浅谈编译原理课程的定位

赵银亮
(西安交通大学 计算机系,陕西 西安 710049)

摘要:人才培养目标的多样性导致编译原理课程在课程体系中的核心地位变得越来越模糊,影响到课程体系的有效性。结合课程教学实践和对典型教学大纲的分析,从原理、算法、实现三个有关联的视觉,分析了课程相关性、人才类型、学科层次等因素对编译原理课程教学的作用,提出了该课程大纲的设计原则,初步设计了针对拔尖型、学术型和工程型人才的方案。经初步评估表明该方案能够适应学科现状对人才培养的要求。

关键词:编译原理;课程定位;课程体系;大纲设计
中图分类号:G642 文献标识码:A 文章编号:1007-3264(2010)04-0143-05

0 引言

编译原理课程是高等学校计算机专业的课程,一直作为专业核心课程为大学所采纳。就作者所在学校而言,自1977年计算机软件专业设置一来一直就作为专业基础课存在。在目前的课程体系中,随着计算机专业一部分原有的核心课程为其他学科专业所采用,更显示出本课程作为计算机专业独有课程的地位依然稳固。

作为新兴学科的计算机科学,应用面广,新知识层出不穷,课程体系势必需要频繁改革。如ACM针对计算学科课程体系做过几次推荐[1][2],多数高等院校均积极参照它来设计课程体系。在这个大背景下,我们从编译原理单门课程入手,研究课程内容及其与其他课程之间的关系,对于培养各类计算机人才是有意义的。

另一方面,随着高等教育向国际接轨,人才培养的针对性正在得到增强,从而能更好地满足社会需要。科学发展和人才培养目标决定了本门课程对计算机专业人才培养所起作用正在发生变化,人们对该课程定位存在着不同的看法,简单归类如下:

(1) 本课程在一些学校的课程体系没有出现或者被简化,即只涉及到编译器前端的部分内容。可以看出,这种情况下本课程正在退出其核心地位。

(2) 本课程的课程名称与内容出现了分化,课程名称强调理论性和普遍适用性,而课程内容针对小型编译器的实现。可以认为,与课程内容相比,本课程的名称给人留下了更多的想象空间。

(3) 这门课程与其他多门课程内容上发生重叠,导致或者将本课程看作是相关课程的综合演练环节或者把相关课程的内容过多地移动到本课程中。这些造成了课程内涵模糊、课程之间关系不清。

以上这些现象反映出编译原理课程在课程体系中的定位不够清晰明确,也反映出人们对这门课程在人才培养中能发挥什么作用存在较大的分歧。

本文根据相关的教学研究经历,对上述现象进行了分析。针对计算机科学发展以及专业人才培养状况,对本课程的定位问题进行了研究,对课程内容、与相关课程之间的关系等进行了优化设计,初步评估了其有效性。

1 课程教学现状概述

编译原理课程的内容概括地说就是"如何处理程序"的理论与方法。传统上,将编译原理解释为编

收稿日期:2010-06-07
资助项目:国家高技术研究发展计划资助项目(2008AA01Z136)
作者简介:赵银亮(1960-),男,陕西岐山人,西安交通大学计算机系教授。

万方数据

译程序构造原理,强调了编译程序本身如何编写,似乎涉及面比较狭窄,有悖于"原理"一词的普遍适用性内涵。MIT采用"程序语言工程",Stanford及其他一些学校采用"编译程序",也有采用"编译器设计","编译程序构造"等名称,这些名称与内容是相符的,然而相比较而言,没有采用编译原理名称,有局限性。事实上,编译原理课程中所包括的原理和方法应当是通用的,是可以离开编译器而独立存在的,这些原理和方法不仅是构造编译器所必须,对程序进行处理的场合都应该是可以应用的,比如语法分析、语义分析可用于软件的安全性分析,程序分析和优化技术可用于嵌入式系统、可信软件等[3]。因此,该课程教学中存在着两个视角值得考虑:一是看作对程序进行处理的一般原理与方法,弱化编译程序的编写;另一个是看作编译器实现技术。前者强调理论性而后者注重实践性。

