

COMPILERS: PRINCIPLES AND GUIDED PRACTICE

编译原理实践 与指导教程

COMPILERS

PRINCIPLES

AND

GUIDED PRACTICE

许畅 陈嘉 朱晓瑞◎编著



计算机类专业
系统能力培养
系列教材

COMPILERS: PRINCIPLES AND GUIDED PRACTICE

编译原理实践 与指导教程

许畅 陈嘉 朱晓瑞◎编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

编译原理实践与指导教程 / 许畅, 陈嘉, 朱晓瑞编著. —北京: 机械工业出版社, 2015.5
(计算机类专业系统能力培养系列教材)

ISBN 978-7-111-50299-9

I. 编… II. ①许… ②陈… ③朱… III. 编译程序-程序设计-高等学校-教材
IV. TP314

中国版本图书馆 CIP 数据核字 (2015) 第 107402 号

本书面向开设计算机学科的大专院校, 提供一门接近实际 C/C++ 语言的 C-- 语言语法, 给出了详细的实验步骤和指导过程, 引导完成一个实际可用的编译器, 并提供了充分的测试样例来验证编译器实现的正确性。本书的实验设计包括词法分析与语法分析、语义分析、中间代码生成以及目标代码生成四个实验, 贯穿整个编译器设计的全过程。它具有接近实际、提供指导、帮助验证和难度可调四个特点, 并给出了详细的使用方式、时间安排和质量控制方案。

本书的适用读者包括开设计算机学科 (专业方向覆盖软件工程、系统软件等) 的大专院校教师和学生, 也适用于考虑规范化处理领域语言的工程技术人员。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 秦秀真

责任校对: 董纪丽

印刷: 北京瑞德印刷有限公司

版次: 2015 年 6 月第 1 版第 1 次印刷

开本: 185mm × 260mm 1/16

印张: 9

书号: ISBN 978-7-111-50299-9

定价: 29.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

丛书序言

——计算机专业学生系统能力培养和系统课程设置的研究

未来的5~10年是中国实现工业化与信息化融合,利用信息技术与装备提高资源利用率、改造传统产业、优化经济结构、提高技术创新能力与现代管理水平的关键时期,而实现这一目标,对于高效利用计算系统的其他传统专业的专业人员需要了解 and 掌握计算思维,对于负责研发多种计算系统的计算机专业的专业人员则需要具备系统级的设计、实现和应用能力。

1. 计算技术发展特点分析

进入本世纪以来,计算技术正在发生重要发展和变化,在上世纪个人机普及和Internet快速发展基础上,计算技术从初期的科学计算与信息处理进入了以移动互联、物物相联、云计算与大数据计算为主要特征的新型网络时代,在这一发展过程中,计算技术也呈现出以下新的系统形态和技术特征。

(1) 四类新型计算系统

1) **嵌入式计算系统** 在移动互联网、物联网、智能家电、三网融合等行业技术与产业发展中,嵌入式计算系统有着举足轻和广泛的作用。例如,移动互联网中的移动智能终端、物联网中的汇聚节点、“三网融合”后的电视机顶盒等是复杂而新型的嵌入式计算系统;除此之外,新一代武器装备,工业化与信息化融合战略实施所推动的工业智能装备,其核心也是嵌入式计算系统。因此,嵌入式计算将成为新型计算系统的主要形态之一。在当今网络时代,嵌入式计算系统也日益呈现网络化的开放特点。

2) **移动计算系统** 在移动互联网、物联网、智能家电以及新型装备中,均以移动通信网络为基础,在此基础上,移动计算成为关键技术。移动计算技术将使计算机或其他信息智能终端设备在无线环境下实现数据传输及资源共享,其核心技术涉及支持高性能、低功耗、无线连接和轻松移动的移动处理机及其软件技术。

