

《自制编程语言》

图书基本信息

书名：《自制编程语言》

13位ISBN编号：9787115333209

10位ISBN编号：7115333203

出版时间：2013-11

出版社：人民邮电出版社

作者：[日] 前桥和弥

页数：396

译者：刘卓,徐谦,吴雅明

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《自制编程语言》

内容概要

只需编程基础

从零开始自制编程语言

支持面向对象、异常处理等高级机制

本书手把手地教读者用C语言制作两种编程语言：crowbar与Diksam。crowbar是运行分析树的无类型语言，Diksam是运行字节码的静态类型语言。这两种语言都具备四则运算、变量、条件分支、循环、函数定义、垃圾回收等功能，最终版则可以支持面向对象、异常处理等高级机制。所有源代码都提供下载，读者可以一边对照书中的说明一边调试源代码。这个过程对理解程序的运行机制十分有帮助。本书适合有一定基础的程序员和编程语言爱好者阅读。

作者简介

作者简介：

前桥和弥（Maebashi Kazuya）

1969年出生，著有《征服C指针》、《彻底掌握C语言》、《Java之谜和陷阱》等。其一针见血的“毒舌”文风和对编程语言深刻的见地受到广大读者的欢迎。

作者主页：<http://kmaebashi.com/>。

译者简介：

刘卓

2004年开始从事对日软件开发工作，其间还从事技术及软件工程相关培训工作。自2011年开始从事电力行业产品研发。持续关注企业级应用架构和Web客户端技术。

徐谦

6年技术开发及项目经验，曾以技术工程师身份赴日本工作两年，后归国联合创办互联网公司，现居上海继续创业中。主要从事PHP方向的Web开发。热爱开源，曾向Zend Framework等知名PHP开源项目贡献代码，并于Github自主研发运维EvaThumber等开源项目获得国内社区认可。乐于分享技术心得，个人技术博客avnpc.com在国内PHP圈小有影响。

吴雅明

13年编程经验。其中7年专注于研发基于Java EE和.NET的开发框架以及基于UML 2.0模型的代码生成工具。目前正带领团队开发云计算PaaS平台及云计算自动化配置部署的系统。译著有《征服C指针》等。

书籍目录

目 录

第1章 引子	001
1.1 为什么要制作编程语言	002
1.2 自制编程语言并不是很难	003
1.3 本书的构成与面向读者	004
1.4 用什么语言来制作	006
1.5 要制作怎样的语言	007
1.5.1 要设计怎样的语法	007
1.5.2 要设计怎样的运行方式	009
补充知识 “用户”指的是谁?	012
补充知识 解释器并不会进行翻译	012
1.6 环境搭建	012
1.6.1 搭建开发环境	012
补充知识 关于bison与flex的安装	014
1.6.2 本书涉及的源代码以及编译器	015
第2章 试做一个计算器	017
2.1 yacc/lex是什么	018
补充知识 词法分析器与解析器是各自独立的	019
2.2 试做一个计算器	020
2.2.1 lex	021
2.2.2 简单正则表达式讲座	024
2.2.3 yacc	026
2.2.4 生成执行文件	033
2.2.5 理解冲突所代表的含义	034
2.2.6 错误处理	040
2.3 不借助工具编写计算器	041
2.3.1 自制词法分析器	041
补充知识 保留字(关键字)	046
补充知识 避免重复包含	047
2.3.2 自制语法分析器	048
补充知识 预读记号的处理	053
2.4 少许理论知识——LL(1)与LALR(1)	054
补充知识 Pascal/C 中的语法处理诀窍	056
2.5 习题:扩展计算器	056
2.5.1 让计算器支持括号	056
2.5.2 让计算器支持负数	058
第3章 制作无类型语言crowbar	061
3.1 制作crowbar ver.0.1语言的基础部分	062
3.1.1 crowbar是什么	062
3.1.2 程序的结构	063
3.1.3 数据类型	064
3.1.4 变量	064
补充知识 初次赋值兼做变量声明的理由	066
补充说明 各种语言的全局变量处理	067
3.1.5 语句与结构控制	067
补充知识 elif、elsif、elseif的选择	068
3.1.6 语句与运算符	069