ACM计算学科课程体系2001给出的课程CS240s(程序设计语言翻译)是:介绍有关程序设计语言翻译的理论和实践,内容包括编译器设计、词法分析、语法分析、符号表、声明及存储管理、代码生成、以及代码优化技术[2]。当参照制定教学大纲时,面临着很难完整地把它落实到编译原理课程之内,就理论和方法层面而言,量及难度超出了一般学生的承受范围;就编译器实现而言,量及难度更大。其结果是对理论和方法不得不做出剪裁,而对编译器编写而言,容易停留在"玩具程序"阶段[4]。

从近二十年计算机科学发展历史来看,高级语言处于稳定状态,导致编译器技术变得成熟而发展缓慢。随着并行和嵌入式体系结构的广泛应用,这种稳定状态正在发生变化,比如编译器在多核处理器编程以及安全可靠的复杂软件系统中扮演着关键的角色。因此,与"如何构造程序,如何处理程序"有关的原理、方法和技术有了发展空间[3],要求我们重新审视编译原理课程教学的定位问题。

2 编译原理课程与其他课程的关系

虽然本文不涉及课程体系设计,但编译原理与一些课程有相关性。无论相关课程是不是本课程的先修课,在内容上都可能与本课程有交集存在。考虑本课程定位的时候,需要划分出课程的边界,既要确定课程内容,同时也要确定出相关课程中与本课程有重叠的部分。

依据经典教材来确定现有教学体系中的课程内容是一条合理的途径,本课程教学中广泛采用的经典教材有龙书[5]、虎书[6]以及陈火旺等人的教材[7]等. 这些教材内容基本包含在ACM CC2001的CS240s[2]划定的范围之内。总体而言教材指定了一个较大的范围,我们有必要根据计算机科学发展做出设计。

ACM CC2001中将编译原理归为程序设计大类(或称课程模块),其中包括:程序设计原理,程序设计语言,形式语言与自动机,形式语义学等,表明它们之间存在很大的相关性;除此外,与本课程密切相关的还有:离散数学、数据结构、算法设计与分析、计算机组成与结构等。各课程与本课程相关内容见表1。

表1 相关课程内容

编号	课程名称	相关内容
M	离散数学	集合论、图论、代数系统、数理逻辑
D	数据结构	所有数据结构及其操作
A	算法设计与分析	不动点算法、重写系统、模式匹配、图树算法
P	程序设计原理	类型系统、解释器、虚拟机、数据抽象, Python/Lisp
L	程序设计语言	程序结构、作用域、编程风范、程序调试,C, C++, Java
F	形式语言与自动机	有穷自动机、上下文无关文法、正规式
O	计算机组成与结构	指令系统、结构模型、汇编语言
S	形式语义学	指称语义、操作语义

就目前的教学现状而言,课程P、L和F的相关内容(见表1)转移到本课程中讲授,课程负担无形中加大。虽然课程P、L都独立存在,但对本课程支持力度不够。而课程F一般不在课程体系中,则它的相关内容由本课程承担。就作者所在学校而言,课程P(第一学年第2学期)讲C语言,而课程L(第

万方数据

二学年第 1 学期)讲授 C+ +,另外第三学年第 1 学期还开设 Java 语言课,语言介绍和编程训练作为这些课程的重点,对程序设计的理论性和原理性内容不多涉及。所以研究本课程定位需要关注对相关内容的要求。

表 1 所列其他相关内容在相应课程中基本能够得到落实,需要强调的是课程 A 对本课程的支撑不能忽略,因为本课程中使用了多种算法,学生对这些算法没有认识的话,学起来吃力,效果也打折扣。本课程与编译器的目标机直接相关,目标机一般是在汇编语言级别上看到的机器,对于并行编译而言,还涉及到目标机的结构模型。而编译技术反过来支持课程 O,即支持软硬件协同设计。课程 S 提供描述程序语义的形式化手段,学生掌握了这门课程的知识,将对本课程在理论和原理方面有很好的支持,对于语义分析也是理论支持。