3) **并行计算系统** 随着半导体工艺技术的飞速进步和体系结构的不断发展,多核/众核处理机硬件日趋普及,使得昔日高端的并行计算呈现出普适化的发展趋势;多核

技术就是在处理器上拥有两个或更多一样功能的处理器核心，即将数个物理处理器核心整合在一个内核中，数个处理器核心在共享芯片组存储界面的同时，可以完全独立地完成各自操作，从而能在平衡功耗的基础上极大地提高 CPU 性能；其对计算系统微体系结构、系统软件与编程环境均有很大影响；同时，云计算也是建立在廉价服务器组成的大规模集群并行计算基础之上。因此，并行计算将成为各类计算系统的基础技术。

4) 基于服务的计算系统 无论是云计算还是其他现代网络化应用软件系统，均以服务计算为核心技术。服务计算是指面向服务的体系结构 (SOA) 和面向服务的计算 (SOC) 技术，它是标识分布式系统和软件集成领域技术进步的一个里程碑。服务作为一种自治、开放以及与平台无关的网络化构件可使分布式应用具有更好的复用性、灵活性和可增长性。基于服务组织计算资源所具有的松耦合特征使得遵从 SOA 的企业 IT 架构不仅可以有效保护企业投资、促进遗留系统的复用，而且可以支持企业随需应变的敏捷性和先进的软件外包管理模式。Web 服务技术是当前 SOA 的主流实现方式，其已经形成了规范的服务定义、服务组合以及服务访问。

(2) “四化” 主要特征

1) 网络化 在当今网络时代，各类计算系统无不呈现出网络化发展趋势，除了云计算系统、企业服务计算系统、移动计算系统之外，嵌入式计算系统也在物联时代通过网络化成为开放式系统。即，当今的计算系统必然与网络相关，尽管各种有线网络、无线网络所具有的通信方式、通信能力与通信品质有较大区别，但均使得与其相联的计算系统能力得以充分延伸，更能满足应用需求。网络化对计算系统的开放适应能力、协同工作能力等也提出了更高的要求。

2) 多媒体化 无论是传统 Internet 应用服务，还是新兴的移动互联网服务业务，多媒体化是其面向人类、实现服务的主要形态特征之一。多媒体技术是利用计算机对文本、图形、图像、声音、动画、视频等多种信息进行综合处理、建立逻辑关系和人机交互作用的新技术。多媒体技术使计算机可以处理人类生活中最直接、最普遍的信息，从而使得计算机应用领域及功能得到了极大的扩展，使计算机系统的人机交互界面和手段更加友好和方便。多媒体具有计算机综合处理多种媒体信息的集成性、实时性与交互性特点。

3) 大数据化 随着物联网、移动互联网、社会化网络的快速发展，半结构化及非结构化的数据呈几何倍增长。数据来源的渠道也逐渐增多，不仅包括本地的文档、音视频，还包括网络内容和社交媒体；不仅包括 Internet 数据，更包括感知物理世界的数据。从各种类型的数据中快速获得有价值信息的能力，称为大数据技术。大数据具有体量巨大、类型繁多、价值密度低、处理速度快等特点。大数据时代的来临，给各行

各业的数据处理与业务发展带来重要变革,也对计算系统的新型计算模型、大规模并行处理、分布式数据存储、高效的数据处理机制等提出了新的挑战。

4) 智能化 无论是计算系统的结构动态重构,还是软件系统的能力动态演化;无论是传统 Internet 的搜索服务,还是新兴移动互联网的位置服务;无论是智能交通应用,还是智能电网应用,无不显现出鲜明的智能化特征。智能化将影响计算系统的体系结构、软件形态、处理算法以及应用界面等。例如,相对于功能手机的智能手机是一种安装了开放式操作系统的手机,可以随意安装和卸载应用软件,具备无线接入互联网、多任务和复制粘贴以及良好用户体验等能力;相对于传统搜索引擎的智能搜索引擎是结合了人工智能技术的新一代搜索引擎,不仅具有传统的快速检索、相关度排序等功能,更具有用户角色登记、用户兴趣自动识别、内容的语义理解、智能信息化过滤和推送等功能,其追求的目标是根据用户的请求从可以获得的网络资源中检索出对用户最有价值的信息。

