DIY Compiler and Linker

自己动手写编译器、链接器

王博俊 张宇 编著

清华大学出版社

DIY Compiler and Linker

自己动手写编译器、链接器

王博俊 张宇 编著

清华大学出版社

内 容 简 介

本书讲述了一个真实编译器的开发过程，源语言是以 C 语言为蓝本，进行适当简化定义的一门新语言，称之为 SC 语言(简化的 C 语言)，目标语言是大家熟悉的 Intel x86 机器语言。在本书中，读者将看到从 SC 语言定义，到 SCC 编译器开发的完整过程。本书介绍的 SCC 编译器，没有借助 Lex 与 Yacc 这些编译器自动生成工具，纯手工编写而成，更便于学习和理解。为了生成可以直接运行 EXE 文件，本书还实现了一个链接器。读完本书读者将知道一门全新的语言如何定义，一个真实的编译器、链接器如何编写。

本书适合各类程序员、程序开发爱好者阅读，也可作为高等院校编译原理课程的实践教材。

图书在版编目(CIP)数据

自己动手写编译器、链接器/王博俊，张宇编著. --北京：清华大学出版社，2015

ISBN 978-7-302-38136-5

Ⅰ. ①自… Ⅱ. ①王… ②张… Ⅲ. ①C 语言－程序设计 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2014)第227835号

责任编辑：袁勤勇　徐跃进

封面设计：傅瑞学

责任校对：李建庄

责任印制：李红英

出版发行：清华大学出版社

网　　址：http://www.tup.com.cn，http://www.wqbook.com

地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084

社 总 机：010-62770175　　邮　　购：010-62786544

投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn

质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 刷 者：北京富博印刷有限公司

装 订 者：北京市密云县京文制本装订厂

经　　销：全国新华书店

开　　本：185mm×260mm　　**印　　张**：22.25　　**字　　数**：557 千字

版　　次：2015 年 2 月第 1 版　　**印　　次**：2015 年 2 月第1 次印刷

印　　数：1～2000

定　　价：44.50 元

产品编号：059284-01

序言

因为工作的关系，我经常和各企业的技术负责人交流。话题谈着谈着常常会转到他们目前共同的难题——技术人员招聘。这时不少人都会感慨，中国能做系统软件开发的技术人员太少，这方面的人太难找了。随着中国企业的发展，做系统和平台的需求不断增加，这种供需矛盾将越来越明显。

究其原因，很容易想到的是我们的高校教育、课程设置。美国顶尖大学计算机系基础课程教学里都非常重视项目实践，操作系统课往往要真的开发一个像模像样的操作系统原型，编译器课也真的要自己设计并实现一门有创新性的小语言……

在计算机科学的各门课程中，编译器的设计实践有着特殊的重要性。“龙书”的主要作者、哥伦比亚大学教授 Alfred V. Aho 曾经列举过编译器实践有诸多好处：

- 能让学生领悟到理论与实践的完美结合。比如编译原理所涵盖的正则表达式和自动机，在各种场合的应用是极其广泛的，对正则的掌握程度，从某种意义上讲甚至可以作为技术人员水平的一种尺度。
- 深入探索计算思维的多样性。与人类语言一样，不同类型的编程语言其实代表了不同的思维方式。只用过命令式语言的人可能没有想到，开启了大数据领域的 Map 与 Reduce，其实在函数式语言是一种非常常见的东西。

的确，深入了解编译器和编译原理，对于技术人员更好地理解和掌握自己最常用的语言和系统，从而提升自己的内力是有极大好处的。另一方面，随着 DSL（领域特定语言）的流行，需要技术人员开发自己语言的机会也越来越多。

然而，编译原理是计算机科学里公认比较难的一门课。虽然目前国外比较重要的编译理论教材（比如龙书的《编译原理》、虎书《现代编译原理》的 C 语言和 Java 版本、鲸书《高级编译器设计与实现》）基本上都有了中文版和英文影印版，但这些书往往更偏重理论，而且门槛较高，不太适合指导一线技术人员实践和自学。我认识的一位美籍华人技术专家 Ronald Mak 在 Wiley 出版过一本基于 Java 的“Writing Compilers and Interpreters”，比较贴近实践，但部头较大，而且没有看到中文版。