3　编译原理课程与人才培养目标关系

把授课对象作为决定课程定位的主要因素之一,其合理性是明显的。授课对象的特点影响对本课程的内容进行剪裁,甚至不选择本课程。授课对象的特点主要区分为学术型和工程型两类,同时与所在学校及学科的不同层次相关。

按照目前状况,人才培养的目标大致分为学术型人才培养和工程型人才培养。学术型人才在取得本科学位以后,将会继续学业到硕士和博士阶段,将来的主要职业以从事学术研究为主;而工程型人才或者本科毕业或者到硕士毕业,主要职业是在工业界发展,从事计算机有关开发和应用方面的工作。工程型人才的社会需求量一般远大于学术型人才,但是学术型人才培养对提高国家的学术竞争力至关重要。虽然本科培养方案中尚没有进行区分,但未来走向决定了不同类型的人才在本科阶段的基础应有区别。因此,按照培养目标可分为拔尖型人才、学术型人才以及工程型人才三类。编译原理课程定位有必要参照这三类人才培养要求来确定,对于拔尖人才培养要求全面和系统地掌握“构造和处理程序”的理论和方法;工程型人才培养重点在于编写编译程序的技术以及编译程序的结构设计等内容。学术型人才则介于二者之间。

学校及学科的层次定位是决定本课程定位另一个主要因素,它实际上划分了学生的范围,进而决定学生的平均接受能力的大小。学校及学科层次根据建设目标不仿大致分为国际高水平、国际水平和普通大学。第一类大学一般有拔尖人才培养任务;第一、二类大学学术型人才培养实力强;对于工程应用人才培养,由于社会需求很大,各类学校都会重视。

对于拔尖人才培养,编译原理的定位应该包含一个全面的、强调理论性的课程内容,类似于 Stanford 和 MIT 等一流大学的课程,目的是让学生在编程语言这个模块上打下系统深入的专业基础。对于学术型人才设计一个主要以强调理论性的课程内容的定位,而对应用型人才培养侧重于编译器设计和实现,打好开发实用编译系统的基础。

4　课程大纲的设计及评价

4.1　课程视觉

对课程内容从如图 1 所示的视觉进行观察,原理部分包括“如何处理程序”的一般的理论和方法,这是最基本的部分,算法是对原理和方法的一种过程性描述,而实现是指具体的程序设计语言实现,也就是程序。强调理论性的课程教学思路偏向于原理,而强调实践性的课程教学思路偏向于实现,都可以作为课程大纲设计的参照点。

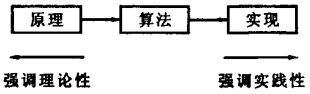

图 1　课程视觉

强调理论性的课程内容的设计倾向于理论和方法,算法次之,弱化程序实现,即可以认为算法和实现只不过是对编译原理的其他描述形式,编译原理课程的重点在如何学习编译原理本身,其他两个视觉是次要的。况且相关课程:算法设计与分析、程序设计语言等课程重点介绍了如何用算法来描述解决问题,以及如何用程序实现对问题的解决。

强调实践性的课程内容的设计倾向于编译程序本身,算法次之,弱化原理部分,即完全可以从剖析具体的编译系统入手,并结合算法达到对编译技术的理解和掌握。

4.2　设计原则

根据前面的分析可以知道确定编译原理课程大纲和内容需要综合多种因素,具体反映在如下六个设计原则中:

(1) 对学术型人才选择原理方法为主,对工程型人才强调实践性;

(2) 课内学时与课程内容的量匹配;

万方数据

(3) 相关课程中提供足够的支持;

(4) 课程内容与学生接受能力匹配;

(5) 学校和学科的层次与学生平均接受能力成正相关;