2. 系统能力的主要内涵及培养需求

(1) 主要内涵

计算机专业学生的系统能力的核心是掌握计算系统内部各软件/硬件部分的关联关系与逻辑层次;了解计算系统呈现的外部特性以及与人 and 物理世界的交互模式;在掌握基本系统原理的基础上,进一步掌握设计、实现计算机硬件、系统软件以及应用系统的综合能力。

(2) 培养需求

要适应“四类计算系统,四化主要特征”的计算技术发展特点,计算机专业人才培养必须“与时俱进”,体现计算技术与信息产业发展对学生系统能力培养的需求。在教育思想上要突现系统观教育理念,在教学内容中体现新型计算系统原理,在实践环节上展现计算系统平台技术。

要深刻理解系统化专业教育思想对计算机专业高等教育过程所带来的影响。系统化教育和系统能力培养要采取系统科学的方法,将计算对象看成一个整体,追求系统的整体优化;要夯实系统理论基础,使学生能够构建出准确描述真实系统的模型,进而能够用于预测系统行为;要强化系统实践,培养学生能够有效地构造正确系统的能力。

从系统观出发,计算机专业的教学应该注意教学生怎样从系统的层面上思考(设计过程、工具、用户和物理环境的交互),讲透原理(基本原则、架构、协议、编译以及仿真等等),强化系统性的实践教学培养过程和内容,激发学生的辩证思考能力,帮助他们理解和掌控数字世界。

3. 计算机专业系统能力培养课程体系设置总体思路

为了更好地培养适应新技术发展的、具有系统设计和系统应用能力的计算机专门人才，我们需要建立新的计算机专业本科教学课程体系，特别是设立有关系统级综合性课程，并重新规划计算机系统核心课程的内容，使这些核心课程之间的内容联系更紧密、衔接更顺畅。

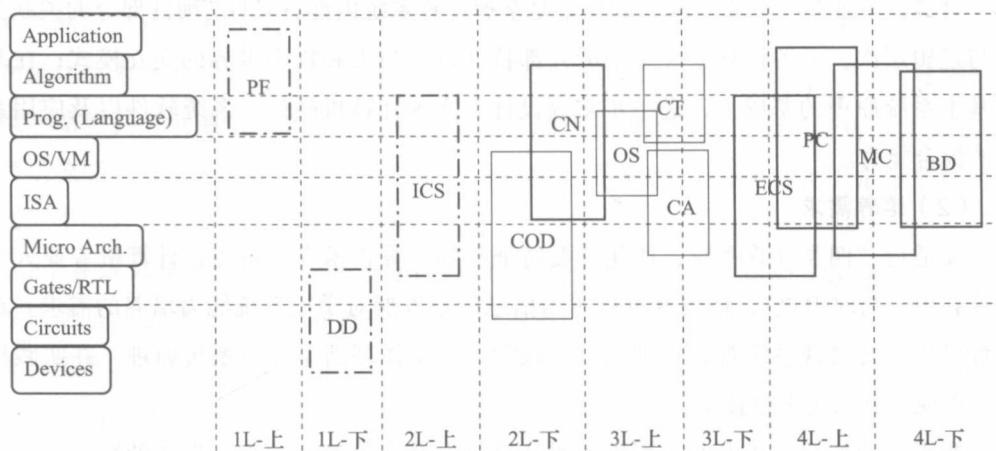
我们建议把课程分成三个层次：计算机系统基础课程、重组内容的核心课程、侧重不同计算系统的若干相关平台应用课程。

第一层次核心课程包括：“程序设计基础（PF）”、“数字逻辑电路（DD）”和“计算机系统基础（ICS）”。

第二层次核心课程包括：“计算机组成与设计（COD）”、“操作系统（OS）”、“编译技术（CT）”和“计算机系统结构（CA）”。

第三层次核心课程包括：“嵌入式计算系统（ECS）”、“计算机网络（CN）”、“移动计算（MC）”、“并行计算（PC）”和“大数据并行处理技术（BD）”。

基于这三个层次的课程体系中相关课程设置方案如下图所示。



图中左边部分是计算机系统的各个抽象层，右边的矩形表示课程，其上下两条边的位置标示了课程内容在系统抽象层中的涵盖范围，矩形的左右两条边的位置标示了课程大约在哪个年级开设。点划线、细实线和粗实线分别表示第一、第二和第三层次核心课程。

