2011年7月，我离开网易以后开始全职从事商业Lisp软件相关的开发工作。
可能我还不是一个很好的译者，但作为一个经验丰富的Common Lisp程序员，我相信自己翻译这本书是合适的。
计算机领域每天都在高速发展，新语言和新技术的产生速度早已超过了一般人的学习速度。
对于一个计算机领域的从业人员或爱好者来说，学习通常是为了更好地应用，把所有时间都用来学习而无暇具体应用也是本末倒置。
在这种情况下，有选择地学习最有用、最不易变质的知识，以及甄别各种计算机知识的重要程度和相互关系的能力就显得非常重要了。
从计算机语言的发展历史来说，如果一门语言可以存活50年，那么它的内在生命力很可能保证其继续长期存活下去，一个人用这门语言写下的代码也将比其他语言的代码更有可能长久地造福后人。
总之，希望这本书能将读者顺利带入Lisp领域。
学习一门新的语言总是要花些成本的，但我想说，和其他任何语言相比，花在理解Lisp上的时间和精力将绝对是物超所值的，即便相当多的读者可能没有机会在短期内将Lisp用于他们的日常工作。
之所以这样说是有原因的：C和Lisp是编程语言的两个极端，大多数人已经熟悉了C的那一端，但如果他们还熟悉另一端的话，那么迅速理解几乎所有其他的编程语言将不再是问题。
译者简介　　田春，网名“冰河”，Glority Software资深软件工程师，前网易杭州研究院高级开发工程师和系统管理员，资深Common Lisp程序员；2001～2005年就读于浙江大学，2003年起开始学习Commom Lisp，精通Lisp史和各种实现，2007年起成为 LispWorks 付费用户；Common Lisp社区的网络专家，开源项目cl-net-snmp（SNMP协议库）的作者，usocket跨平台网络库的主要维护者，common-lisp.net站点管理员，水木社区（newsmth.net）函数型编程语言（FuncProgram）版主，美国Versata/Gensym公司技术顾问；曾在ILC 2009（国际Lisp会议）上发表学术论文，在《程序员》杂志上发表Common Lisp专题文章；曾在2008年翻译了Paul Graham的On Lisp一书，并在网上撰写过大量相关的技术文章。

<<实用Common Lisp编程>>

内容概要

《实用Common Lisp编程》是一本不同寻常的Common Lisp入门书。
《实用Common Lisp编程》首先从作者的学习经过及语言历史出发，随后用21个章节讲述了各种基础知识，主要包括：REPL及Common Lisp的各种实现、S-表达式、函数与变量、标准宏与自定义宏、数字与字符以及字符串、集合与向量、列表处理、文件与文件I/O处理、类、FORMAT格式、符号与包，等等。
而接下来的9个章节则翔实地介绍了几个有代表性的实例，其中包含如何构建垃圾过滤器、解析二进制文件、构建ID3解析器，以及如何编写一个完整的MP3 Web应用程序等内容。
最后还对一些未介绍内容加以延伸。

《实用Common Lisp编程》内容适合Common Lisp初学者及对之感兴趣的相关人士。

<<实用Common Lisp编程>>

作者简介

Peter Seibel 从作家演变成程序员，又从程序员演变成作家，其职业生涯可谓一波三折。
他在获得英语专业学士学位后做过一段时间的记者工作，后来被Web所吸引。
在20世 纪90年代早期，他用Perl建立了Mother Jones杂志和Organic Online网站。
他作为WebLogic的早期雇员参与了Java革命，随后又在加州大学伯克利分校教授Java编程。
他也是第二代Lisp程序员之 一，并曾经是Symbolics的早期股东。
2003年他辞去技术工作，潜心研究Lisp，并凭借本书获得Jolt生产效率大奖。
2009年出版了名噪一 时的访谈录《编程人生》（Coders at Work）。

<<实用Common Lisp编程>>

书籍目录

<<实用Common Lisp编程>>

编辑推荐

由塞贝尔编著的《实用Common Lisp编程》展示了Lisp的威力，不仅表现在传统领域上，例如仅使用短短26行代码就开发出一个完整的单元测试框架，而且还表现在一些全新的领域上，诸如解析二进制MP3文件、构建浏览歌曲集的Web应用、在Web上传播音频流等。
读过本书，你将体会到Lisp具有Pytl-ion等脚本语言的简洁性、C++的高效性，以及在设计语言扩展时无与伦比的灵活性。
《实用Common Lisp编程》内容适合Common Lisp初学者及对之感兴趣的相关人士。

<<实用Common Lisp编程>>

版权说明