(6) 课程教学目标与人才培养目标相结合。

4.3 设计结果

设计结果如表2所示。由设计原则(3),如果相关课程同时在课程体系中存在则假定这些内容已经在那些相关课程中讲授并假定学生已达到熟练掌握程度;如果课程体系中没有,则这些内容归到本课程内容中,其结果是增加了课程内容的量,需要在增加课时数和裁剪课程内容上取得折中。

课程内容分为6个模块,模块1-2和4-5分别涉及前端和后端,模块3与前后端都有关,而模块6为课程实践内容。

表2 课程大纲设计方案

序号	模块名称	内容A. 强调理论性	内容B. 强调实践性
1	词法分析与语法分析	正规式*;上下文无关文法* 有穷自动机*;LL(1)和LR(k)分析方法	分析器的自动生成 上下文无关文法*
2	语义分析与中间代码生成	属性文法;中间表示;语言机制* 类型系统*;类型检查及推理*	属性文法;抽象语法树 典型语言机制*;自动语义分析
3	运行时系统	栈结构及存储管理I÷ 垃圾回收算法	栈结构及存储管理II÷ 符号表
4	代码生成	基本块和流图;代码生成算法 寄存器分配算法	代码生成器的生成器 DAG结构
5	代码优化	控制流分析;数据流分析;程序变换	代码优化初步
6	课程实践	Cool或MiniJava	典型编译系统

* 为相关课程内容

÷I指处理各类典型的过程活动、作用域、名字绑定;II指主流语言如C++/Java的过程活动、作用域和名字绑定

课程大纲设计方案与人才培养的三种目标的对应关系如下:学术型对应于表2中内容A一列,其中模块2、4和5参考学校和学科定位做出较大剪裁,同时由于表2中内容A一列带'*'号的内容可能不能够通过相关课程获得完全支持,因而占用本课内的学时。模块6指采用小型语言进行课内算法的练习,比较理想的是选择Cool或MiniJava,当然根据学校情况可以再简化或做出别的选择[8]。拔尖人才对应于表2中内容A一列,在课内学时以及相关课程支持方面得到保证。工程型人才对应于表2中内容B一列,同时参考学校和学科定位,增强对编译程序的结构分析的深度。

课程大纲设计中剪裁课程内容的途径表现在选取语言机制、确定目标机以及选取代码优化方法等几个方面。上述三个方面涉及的范围广泛,如何选取对课程内容的难度和量都有敏感的影响。

4.4 初步评价

分析国际一流大学如Princeton、Stanford和MIT的编译原理课程,其课程内容大致归为表2中内容A一列,即强调理论性的内容上,这是拔尖人才的设计方案内容。分析国内一流大学课程内容,少数学校所讲的前端部分与表2对应,后端部分涉及到基本的代码生成和代码优化内容。而其他开设本课程的学校中,一般只讲授到中间代码优化,离程序分析和优化有一定距离。考虑到程序分析和优化是目前研究和应用的热点,本方案中强调了这一部分。

对于工程型人才培养,目的是让学生掌握开发编译程序的知识和能力,因为,针对一个合适的典型编译系统开展教学十分必要,而对于理论成熟的前端,立足于自动生成工具的掌握,可以不必追求最初的原理和理论。对于一般大学而言也可以作为选修课。

在课程内容的剪裁上,针对不同程度的学生,在语言机制部分可选择面向对象的语言机制或者仅选面向过程的;在确定目标机时可选精简指令集如MIPS或者堆栈机器等;代码优化方法有很多种,比较简单的课程内容只涉及到典型代码优化思想;某些代码优化方法难度较大,再加上数据流分析及程序变换,构成了有难度的课程内容。