从图中可以看出，该课程体系的基本思路是：先讲顶层比较抽象的编程方面的内容；再讲底层有关系统的具体实现基础内容；然后再从两头到中间，把顶层程序设计的内容和底层电路的内容按照程序员视角全部串起来；在此基础上，再按序分别介绍计算机系统硬件、操作系统和编译器的实现细节。至此的所有课程内容主要介绍单处

理器系统的相关内容，而计算机体系结构主要介绍各不同并行粒度的体系结构及其相关的操作系统实现技术和编译器实现技术。第三层次的课程没有先后顺序，而且都可以是选修课，课程内容应体现第一和第二层次课程内容的螺旋式上升趋势，也即第三层次课程内容涉及的系统抽象层与第一和第二层次课程涉及的系统抽象层是重叠的，但内容并不是简单重复，应该讲授在特定计算系统中的相应教学内容。例如，对于“嵌入式计算系统（ECS）”课程，虽然它所涉及的系统抽象层与“计算机系统基础（ICS）”课程涉及的系统抽象层完全一样，但是，这两门课程的教学内容基本上不重叠。前者着重介绍与嵌入式计算系统相关的指令集体系结构设计、操作系统实现和底层硬件设计等内容，而后者着重介绍如何从程序员的角度来理解计算机系统设计及实现中涉及的基础内容。

与传统课程体系设置相比，最大的不同在于新的课程体系中有一门涉及计算机系统各个抽象层面的能够贯穿整个计算机系统设计及实现的基础课程：“计算机系统基础（ICS）”。该课程讲解如何从程序员角度来理解计算机系统，可以使程序员进一步明确程序设计语言中的语句、数据和程序是如何在计算机系统中实现和运行的，让程序员了解不同的程序设计方法为什么会有不同的性能等。

此外，新的课程体系中，强调课程之间的衔接和连贯，主要体现在以下几个方面。

1) “计算机系统基础”课程可以把“程序设计基础”和“数字逻辑电路”之间存在于计算机系统抽象层中的“中间间隔”填补上去并很好地衔接起来，这样，到2L-上结束的时候，学生就可以通过这三门课程清晰地建立单处理器计算机系统的整机概念，构造出完整的计算机系统的基本框架，而具体的计算机系统各个部分的实现细节再通过后续相关课程来细化充实。

2) “数字逻辑电路”、“计算机组成与设计”、“嵌入式计算系统”中的实验内容之间能够很好地衔接，可以规划一套承上启下的基于FPGA开发板的综合实验平台，让学生在统一的实验平台上从门电路开始设计基本功能部件，然后再以功能部件为基础设计CPU、存储器和外围接口，最终将CPU、存储器和I/O接口通过总线互连为一个完整的计算机硬件系统。

3) “计算机系统基础”、“计算机组成与设计”、“操作系统”和“编译技术”之间能够很好地衔接。新课程体系中“计算机系统基础”和“计算机组成与设计”两门课程对原来的“计算机系统概论”和“计算机组成原理”的内容进行了重新调整和统筹规划，这两门课程的内容是相互密切关联的。对于“计算机系统基础”与“操作系统”、“编译技术”的关系，因为“计算机系统基础”以Intel x86为模型机进行讲解，所以它为“操作系统”（特别是Linux内核分析）提供了很好的体系结构基础。同时，在“计算机

