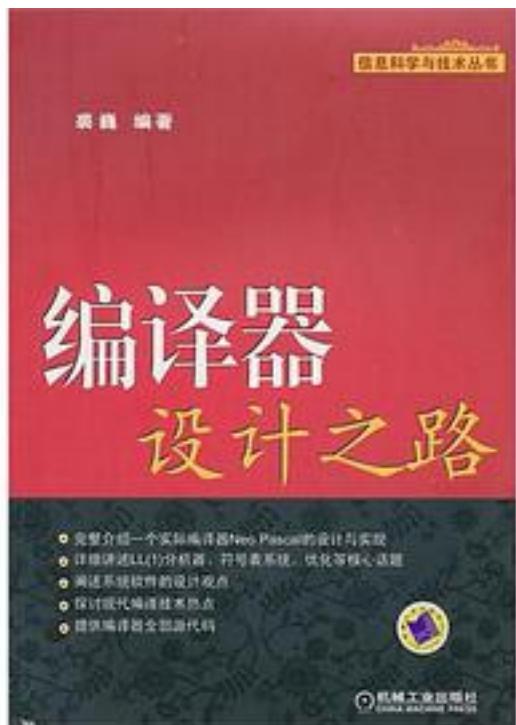


编译器设计之路



[编译器设计之路 下载链接1](#)

著者:裘巍

出版者:机械工业出版社

出版时间:2011-1

装帧:平装

isbn:9787111321644

《编译器设计之路》系统地介绍了一个实际的Pascal编译器Neo Pascal的设计与实现。结合Neo Pascal的源代码，详细讲述了LL (1) 语法分析器、符号表系统、中间表示、类型系统、优化技术、运行时刻的存储管理、代码生成器等编译器设计的核心话题。各章都附有少量以实践应用为主的练习题，既可作为阅读思考题，也可作为课程设计选题。

与国内其他介绍编译技术的图书相比，《编译器设计之路》更关注的是编译器的实现细节，而不仅仅局限于理论阐述。《编译器设计之路》可供从事编译器设计相关工作的工程人员阅读，也可作为高等院校计算机专业的编译原理课程参考书。

作者介绍:

目录: 前言

第1章 概述 1

1.1 编译技术概述 1

1.1.1 程序设计语言基础 1

1.1.2 程序设计语言的翻译机制 4

1.1.3 编译器的基本结构 5

1.2 Pascal语言基础 8

1.2.1 Pascal语言简介 8

1.2.2 Pascal程序基本组成 9

1.2.3 Pascal的声明部分 10

1.2.4 Pascal的类型 12

1.2.5 Pascal的运算符 15

1.2.6 Pascal的语句 17

1.3 开发环境与Delphi基础 18

1.3.1 开发环境与文件列表 18

1.3.2 Delphi基础 19

1.4 深入学习 24

1.5 实践与思考 25

1.6 大师风采——Niklaus Wirth 25

第2章 词法分析 26

2.1 词法分析概述 26

2.1.1 词法分析的任务 26

2.1.2 单词的分类 28

2.2 词法分析器的设计 28

2.2.1 识别单词 28

2.2.2 转换图 29

2.2.3 构造词法分析器 31

2.3 词法分析器的实现 35

2.3.1 词法定义 35

2.3.2 构造转换图与转换表 36

2.3.3 相关数据结构 38

2.3.4 源代码实现 40

2.4 深入学习 44

2.5 实践与思考 45

2.6 大师风采——Dennis M. Ritchie 45

第3章 语法分析 47

3.1 程序设计语言的语法描述 47

3.1.1 上下文无关文法 47

3.1.2 推导 52

3.1.3 语法树 54

3.1.4 归约简介 57

3.2 语法分析概述 58

3.2.1 语法分析的任务 58

3.2.2 自上而下的语法分析法 59

3.2.3 构造语法分析器 64

3.3 语法分析器的实现 71

3.3.1 文法定义 71

3.3.2 语法分析表 76

3.3.3 源代码实现 86

3.4 深入学习	90
3.5 实践与思考	91
3.6 大师风采——Edsger Wybe Dijkstra	92
第4章 符号表系统	93
4.1 语义分析概述	93
4.1.1 程序设计语言的语义	93
4.1.2 语义分析与IR生成的任务	94
4.1.3 语法制导翻译	95
4.2 符号表设计	98
4.2.1 符号表概述	98
4.2.2 符号表的逻辑结构	99
4.2.3 符号表的实例分析	109
4.3 声明部分的实现	111
4.3.1 相关数据结构	111
4.3.2 主程序首部声明	113
4.3.3 包含文件声明部分	114
4.3.4 标号声明部分	118
4.3.5 常量声明部分	119
4.3.6 类型声明部分	120
4.3.7 变量声明部分	149
4.3.8 过程、函数声明部分	152
4.4 深入学习	154
4.5 实践与思考	155
4.6 大师风采——John Backus	155
第5章 中间表示	156
5.1 IR概述	156
5.1.1 IR的作用	156
5.1.2 IR设计及其级别	157
5.1.3 设计IR的重要意义	159
5.2 IR生成	160
5.2.1 三地址代码概述	160
5.2.2 Neo Pascal三地址代码的实现	164
5.2.3 翻译机制概述	168
5.3 语句翻译概述	170
5.3.1 语句翻译基础	170
5.3.2 翻译辅助函数及其实现	173
5.4 if语句	176
5.4.1 if语句的翻译	176
5.4.2 源代码实现	177
5.5 while/repeat语句	181
5.5.1 while语句的翻译	181
5.5.2 源代码实现	181
5.5.3 repeat语句的翻译	184
5.6 for语句	184
5.6.1 for语句的翻译	184
5.6.2 源代码实现	186
5.7 case语句	192
5.7.1 case语句的翻译	192
5.7.2 源代码实现	193
5.8 其他语句	199
5.8.1 break、continue语句的翻译	199
5.8.2 goto语句的翻译	201

5.8.3 asm 语句的翻译	204
5.9 深入学习	208
5.10 实践与思考	208
5.11 大师风采——Kenneth E. Iverson	209
第6章 表达式语义	210
6.1 表达式概述	210
6.2 类型系统基础	211
6.2.1 类型基础	211
6.2.2 类型系统	212
6.2.3 类型转换	217
6.3 类型系统的实现	218
6.3.1 类型系统的设计	218
6.3.2 IR的操作数	221
6.3.3 类型相容的实现	222
6.3.4 类型推断的实现	223
6.4 表达式翻译	226
6.4.1 表达式翻译基础	226
6.4.2 深入表达式翻译	229
6.4.3 表达式翻译的实现	230
6.5 操作数翻译	247
6.5.1 操作数的地址与形态	247
6.5.2 操作数翻译基础	248
6.5.3 简单变量操作数的翻译	252
6.5.4 记录字段操作数的翻译	262
6.5.5 数组翻译基础	265
6.5.6 数组元素操作数的翻译	270
6.5.7 指针运算的翻译	280
6.6 深入学习	286
6.7 实践与思考	286
6.8 大师风采——Alan Kay	287
第7章 优化技术	288
7.1 优化概述	288
7.1.1 什么是优化	288
7.1.2 优化级别	289
7.2 控制流分析	290
7.2.1 流图与基本块	290
7.2.2 流图的数据结构	292
7.2.3 流图的构造	293
7.2.4 优化的分类	297
7.3 数据流分析	298
7.3.1 数据流的相关概念	298
7.3.2 数据流分析的策略	298
7.3.3 活跃变量分析	299
7.3.4 ud链与du链	301
7.3.5 更多数据流问题	302
7.4 数据流分析的实现	303
7.4.1 定值点与引用点分析的基础	303
7.4.2 定值点、引用点分析的相关数据结构	305
7.4.3 定值点、引用点分析的实现	307
7.4.4 活跃变量分析的实现	312
7.4.5 ud链、du链分析的实现	314
7.5 常量传播与常量折叠	321