偶然的机会，我得知王博俊在工作之余，写了一本以简化的 C 语言为例子讲述编译器和链接器实践的书。浏览了初稿之后，感觉全书内容简明，容易上手，又不失全面和系统，正好弥补了这方面的空白。特向大家推荐。

CSDN 暨《程序员》杂志总编　刘江

2015 年 1 月

自序

纸上得来终觉浅，绝知此事要躬行。

——陆游

编译原理与技术的一整套理论在整个计算机科学领域占有相当重要的地位，学习它对程序设计人员有很大的帮助。我们考究历史会发现那些人人称颂的程序设计大师都是编译领域的高手，像写出 BASIC 语言的比尔·盖茨，Sun 公司的 Java 之父等，在编译领域都有很深的造诣。曾经在世界首富宝座上稳坐多年的比尔·盖茨也是从给微机编写 BASIC 语言编译器起家的，也正是这个 BASIC 编译器为比尔·盖茨和保罗·艾伦的微软帝国奠定了基础。这个编写 BASIC 语言编译器的经历，开启了比尔·盖茨的辉煌职业生涯。

编译器是一种相当复杂的程序，编写甚至读懂这样的一个程序都非易事，大多数的计算机科学家和专业人员也从来没有编写过一个完整的编译器。但是，几乎所有形式的计算都要用到编译器，而且任何一个与计算机打交道的专业人员都应掌握编译器的基本结构和操作。除此之外，计算机应用程序中经常遇到的一个任务就是命令解释程序和界面程序的开发，这比编译器要小，但使用的却是相同的技术。因此，掌握这一技术具有非常大的实际意义。

李国杰院士说："随着微处理器技术的飞速发展，处理器性能在很大程度上取决于编译器的质量，编译技术成为计算机的核心技术，地位变得越来越重要。我国要发展自己的微处理器事业，必然要有自己的编译技术作为后盾。"

回过头来说一说是什么样的原因使我萌生了写这样一本书的想法。作者学习其他计算机课程感觉没有特别难懂的，唯独看编译原理的教材，看完了云里雾里的，感觉一知半解，我感觉可能是学的教材过于理论化，于是到书店把所有跟编译原理有关的书籍统统买回家，当然这也包括大家公认的编译原理三大经典书籍（龙书、虎书、鲸书）在内，每一本我都从头到尾翻一遍，好像什么都懂了，又感觉要真的自己动手写个编译器仍然是只有大师才能完成，对自己还是可望而不可即的事情。并且作者也了解到许多关于编译原理实践的悲观论调："现有的编译器都是用 Lex 和 Yacc 构造的，从头开始手工编写一个完整的编译器几乎是不可能的。"可作者偏偏是那种"明知山有虎，偏向虎山行"的人，要知道早期的编译器可都是纯手工构造的，苦辣酸甜的征程就此开始，可是写个什么语言的编译器？这个编译器怎么定位？这一切都很茫然。

我开始研究编译原理书上的样例，希望能从中找到灵感，给上述问题找到答案。世界著名计算机科学家 N. Worth 编写的 PL/0 语言的编译程序是作者最先研究的编译器，它功能简单、结构清晰、可读性强，被认为是一个非常合适的小型编译程序的学习模型，可这个编译程序不支持数组、结构体、字符串，并且是以假想的栈式机器为例来编写的，而不是直接生成在某种 CPU，某种操作系统环境下直接可以运行的目标语言程序。"PL/0 语言的编译程序"作为编译器的学习模型，也只能算"矬子里面拔将军"，因为没有更好的，也只好将就着用了。至此，编译器定位问题算有了些眉目，作者希望构造一个更适合学习的编译器。

可是，另一个问题接踵而至，为什么那么多开源编译器不能直接用作编译器学习模型呢？我开始研究各个开源编译器的源代码，其中包括 GCC 的源代码，由于 GCC 支持多个前端语言和各种后端机器平台、AST（Abstract Syntax Tree）和 RTL（Register Transfer Language）又成了绕不过去的坎，还没学会怎么编写针对一种源语言、一种目标机器的编译器，就要去学习支持多种源语言多个机器平台的编译器，就好比一个婴儿还没学会走路就要学跑，这注定是要跌跟头的。