系统基础”课程中为了清楚地解释程序中的文件访问和设备访问等问题，会从程序员角度简单引入一些操作系统中的相关基础知识。此外，在“计算机系统基础”课程中，会讲解高级语言程序如何进行转换、链接以生成可执行代码的问题；“计算机组成与设计”中的流水线处理等也与编译优化相关，而且“计算机组成与设计”以 MIPS 为模型机进行讲解，而 MIPS 模拟器可以为“编译技术”的实验提供可验证实验环境，因而“计算机系统基础”和“计算机组成与设计”两门课程都与“编译技术”有密切的关联。“计算机系统基础”、“计算机组成与设计”、“操作系统”和“编译技术”这四门课程构成了一组计算机系统能力培养最基本的核心课程。

从“计算机系统基础”课程的内容和教学目标以及开设时间来看，位于较高抽象层的先行课（如程序设计基础和数据结构等课程）可以按照原来的内容和方式开设和教学，而作为新的“计算机系统基础”和“计算机组成与设计”先导课的“数字逻辑电路”，则需要对传统的教学内容，特别是实验内容和实验手段方面进行修改和完善。

有了“计算机系统基础”和“计算机组成与设计”课程的基础，作为后续课程的操作系统、编译原理等将更容易被学生从计算机系统整体的角度理解，课程内容方面不需要大的改动，但是操作系统和编译器的实验要以先行课程实现的计算机硬件系统为基础，这样才能形成一致的、完整的计算机系统整体概念。

本研究还对 12 门课程的规划思路、主要教学内容及实验内容进行了研究和阐述，具体内容详见公开发表的研究报告。

4. 关于本研究项目及本系列教材

机械工业出版社华章公司在较早的时间就引进出版了 MIT、UC-Berkeley、CMU 等国际知名院校有关计算机系统课程的多种教材，并推动和组织了计算机系统能力培养相关的研究，对国内计算机系统能力培养起到了积极的促进作用。

本项研究是教育部 2013 ~ 2017 年计算机类专业教学指导委员会“计算机类专业系统能力培养研究”项目之一，研究组成员由国防科技大学王志英、北京航空航天大学马殿富、西北工业大学周兴社、南开大学吴功宜、武汉大学何炎祥、南京大学袁春风、北京大学陈向群、中国科技大学安虹、上海交通大学臧斌宇、复旦大学张亮、机械工业出版社华章公司温莉芳等组成，研究报告分别发表于中国计算机学会《中国计算机科学技术发展报告》及《计算机教育》杂志。

本系列教材编委会在上述研究的基础上对本套教材的出版工作经过了精心策划，选择了对系统观教育和系统能力培养有研究和实践的教师作为作者，以系统观为核心编写了本系列教材。我们相信本系列教材的出版和使用，将对提高国内高校计算机类专业学生的系统能力和整体水平起到积极的促进作用。

“计算机类专业系统能力培养系列教材”编委会组成如下：

主 任 王志英

副主任 马殿富

委 员 周兴社 吴功宜 何炎祥 袁春风 陈向群 安 虹 臧斌宇 张 亮
温莉芳

秘 书 姚 蕾

此外，本系列教材的出版得到赛灵思电子科技有限公司和英特尔有限公司的支持。

“计算机类专业系统能力培养系列教材”编委会

2015年5月

前 言

本书与机械工业出版社于2009年出版的南京大学赵建华、郑滔和戴新宇所译的《编译原理》^①课本配合使用。该课本对应的英文版教材是美国哥伦比亚大学、斯坦福大学和Avaya实验室的Alfred V. Aho、Monica S. Lam、Ravi Sethi和Jeffrey D. Ullman所著的《Compilers: Principles, Techniques and Tools》，由于该书封面配有骑士和恐龙的图案，也被称为龙书。在下文中，如无特殊说明，课本均指该书（无论是中文翻译版还是英文原版）。

需要指出的是，虽然本书配合课本使用，但其内容已经包括所有相关的资料，因而它是内在完整的。这意味着，即使教学时使用其他编译原理相关的教材，本书仍能作为实践课程的教材，而不会出现所需资料不完整的情况。