7.5.1 常量传播基础	321
7.5.2 常量传播的实现	324
7.6 复写传播	328
7.6.1 复写传播的基础	328
7.6.2 复写传播的实现	330
7.7 代数简化	333
7.7.1 代数简化基础	333
7.7.2 代数简化的实现	334
7.8 跳转优化	339
7.8.1 跳转优化基础	339
7.8.2 条件跳转优化的实现	341
7.8.3 连续跳转优化的实现	343
7.9 冗余代码删除	345
7.9.1 冗余代码删除基础	345
7.9.2 死代码删除的实现	346
7.9.3 不可到达代码删除的实现	348
7.10 深入学习	350
7.11 实践与思考	350
7.12 大师风采—Richard Stallman	351
第8章 运行时刻的存储管理	352
8.1 存储管理概述	352
8.1.1 存储区域	352
8.1.2 存储布局	354
8.1.3 存储分配基础	356
8.2 栈式存储分配	357
8.2.1 栈式存储分配基础	357
8.2.2 i386栈式存储分配	360
8.2.3 深入理解栈式存储分配	365
8.3 存储分配的实现	368
8.4 存储优化	372
8.4.1 存储优化基础	372
8.4.2 存储优化的实现	374
8.5 深入学习	381
8.6 实践与思考	382
8.7 大师风采—Bjarne Stroustrup	382
第9章 目标代码生成	383
9.1 目标代码生成概述	383
9.1.1 目标代码生成基础	383
9.1.2 指令选择	384
9.1.3 寄存器分配	385
9.2 目标机简介	386
9.2.1 目标机结构	386
9.2.2 浮点处理单元	387
9.2.3 操作数寻址方式	391
9.2.4 ptr操作符	392
9.2.5 一个完整的汇编程序	393
9.3 构造代码生成器	393
9.3.1 自动代码生成器基础	393
9.3.2 指令模板	394
9.3.3 寄存器描述	397
9.3.4 寄存器分配	398
9.3.5 代码生成器的基本结构	402

9.4 深入学习 413
9.5 实践与思考 413
9.6 大师风采——Peter Naur 413
第10章 GCC内核与现代编译
技术概述 414
10.1 编译技术的现状及发展 414
10.2 GCC内核分析 415
10.2.1 GCC的基本结构 415
10.2.2 GENERIC 416
10.2.3 GIMPLE 416
10.2.4 SSA 426
10.2.5 RTL概述 428
10.2.6 RTX 430
10.3 动态编译技术简介 436
10.3.1 动态编译技术基础 436
10.3.2 运行时特定化 437
10.3.3 动态二进制翻译 439
10.4 并行编译技术简介 441
10.4.1 并行编译技术基础 441
10.4.2 并行计算机及其编译系统 443
10.5 深入学习 446
10.6 大师风采——Alan Perlis 447
参考文献 448
• • • • • (收起)

[编译器设计之路](#) [下载链接1](#)

标签

编译原理

编译器

计算机科学

计算机

Compiler

程序语言

编译

语言

评论

<http://neopascal.sourceforge.net>

粗略翻了几章。

排版太不讲究了，代码居然不是用等宽字体，而且代码可读性差，命名难懂，还大量充斥着各种魔数，说实话还不如用伪代码，不过看上去至少作者在这方面还是有一定研究的，没乱写

这书的代码也太奇葩，讲解得也琐碎不清晰。如果不是国人写的，还挑战了下优化，就只值一星。

务必结合理论的那些书读起来，更是畅快务必！//完整源码，难能可贵

理论内容讲得不够高级，代码还是很不错的。

对编译器的关键技术有简单的实现

比部分国内书要好，但读起来没有上瘾的感觉。代码下下来跑不通，居然用类实例访问了stack类的protected变量，见Type.cpp，书中每章末尾都列举了很多拓展资料，蛮好。给中好评。

别被龙书绑架了，龙书老版的不知道，新版的翻译太挫了，而且一堆堆的概念，很晦涩。这本个人觉得相当不错，咬文嚼字的理论能用工程的思维来简洁明了的阐释清楚……当你读懂这本，再回过头去看龙书，那一定很轻松了。
PS：我没有否定龙书的地位，而在肯定这本书的价值，难得的国产好书~

这是我读过的最好的编译技术相关的书（少说7、8本）！跟国外的比呢，符合中国人的阅读习惯，虽然少了一些所谓的“高尖端”的理论，但把编译原理相关的基础介绍得如行云流水，来龙去脉都不少，这个是国外的龙书鲸书等没有的。至于国内的所谓院士教授编的那些个“书”，连提鞋都不配！他们顶多是剪刀手，哈哈！
难能可贵的是作者居然还提供源码，这是国内所谓教授能比的吗？

[编译器设计之路](#) [下载链接1](#)

书评

[编译器设计之路](#) [下载链接1](#)