一面是过于简化的编译器学习模型，另一面是过于复杂的开源编译器，作为学习模型都不太合适。到这里，编译器定位问题算是彻底想清楚了，作者要构造一个教大家如何自己动手写编译器的学习模型。这个模型包括两大部分，第一部分是语言定义，第二部分是这个语言编译器的实现，这个编译器只支持一种源语言，目标语言也只支持一种。这个语言应该具备目前流行的高级语言的最主要特征。这个编译器要结构清晰，代码量要尽可能少，要能体现编写一个实用的编译器的完整过程与技术。这个编译器可以生成在操作系统中直接运行的 exe 文件，只要双击或在命令行执行就能看到结果的那种。

接下来作者开始思考另一个问题，编写个什么语言的编译器？作者研究了目前最流行的几种编程语言 C、C++ 、C＃、Objective-C、Java，其中 C 语言是最简单的了，只有 32 个关键字，但是作者研究发现，C 语言还是有许多冗余的成分，作为学习模型还可以更简单一些。作者最终以 C 语言为蓝本，进行适当简化定义了一门新的语言，仅有 15 个关键字，称为 SC 语言。目标语言选择大家熟悉的 Intel x86 机器语言，编译器命名为 SCC 编译器。

在本书中，读者将看到从 SC 语言定义，到 SCC 编译器开发的完整过程。读完本书你将知道一门全新的语言如何定义，一个真实的编译器如何编写，这些对你来说将不再神秘，编译原理讲的理论与本书中讲述的 SC 语言定义及 SCC 编译器开发过程，是理论联系实际在编译领域的最好阐释。

如本书作为编译原理实践教材，作者建议安排 10 学时讲授。

本书投稿后，有幸请 CSDN 暨《程序员》杂志总编、刘江老师阅读了本书的初稿，并为本书做序，在此向刘老师表示最衷心的感谢。

本书临近出版之际，承蒙清华大学王生原老师阅读了本书终稿，并对书稿做了中肯评价："本书特色鲜明，内容有深度，文笔也很不错，很值得出版。本书最大的特色是所选的目标平台，即 x86 处理器以及微软系统的 COFF 目标文件格式，这在教材中很少见到，可为国内的编译教学实践提供别具一格的素材。"同时，王老师还对本书提出了宝贵建议。在这里，向王老师表示由衷的敬意和最诚挚的感谢。

我还要感谢我的家人，他们的支持与鼓励是本书得以完成的保障。

要列出所有对本书出版有所帮助的人名是不可能的，因为有些困难是通过互联网解决的，我甚至不知道他们的名字。在此，谨向他们一并表示感谢！

最后，回想本书 6 年的写作历程，愿以蒲松龄的一副对联与读者共勉：

有志者，事竟成，破釜沉舟，百二秦关终属楚；

苦心人，天不负，卧薪尝胆，三千越甲可吞吴。

王博俊

2015 月 1 月

目录

第1章 引　言

世上无难事，只怕有心人

——民谚

编译器是将一种语言翻译为另一种语言的计算机程序。编译器将源语言编写的程序作为输入，而产生用目标语言编写的等价程序。通常，源语言为高级语言（面向人的语言），如C、C++ 、FORTRAN 等，而目标语言为机器语言（面向目标机的语言），如 Intel x86、ARM、MIPS、SPARC 等。

本书将和读者一起编写一个完整的 SCC（Simplified C Compiler）编译器，源语言是新定义的一门语言，称为 SC（Simplified C）语言，也就是简化的 C 语言，目标语言是 Intel x86 机器语言。SCC 编译器编译过程如图 1.1 所示。

图 1.1　SCC 编译器编译过程

让我们一起踏上编写 SCC 编译器的美妙旅程，一路上可能鲜花烂漫，也可能荆棘丛生，作者作为本次旅行的导游，接下来要先给大家讲一下旅行指南，让大家对本次旅程有个大概了解。