设计思想

通常而言，编译原理实践课程较难设计，原因是其对应的理论教材以传授知识为主，离具体实践有较大的距离。而常规的实践课程大多会给出一门简易语言的语法，要求学生实现对应于该语言的编译器。由于缺乏规范化的指导，时常导致要么降低教学要求，允许学生较为随意地实现编译器；要么要求过高，致使学生们无从下手，从而对编译原理实践课程产生较大的抵触心理。针对这些问题，本书面向开设计算机学科的大专院校，提供一门接近实际C/C++的C--语言语法，给出了详细的实验步骤和指导过程，引导性地完成一个实际可用的编译器，并提供了充分的测试样例来验证编译器实现的正确性。

本书共分为四章，分别关注编译器设计的四个重要阶段——词法分析与语法分析、语义分析、中间代码生成以及目标代码生成。每章都给出具体的实验要求、实验指导以及测试样例等，它们共同覆盖了一个实用编译器的设计与实现的全过程。

本书的实验设计具有四个特点：一是接近实际，所采用的语言是C--，接近现实

^① Alfred V. Aho, 等. 编译原理 [M]. 赵建华, 郑滔, 戴新宇, 译. 北京: 机械工业出版社, 2009.

中常用的 C/C++，这使得所设计的编译器非常实用，甚至在特定的领域可以直接或经过少量修改后使用；二是配有指导教程，引导性地完成整个编译器的设计与实现，不会出现面对实验要求无从下手或不得不求助第三方资料的情况；三是具备验证帮助，提供大量的测试样例来验证编译器实现的正确性，而无需自行特别设计测试用例；四是难度可调，提供多种实验执行方案，既可统一难度要求，也可区分必做内容和选做内容，更可实现分组方案，使得每个组队实现不同的功能组合，激发学生的思考和锻炼协作能力。

使用方式

本书四章对应四个实验，前后贯穿，分别为词法分析与语法分析、语义分析、中间代码生成以及目标代码生成，简称为**实验一**、**实验二**、**实验三**和**实验四**。每个实验依赖于其前面的实验，需按顺序进行。

在四个实验中，除了最后的实验之外，前三个实验均有**必做部分**和**选做部分**，而选做部分又进一步分为几个不同的要求。具体而言，实验一的要求包括必做部分和三个选做部分（要求 1.1、要求 1.2 和要求 1.3），实验二的要求包括必做部分和三个选做部分（要求 2.1、要求 2.2 和要求 2.3），实验三的要求包括必做部分和两个选做部分（要求 3.1 和要求 3.2），最后的实验四则只有必做部分的要求。

这四个实验的设计特别考虑了不同院校的不同学生的能力和考核要求上的差异，具备多种使用方式。

1) **所有学生相同要求** 这是最简单的使用方式，适用于学习编译原理的所有学生需要通过相同考核要求的情况。具体而言，可细分为下面三种情况：

- 所有学生最低要求：完成所有实验的必做部分即可得满分。
- 所有学生最高要求：完成所有实验的必做部分和选做部分才可得满分。
- 所有学生自选要求：完成所有实验的必做部分即可得满分；若能完成实验的选做部分，则视完成情况给予额外奖励。

2) **不同学生不同要求** 这种使用方式适用于学习编译原理的学生需要通过不同考核要求的情况。比如强化班和普通班的学生一起上编译原理课程，则可要求强化班的学生完成所有实验的必做部分和选做部分才可得满分，而普通班的学生完成所有实验的必做部分即可得满分，若能完成实验的选做部分，则可获得额外奖励。

3) **不同组队不同要求** 这是最复杂的使用方式，适用于允许学生自由组队以共同完成实验要求的情况。推荐的组队规模是 1 ~ 3 人，如两人组队则为**正常模式**可获得实验满分，如三人组队则为**互助模式**需减少实验总分（如变为原来满分的 90%），如单

人组队则为**高手模式**可提高实验总分(如变为原来满分的110%)。在实验要求方面,仍可考虑不同的考核要求而选择不同的必做和选做部分,或者完成指定的或随机选择的实验要求。比如一位强化班的学生需要完成所有实验的必做部分和选做部分才可得满分,他可以单人组队进入高手模式以获得更高的总分,也可以两人组队进入正常模式以减少实验的难度。