以上分析表明,表2所示的方案在一定程度上能支持不同人才培养目标以及不同层次大学对编译

万方数据

原理课程的教学内容要求。

5 结束语

本文从原理、算法、实现三个有关联的视觉,分析了课程相关性、人才类型、学科层次等因素对编译原理课程教学产生的影响,给出了编译原理课程大纲的设计原则并进行了初步设计,得到针对拔尖人才、学术型人才和工程型人才三个培养目标的设计方案。这个设计方案有强调理论性和强调实践性两个极端,前者能够充分体现"分析与处理程序"所需的原理和方法,有助于学生在计算机科学核心内容方面打下扎实的基础;而后者有助于学生在开发和设计编译系统方面打下扎实的基础,包括对基本技术的掌握和动手能力的培养。

参 考 文 献

[1] David Hemmendinger, The ACM and IEEE-CS Guidelines for Undergraduate CS Education[J]. Communications of the ACM,2007,50(5):46-53.

[2] Joint Task Force on Computing Curricula. Computing Curricula 2001 - Computer Science. Final report[DB/OL]. http://www. computer. org/portal/cms docs ieeecs/ieeecs/education/cc2001/cc2001.pdf.

[3] Mary Hall, David Padua, Keshav Pingali. Compiler research: the next 50 years[J]. Communications of the ACM. 2009,52(2):60-67.

[4] John Aycock, The ART of compiler construction projects [J]. ACM SIGPLAN Notices. 2003,38(12):28-32.

[5] A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman. Compilers: Principles, Techniques, and Tools (2nd Edition)[M]. Addison. Wesley, 2007 (赵建华等译,机械工业出版社,2009).

[6] Andrew Appel with Jens Palsberg, Modern Compiler Implementation in Java[M]. 2nd edition. Cambridge University Press, 2002.

[7] 陈火旺等.程序设计语言编译原理(第 3 版)[M].北京:国防工业出版社,2001.

[8] W. M. Waite. The compiler course in today's curriculum: three strategies [C]. Proceedings of the 37th SIGCSE technical symposium on Computer science education. Houston, 2006.

On Course Position of Compiling Principle

ZHAO Yin-liang

(Department of Computer Science,Xi'an Jiaotong University, Xi'an 710049,China)

Abstract: The position of the compiler course is becoming more and more blurry on the core of the whole curricular system because of the diversity of the purpose for education, which has been impacting the effectiveness of the curricular system. This paper, according to the instructional practice and analysis on typical syllabus, discusses the action on the teaching of compiler course made by every kind of factors like curricular correlativity, student characteristic, discipline hierarchy and so on, from the three relative perspectives: principle, algorithm and implementing, based on which this paper proposes the principle on designing the syllabus, and gives an initial plan aiming at instructing students in different patterns, including prominent talent, academic and project type. The preliminary evaluation shows that it fits to requirements of fostering goal based on current situation of the computer science.

Key words: compiling principle; course position; curriculum system; syllabus design

万方数据

浅谈编译原理课程的定位

作者： 赵银亮， ZHAO Yin-liang
作者单位： 西安交通大学,计算机系,陕西,西安,710049
刊名： 西安邮电学院学报
英文刊名： JOURNAL OF XI'AN UNIVERSITY OF POST AND TELECOM
年，卷(期)： 2010,15(4)

参考文献(9条)

1. W.M.Waite The compiler course in today's curriculum:three strategies 2006

2. 陈火旺 程序设计语言编译原理 2001

3. Andrew Appel;Jens Palsberg Modern Compiler Implementation in Java 2002

4. 赵建华 查看详情 2009

5. A.V.Aho;M.S.Lam;R.Sethi;J.D.Ullman Compilers:Principles,Techniques,and Tools 2007

6. John Aycock The ART of compiler construction projects[外文期刊] 2003(12)

7. Mary Hall;David Padua;Keshav Pingali Compiler research:the next 50 years 2009(02)

8. Joint Task Force on Computing Curricula Computing Curricula 2001-Computer Science

9. David Hemmendinger The ACMand IEEE-CS Guidelines for Undergraduate CS Education 2007(05)

本文链接：http://d.g.wanfangdata.com.cn/Periodical_xaydxyxb201004038.aspx