3.1.7 内置函数	069
3.1.8 让crowbar支持C语言调用	070
3.1.9 从crowbar中调用C语言（内置函数的编写）	071
3.2 预先准备	071
3.2.1 模块与命名规则	072
3.2.2 内存管理模块MEM	073
补充知识 valgrind	075
补充知识 富翁式编程	075
补充知识 符号表与扣留操作	076
3.2.3 调试模块DBG	076
3.3 crowbar ver.0.1的实现	077
3.3.1 crowbar的解释器——CRB_Interpreter	077
补充知识 不完全类型	080
3.3.2 词法分析——crowbar.l	081
补充知识 静态变量的许可范围	084
3.3.3 分析树的构建——crowbar.y 与create.c	085
3.3.4 常量折叠	089
3.3.5 错误信息	089
补充知识 关于crowbar中使用的枚举型定义	091
3.3.6 运行——execute.c	092
3.3.7 表达式评估——eval.c	096
3.3.8 值——CRB_Value	104
3.3.9 原生指针型	105
3.3.10 变量	106
3.3.11 字符串与垃圾回收机制——string_pool.c	108
3.3.12 编译与运行	110
第4章 数组和mark-sweep垃圾回收器	113
4.1 crowbar ver.0.2	114
4.1.1 crowbar的数组	114
4.1.2 访问数组元素	115
4.1.3 数组是一种引用类型	116
补充知识 “数组的数组” 和 multidimensional arrays	116
4.1.4 为数组添加元素	118
4.1.5 增加(模拟)函数调用功能	118
4.1.6 其他细节	118
4.2 制作mark-sweep GC	119
4.2.1 引用数据类型的结构	119
4.2.2 mark-sweep GC	121
补充知识 引用和immutable	123
4.2.3 crowbar栈	124
4.2.4 其他根	127
4.2.5 原生函数的形式参数	128
4.3 实现GC本身	129
4.3.1 对象的管理方法	129
4.3.2 GC何时启动	129
4.3.3 sweep阶段	132
补充知识 GC现存的问题	133
补充知识 Coping GC	134
4.4 其他修改	136

4.4.1 修改语法	136
4.4.2 函数的模拟	137
4.4.3 左值的处理	139
4.4.4 创建数组和原生函数的书写方法	142
4.4.5 原生指针类型的修改	144
第5章 中文支持和Unicode	147
5.1 中文支持策略和基础知识	148
5.1.1 现存问题	148
5.1.2 宽字符（双字节）串和多字节字符串	149
补充知识 wchar_t 肯定能表示1个字符吗？	150
5.1.3 多字节字符/宽字符之间的转换函数群	150
5.2 Unicode	153
5.2.1 Unicode的历史	153
5.2.2 Unicode的编码方式	154
补充知识 Unicode可以固定（字节）长度吗？	156
5.3 crowbar book_ver.0.3的实现	156
5.3.1 要实现到什么程度？	156
5.3.2 发起转换的时机	157
5.3.3 关于区域设置	158
5.3.4 解决0x5C问题	158
补充知识 失败的 #ifdef	160
5.3.5 应该是什么样子	160
补充知识 还可以是别的样子——Code Set Independent	161
第6章 制作静态类型的语言Diksam	163
6.1 制作Diksam Ver 0.1语言的基本部分	164
6.1.1 Diksam的运行状态	164
6.1.2 什么是Diksam	165
6.1.3 程序结构	165
6.1.4 数据类型	166
6.1.5 变量	166
6.1.6 语句和流程控制	167
6.1.7 表达式	167
6.1.8 内建函数	168
6.1.9 其他	168
6.2 什么是静态的/执行字节码的语言	169
6.2.1 静态类型的语言	169
6.2.2 什么是字节码	169
6.2.3 将表达式转换为字节码	170
6.2.4 将控制结构转换为字节码	173
6.2.5 函数的实现	173
6.3 Diksam ver.0.1的实现——编译篇	175
6.3.1 目录结构	175
6.3.2 编译的概要	176
6.3.3 构建分析树（create.c）	176
6.3.4 修正分析树（fix_tree.c）	179
6.3.5 Diksam的运行形式——DVM_Executable	185
6.3.6 常量池	186
补充知识 YARV 的情况	187
6.3.7 全局变量	188

6.3.8 函数	189
6.3.9 顶层结构的字节码	189
6.3.10 行号对应表	190
6.3.11 栈的需要量	190
6.3.12 生成字节码 (generate.c)	191
6.3.13 生成实际的编码	193
6.4 Diksam虚拟机	197
6.4.1 加载/链接DVM_Executable到DVM	200
6.4.2 执行——巨大的switch case	202
6.4.3 函数调用	204
第7章 为Diksam引入数组	207
7.1 Diksam中数组的设计	208
7.1.1 声明数组类型的变量	208
7.1.2 数组常量	209
补充知识 D 语言的数组	210
7.2 修改编译器	210
7.2.1 数组的语法规则	210
7.2.2 TypeSpecifier结构体	212
7.3 修改DVM	213
7.3.1 增加指令	213
补充知识 创建Java 的数组常量	215
补充知识 C 语言中数组的初始化	217
7.3.2 对象	217
补充知识 ArrayStoreException	218
7.3.3 增加null	219
7.3.4 哎! 还缺点什么吧?	219
第8章 将类引入Diksam	221
8.1 分割源文件	222
8.1.1 包和分割源代码	222
补充知识 #include、文件名、行号	225
8.1.2 DVM_ExecutableList	225
8.1.3 ExecutableEntry	226
8.1.4 分开编译源代码	227
8.1.5 加载和再链接	230
补充知识 动态加载时的编译器	233
8.2 设计Diksam中的类	233
8.2.1 超简单的面向对象入门	233
8.2.2 类的定义和实例创建	237
8.2.3 继承	239
8.2.4 关于接口	241
8.2.5 编译与接口	242
8.2.6 Diksam怎么会设计成这样?	243
8.2.7 数组和字符串的方法	245
8.2.8 检查类的类型	246
8.2.9 向下转型	246
8.3 关于类的实现——继承和多态	247
8.3.1 字段的内存布局	247
8.3.2 多态——以单继承为前提	249
8.3.3 多继承——C++	250