4) **编译优化额外奖励** 实验三的设计特别考虑了编译优化的程度,即中间代码的执行效率(具体要求见实验三的实验内容部分)。编译优化属于额外奖励部分,如果中间代码的执行效率位于所有学生实验的前50%,则可获得实验三满份额外20%的奖励(即总分变为原来满分的120%);如果中间代码的执行效率位于所有学生实验的前20%,则可获得实验三满份额外50%的奖励(即总分变为原来满分的150%)。这部分额外奖励与前面的三种使用方式既不重叠也不冲突,可根据实际情况考虑是否采用。

时间安排

四个实验的安排依顺序进行,每个实验持续四周,每完成一个实验即可开始下一个实验,但实验一的开始可推迟两周,以等待对应的课本知识讲授。一般而言,如果一个学期时长18周,从第3周开始实验一,在第6周末结束实验一并在第7周开始实验二,在第10周末结束实验二并在第11周开始实验三,在第14周末结束实验三并在第15周开始实验四,在第18周末结束实验四以完成整个编译实践。如果一个学期时长更短或更长,可做相应调整。比如在更短的情况下,可以适当减少实验一的推迟时间(由两周减为一周)或实验三的时间(由四周减为三周)。

上面的实验时间安排是根据课本的知识体系而制订的,如果是其他编译原理教材,只要有下述的内容,也可模仿进行:

第1~6周:引论、词法分析、语法分析(实验一);

第7~10周:语法制导的翻译、中间代码生成(实验二);

第11~14周:中间代码生成、运行时刻环境、代码生成(实验三);

第15~18周:代码生成、机器无关的优化(实验四)。

如果由于教学上的特殊安排或别的原因,导致课本理论知识的讲解落后于实验安排,也可以采取推迟部分实验的结束时间并同时保证后续实验的开始时间的方法来安排实验。比如,若语法分析部分的课程讲授无法在第6周完成,那么可以允许实验一推迟一至两周结束(即在第7周或第8周结束),而实验二仍保持在第7周开始。这样既可以帮助部分学生更好地完成实验一(他们可能因为没有学习相应的课本知识而无

法进行实验一)，也可以允许另外的学生正常地开始实验二（与实验相关的内容在本书的实验指导部分已经提供，课本知识并不是必需的，但能帮助学生更好地理解知识）。

质量控制

编译原理实践的质量控制是指在完成实验的过程中，鼓励学生增强学习兴趣，提高所实现编译器的质量以及抑制不良行为（如作弊）的发生。本书的实验设计正有这方面的考虑：

1) **提高实验质量** 本书从测试用例和编译优化两方面来提高实验质量。

- **测试用例**：本书提供了大量测试样例来帮助学生自检编译器的实现是否符合实验要求，教学中可引入更多的额外测试用例（可根据实验要求和书中测试样例推导出）来进一步测试学生所实现编译器的质量，这将促使学生们考虑更多的细节来更好地实现编译器。
- **编译优化**：实验三的设计特别考虑了编译优化的额外奖励，这将鼓励学生们努力尝试以获得更多的额外奖励。

2) **抑制不良行为** 本书从分组实验和克隆检测两方面来抑制不良行为。

- **分组实验**：本书的实验设计了必做部分和选做部分，而选做部分又分为不同的要求，它们之间相互独立。在允许学生自由组队来协作完成实验内容的情况下，可进行必做部分和选做部分的随机组合。比如，根据抽签随机决定一组学生必须完成实验一的必做部分和选做部分的要求 1.1，而另一组学生必须完成实验一的必做部分和选做部分的要求 1.2，以此类推，后续的实验也可随机组合实验内容。这样可以增强不同组队间需要完成的实验内容的差异性，增加实验作弊的难度。也可以进一步要求不允许完成其他组队的选做实验（如前一组学生不能完成实验一选做部分的要求 1.2），这样也可以抑制不同组队之间的作弊行为（可通过测试用例区别出是否跨组完成了实验要求）。但要注意这将与第一种使用方式中的“所有学生自选要求”的情况相抵触，在实验执行中要注意。
- **克隆检测**：最后，本书的实验设计也允许进行实验代码间的克隆检测，特别是在分组实验的情况下，学生间实验代码几乎无法近似。为避免代码复制后经过简单变量名和函数名替换后重新提交为新的实验代码，建议采用**基于可执行代码的克隆检测**。采用必要的克隆检测可以进一步抑制学生实验作弊的行为。