1.1　HelloWorld 编译过程分析

马克思主义认识论认为，认识是一个在实践基础上，由感性认识上升到理性认识，又由理性认识回到实践的辨证发展过程。本书的开头首先分析 SC 语言编写的 HelloWorld 程序的编译过程，以便对本书将要实现的 SCC 编译器的各个阶段的功能有个感性认识，第 2 章学习编写 SCC 编译器用到的编译原理知识，从第 3 章开始 SCC 编译器的实践过程。

1.1.1　HelloWorld 程序源文件

HelloWorld 作为所有编程语言的入门程序，占据着无法改变的地位，这个例程是从 Kernighan & Ritchie 合著的 *The C Programme Language* 开始有的，因为它的简洁、实用，可谓麻雀虽小，五脏俱全，一门语言的 HelloWorld 程序可以看作这门语言语法结构的一个缩影，因此后来几乎所有学习各种计算机语言的书都以 HelloWorld 程序作为学习这门语言的入门程序。下面就先看看用 SC 语言编写的 HelloWorld 程序。

```
1. /*************************************************************
2. * HelloWorld.c 源文件
3. *************************************************************/
4. int main()
5. {
6. printf("Hello World!\n");
7. return 0;
8. }
9.
10. void _entry()
11. {
12. int ret;
13. ret=main();
14. exit(ret);
15. }
16.
```

上面的程序似乎与C语言编写的没什么区别。但是,仔细一看会发觉用SC语言编写的HelloWorld程序多出一个_entry函数,它是干什么用的?从字面上理解应该是程序的入口点,没错,_entry是上面SC语言HelloWorld程序的真正入口点。可能你认为,看来SC语言程序与C语言的入口点是有区别的,SC语言程序的入口点是_entry,C语言程序的入口点是main函数。

讲述C语言的书上都说"main是C语言程序的入口",不知道大家是否对这句话的正确性怀疑过,是否深入探究过。在Visual C++下,控制台程序的入口函数是mainCRTStartup,由mainCRTStartup调用用户编写的main函数;图形用户界面(GUI)程序的入口函数是WinMainCRTStartup,由WinMainCRTStartup调用用户编写的WinMain函数;mainCRTStartup及WinMainCRTStartup函数的代码封装在已经编译好的lib库中,由链接器自动链接到生成的可执行文件中,所有这一切都是链接器悄悄干的,当然悄悄干的可不一定都是坏事,也可能无名英雄做好事。mainCRTStartup及WinMainCRTStartup函数就是这样的无名英雄,它们将该类程序开始都要做的一些必备且有技术含量的工作,悄悄地做了,并且背着广大程序员"悄悄"地做。SCC编译器是一个学习模型,目的是让可执行文件中的每一函数都由用户自己的代码产生,不要有那么多"潜规则",这样更有助于对编译过程的深入理解。

从SC语言编写的HelloWorld程序源代码中,大致可以了解以下内容:SC语言还是保持了C语言的绝大多数功能,支持函数,函数以{开始,以}结束,支持变量声明,变量可以为int型,函数可以返回int型值,也可以返回void表示没有返回值,支持函数调用,支持字符串的处理。

1.1.2 词法分析

词法分析器读入HelloWorld.c源程序字符流,将它们组织为一系列具有词法含义的单词,词法分析器将给每个单词编上号,以便为语法分析提供便利。HelloWorld程序的单词编码表如表1.1所示。

表 1.1　HelloWorld 单词编码表

单　词	编　码	单　词	编　码
关　键　字		常　量	
int	KW_INT	"Hello World!\n"	TK_CSTR
return	KW_RETURN	0	TK_CINT
void	KW_VOID	标识符	
运算符,分隔符		main	TK_IDENT+0
(	TK_OPENPA	printf	TK_IDENT+1
)	TK_CLOSEPA	_entry	TK_IDENT+2
{	TK_BEGIN	ret	TK_IDENT+3
;	TK_SEMICOLON	exit	TK_IDENT+4
}	TK_END		
=	TK_ASSIGN		

单词列比较清楚,就是程序中一个个独立单词,编码列中 KW_INT、TK_OPENPA、TK_CSTR、TK_IDENT 等,代表什么呢,它们都是单词枚举类型中的标识符,当成整型常量来理解就可以了。从表 1.1 中可以看出单词将被分为关键字、运算符、常量及标识符,每一个标识符都将有一个独立编码。