8.3.4 Diksam的多继承	252
补充知识 无类型语言中的继承	254
8.3.5 重写的条件	254
8.4 关于类的实现	256
8.4.1 语法规则	256
8.4.2 编译时的数据结构	258
8.4.3 DVM_Executable中的数据结构	260
8.4.4 与类有关的指令	262
补充知识 方法调用、括号和方法指针	263
8.4.5 方法调用	264
8.4.6 super	266
8.4.7 类的链接	266
8.4.8 实现数组和字符串的方法	267
8.4.9 类型检查和向下转型	267
补充知识 对象终结器 (finalizer) 和析构函数 (destructor)	268
第9章 应用篇	271
9.1 为crowbar引入对象和闭包	272
9.1.1 crowbar的对象	272
9.1.2 对象实现	273
9.1.3 闭包	274
9.1.4 方法	276
9.1.5 闭包的实现	278
9.1.6 试着跟踪程序实际执行时的轨迹	281
9.1.7 闭包的语法规则	284
9.1.8 普通函数	284
9.1.9 模拟方法 (修改版)	285
9.1.10 基于原型的面向对象	286
9.2 异常处理机制	286
9.2.1 为crowbar引入异常	286
9.2.2 setjmp()/longjmp()	289
补充知识 Java 和C# 异常处理的不同	293
9.2.3 为Diksam引入异常	295
补充知识 catch 的编写方法	296
9.2.4 异常的数据结构	297
9.2.5 异常处理时生成的字节码	299
9.2.6 受查异常	301
补充知识 受查异常的是与非	303
补充知识 异常处理本身的是与非	304
9.3 构建脚本	305
9.3.1 基本思路	306
9.3.2 YY_INPUT	307
9.3.3 Diksam的构建脚本	308
9.3.4 三次加载/ 链接	308
9.4 为crowbar引入鬼车	309
9.4.1 关于“鬼车”	309
9.4.2 正则表达式常量	310
9.4.3 正则表达式的相关函数	311
9.5 其他	312
9.5.1 foreach 和迭代器 (crowbar)	312

9.5.2 switch case (Diksam)	314
9.5.3 enum (Diksam)	315
9.5.4 delegate (Diksam)	316
9.5.5 final、const (Diksam)	319
附录A crowbar语言的设计	322
附录B Diksam语言的设计	336
附录C Diksam Virtual Machine 指令集	359
编程语言实用化指南——写在最后	369
参考文献	375

精彩短评

- 1、:无
- 2、前边还不错，越往后越抽象，说语法的篇幅大于实现，整体还是值得阅读
- 3、对语言背后的思考有了更多的了解，打算配上代码在读一遍。翻译偶尔有些生硬。
- 4、有趣的技术书。书里还有翻页小动画。
- 5、感谢@soasme 借书，这方面的书看过不少，但此本是少见的从无到有去玩的书，看Lua源码这种大家都看懂，但背后怎么一步一步架构起来的却更有意义一些。
- 6、已暂停，只读了一章，觉得受益匪浅。编译器原理，待我再积累一段时间，重新再战。
- 7、入门学习，希望能开发自己的语言
- 8、毕设是编译器，这本书帮了很大忙
- 9、从实践角度入手，让人加深了对语言的理解，好书！
- 10、显然这本书虽然初级但入门来说非常不错
- 11、玩具式语言，代码质量一般。如果是初学者会比较合适..
- 12、还凑合的入门书
- 13、有些部分真的翻译得莫名其妙。好歹把一些专业名词翻对吧==
- 14、还不错，科普一下，龙书真心啃不动
- 15、主要介绍如何自己制作编程语言。文中介绍两种语言，一种是动态无类型，一种是静态的。觉得本书作者在陈述自己设计方面欠缺功底，看后不容易理解。而且设计的两个语言太大，不适合初学者学习。
- 16、再看看
- 17、一般，主要是熟悉一下里面介绍的几个工具，对于编译器思想介绍的不深。可以先看看这本书，找找感官上的认识，然后还是得看龙书
- 18、好多东西都讲不太清楚，需要百度的次数比较多
- 19、更加深入的理解C
- 20、选题和写法很有吸引力（让人看不懂放下，然后又拿起看），但是缺少一章解析全书思路的概论，最后只好虎头蛇尾般读完.....
- 21、挺好的，让人对一些程序概念的实现有了一些了解
- 22、gc unicode closure 等等 可以一阅
- 23、--不太喜欢这书，yacc什么的感觉用着好恶心

《自制编程语言》

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com