以上介绍了本书编译原理实践的设计思想、使用方式、时间安排和质量控制。在使用过程中如有任何问题，欢迎和我们交流；如有不妥当之处，也敬请指出。

目 录

丛书序言

前言

第 1 章 词法分析与语法分析	1
1.1 实验内容	1
1.1.1 实验要求	1
1.1.2 输入格式	2
1.1.3 输出格式	2
1.1.4 测试环境	3
1.1.5 提交要求	3
1.1.6 样例 (必做内容)	4
1.1.7 样例 (选做要求)	7
1.2 实验指导	11
1.2.1 词法分析概述	12
1.2.2 GNU Flex 介绍	13
1.2.3 Flex: 编写源代码	14
1.2.4 Flex: 书写正则表达式	17
1.2.5 Flex: 高级特性	19
1.2.6 词法分析提示	21
1.2.7 语法分析概述	22
1.2.8 GNU Bison 介绍	24
1.2.9 Bison: 编写源代码	26
1.2.10 Bison: 属性值的类型	28
1.2.11 Bison: 语法单元的位置	30

1.2.12	Bison: 二义性与冲突处理	31
1.2.13	Bison: 源代码的调试	33
1.2.14	Bison: 错误恢复	35
1.2.15	语法分析提示	36
第 2 章	语义分析	38
2.1	实验内容	38
2.1.1	实验要求	38
2.1.2	输入格式	40
2.1.3	输出格式	41
2.1.4	测试环境	41
2.1.5	提交要求	41
2.1.6	样例 (必做内容)	42
2.1.7	样例 (选做要求)	48
2.2	实验指导	51
2.2.1	属性文法	52
2.2.2	符号表	53
2.2.3	支持多层作用域的符号表	56
2.2.4	类型表示	58
2.2.5	语义分析提示	61
第 3 章	中间代码生成	63
3.1	实验内容	63
3.1.1	实验要求	63
3.1.2	输入格式	66
3.1.3	输出格式	66
3.1.4	测试环境	67
3.1.5	提交要求	67
3.1.6	样例 (必做内容)	67
3.1.7	样例 (选做要求)	70
3.2	实验指导	73
3.2.1	中间代码的分类	74
3.2.2	中间代码的表示 (线形)	76

3.2.3	中间代码的表示 (树形)	77
3.2.4	初探运行时环境	78
3.2.5	翻译模式 (基本表达式)	81
3.2.6	翻译模式 (语句)	83
3.2.7	翻译模式 (函数调用)	84
3.2.8	翻译模式 (数组与结构体)	85
3.2.9	中间代码生成提示	86
第 4 章 目标代码生成		88
4.1	实验内容	88
4.1.1	实验要求	88
4.1.2	输入格式	89
4.1.3	输出格式	90
4.1.4	测试环境	90
4.1.5	提交要求	90
4.1.6	样例	91
4.2	实验指导	95
4.2.1	QtSPIM 简易教程	95
4.2.2	MIPS32 汇编代码书写	97
4.2.3	指令选择	100
4.2.4	寄存器分配 (朴素寄存器分配算法)	102
4.2.5	寄存器分配 (局部寄存器分配算法)	103
4.2.6	寄存器分配 (图染色算法)	104
4.2.7	寄存器分配 (活跃变量分析)	106
4.2.8	寄存器分配 (MIPS 寄存器的使用)	107
4.2.9	栈管理	108
4.2.10	目标代码生成提示	113
附录 A C—语言文法		115
附录 B 虚拟机小程序使用说明		122
附录 C 资源下载和安装介绍		125
参考文献		127