HelloWorld 单词编码看上去有些枯燥,好像也并不是很有意思,那么词法分析完成时有什么拿得出手的成果吗? 请看图 1.1,这里将关键字显示为绿色,运算符分隔符显示为红色,常量显示为黄色。

1.1.3　语法分析

语法分析是编译程序的核心部分。语法分析的作用是识别由词法分析给出单词序列是否是给定文法的正确句子(程序)。如果在语法分析阶段语法分析程序仅仅告诉我们"你的 HelloWorld.c 程序符合 SC 语言语法",恐怕多少有点令人沮丧,花了多少天辛辛苦苦写出来的语法分析程序,就能干这点事,恐怕我们一点成就感都没有,因为 HelloWorld.c 符合 SC 语言语法,我们早就知道的。那么费不少工夫写出来的语法分析程序能不能干点有技术含量的活呢? 请看图 1.3,语法分析程序实现了对 HelloWorld 源程序语法缩进功能。图 1.3 与图 1.2 有什么区别呢? 请读者仔细对比一下。有了语法缩进程序,用 SC 语言写的代码再乱也不用担心了,哪怕程序完全写在一行也没关系,只要用语法自动缩进程序处理一下就美观了。

1.1.4　语义分析

编译器必须要忠实地将 SC 语言写的程序编译成机器指令,在语义分析阶段 SCC 编译器要当好"傀儡",循规蹈矩,将"主子"(SC 语言)的意思忠实的传达给 Intel x86 处理器。在语义分析阶段将生成目标文件 HelloWorld.obj,它保存了语义分析的成果,来看一下目标文件生成过程,如图 1.4 所示。

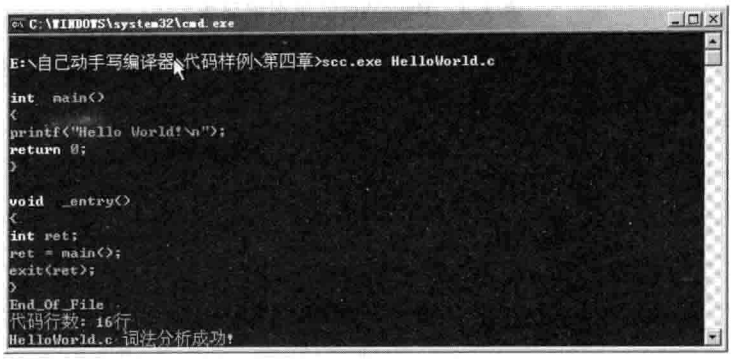

图 1.2 HelloWorld 词法着色成果展示

图 1.3 HelloWorld 语法缩进成果展示

图 1.4 HelloWorld 编译生成目标文件

图 1.5 是 HelloWorld.obj 文件内容，是不是看上去跟天书一样，这是我们看到的第一封天书，第 7 章、第 8 章和第 9 章将对这封天书从不同侧面进行全方位解读。

1.1.5 链接器

严格地说，链接器不属于 SCC 编译器的工作范畴，但 SCC 编译器如果就停留在此处，恐怕有些扫兴。上面的 HelloWorld.obj 文件只能算是个半成品，还没法直接执行，就像一家汽车生产企业，将轮胎、发动机、车身等零部件生产出来了，但是还没有组装，这样的汽车当然没法跑起来，链接器充当着组装车间的角色，它将 HelloWorld.obj 与 C 运行时库链接生成 HelloWorld.exe 可执行文件。这个链接及执行过程如图 1.6 所示。

这可是令人激动的时刻，具有里程碑的意义。SCC 编译器最终可以生成可执行文件，执行 HelloWorld.exe 就会在命令行显示“Hello World!”，同样的 HelloWorld.exe 执行结果，但心情大不一样，因为这可是用自己亲手写的 SCC 编译器编译生成的。

HelloWorld.exe 既熟悉，又神秘，熟悉的是，运行 HelloWorld.exe 就会在命令行打印出“Hello World!”，神秘的是这个文件中到底存着什么，这个文件为什么能够“指挥”计算机