面向编译器实现的编译原理实践教学改革*

刘彬彬 李宏芒 唐益明 胡学钢 李建华 石雷

合肥工业大学计算机学院，合肥 230601

摘 要 根据计算机系统能力培养的要求，为了提高学生对现代编译器设计与实现的能力，本文提出一个面向编译器实现的编译实验方案。该方案围绕 LLVM IR 构建，包括不同难度的文法、不同编译阶段的实验任务以及多个可选的拓展实验。本文为每个实验任务提供相应的代码框架、测试用例以及考核方法，并将所设计的实验方案进行实践教学。最后，根据实际教学效果提出相应改进方案。

关键字 编译原理，编译器实现，LLVM IR，实验设计

Practical Teaching Reform of Compilation Principle Based on Compiler Implementation

Binbin Liu Hongmang Li Yiming Tang Xuegang Hu Jianhua Li Lei Shi

School of Computer Science and Information Engineering of Hefei University of Technology
Hefei 230601, China
binbin.liu@hfut.edu.cn

Abstract—To enhance students' capabilities in computer systems, particularly in modern compiler design and implementation, this paper proposes a compilation experiment scheme tailored towards compiler realization. This scheme revolves around LLVM IR, including experiments of varying degrees of difficulty, spanning across different compilation stages, along with multiple optional extension experiments. For each experimental task, corresponding code frameworks, test cases, and assessment methods are provided to facilitate learning. The proposed experimental scheme has been implemented in practical teaching. Finally, based on the actual teaching outcomes, corresponding improvement plans are proposed.

Keywords—Compilation principle, Compiler implementation, LLVM IR, Experimental design

1 引 言

计算机系统能力是计算机专业学生应具备的基本能力，主要涵盖学生对计算机系统的理解、设计、开发与应用的能力[1,2]。编译原理是一门计算机系统类核心课程，其涵盖丰富的理论和实践知识，对于初学者来说具有一定的难度和挑战性[3-6]。然而，编译原理的教学是理解计算机系统工作方式的关键环节之一。编译原理课程的目标不仅是让学生掌握基本的编译及优化知识，更重要的是通过课程实践，培养学生的软件开发能力、团队合作精神和项目管理能力[7-10]。

编译课程是计算机学科的核心课程之一，它不仅包括编译知识的传授，还包含相关的编译实验环节。为了提高学生对编译器系统的认识，编译实验体系不能只关注编译原理知识的编码复现，更应有现代编译器系统的实验环节。

*基金资助：本文得到教育部产学合作协同育人项目(2023-WB-BJ1087)资助。

为了适应时代的发展和要求，合肥工业大学编译原理课题组经过充分调研后设计一套面向编译器实现的教学实验方案。该实验方案以 LLVM IR 作为重要的媒介，通过选取特定的 IR 指令及其特性以降低实验门槛。考虑到国内正在大力发展自主可控的核心技术和工具，实验方案选择 LoongArch 体系结构作为编译器后端的硬件平台。通过实验方案的实施，学生可以实现一个实际的编译器，并在物理机上测试编译器的输出。此外，该实验方案提供多个拓展实验供学生选修，以满足不同学生的学习需求。该实验方案于 2023 年秋季学期的编译原理课程中进行了首次教学实践，在完成常规的编译实验(词法分析实验、语法分析实验和中间代码生成实验)后，5 支队伍成功地完成整个编译器的实现，1 支队伍完成编译优化拓展实验，取得良好的教学效果和实验反馈。

2 存在的问题与解决思路

合肥工业大学计算机专业的编译原理课程由理论讲授和实践环节组成，其中实践环节侧重于编译原理

知识的复现(如 LL 预测分析法、LR 语法分析、后缀表达式计算和 DAG 优化等),这使得学生对于编译器全貌缺乏有效的认知。另一方面，该编译实验体系已无法满足新时代对计算机系统能力培养的要求，也无法体现出其先进性和挑战性[11,12]。

为了解决以上挑战，编译原理课程组在参考众多高校实验方案的基础上[6,13-15]，秉持产教融合理念，从实际出发采取以下解决策略：

第一，设计面向编译器实现的实验方案，根据现代编译器设计原理，该方案包括编译器前端、中端和后端等多个实验任务，学员在完成各个实验任务后将得到一个实际可用的编译器。

第二，编译器实验选择 LLVM IR 作为重要媒介，使用多个现代编译相关的软件(如 Flex 和 Bison)，以此提高学生使用现代编译工具链的能力。

第三，为各个编译器实验任务提供相应的代码框架，要求学生在框架内完成编码，以此锻炼学生阅读代码的能力。

第四，在编译器实验的基础上，实验体系提供多种不同的拓展实验供学员选择，以期进一步提高学员的综合实践能力。

以上策略都是为了帮助学生更好地理解编译器系统，从而更好掌握编译原理相关知识，同时也能提升学员的工程实践能力。

3 面向编译器实现的编译实验方案

本节将详细介绍面向编译器实现的编译实验方案。首先,通过完成编译器前端、中端和后端的实验任务，学员将实现一个基本的编译器；其次，通过提供不同难度的文法，以满足不同学生的需求；最后，通过多个不同的拓展任务，以进一步提高学员的工程实践能力。该实验方案适用于计算机类专业的本科生，不仅能让学生学习现代编译器设计原理与特点，还能让学生初步积累大规模软件开发的经验。

3.1 实验内容

本文提出的编译原理实验体系总体设计思路如图 1 所示，该体系主要分为两部分：编译器实验和拓展实验。其中，编译器实验是指让学员实现一个完整的编译流程，从而得到一个实际可用的编译器；拓展实验是指在已实现的编译器基础上，挑战更高难度的实验任务。通过编译器实验，学生可以了解现代编译器的设计原理与基本架构；通过拓展实验，不仅加深学生对编译知识的理解与应用，而且可以促进编译知识与不同专业技能之间的交流。

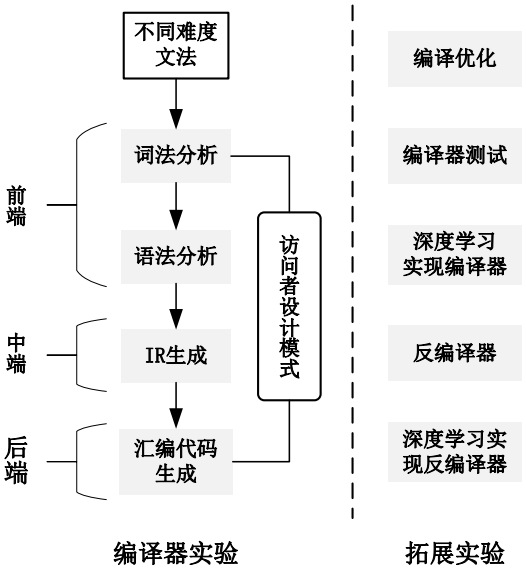

图 1 实验内容总体设计思路

3.2 编译器实验

编译器实验内容的设计旨在提升学生对编译理论知识的掌握和对现代编译器设计的认知。具体的实验任务如图 1 所示，其主要包含四个实验任务：词法分析、语法分析、IR 生成和汇编代码生成，相应的编译流程以及输入输出如图 2 所示。本项目为每一个编译器实验任务提供相应代码框架，学员需要根据要求在代码框架中完成实验任务。编译器实验的代码框架基于访问者设计模式[16]进行组织，这种结构使得代码更模块化，更易于学员理解和编程。

（1） 不同难度文法

本实验文法通过对 C 语言文法进行裁剪(去除结构体、多维数组、指针等复杂语法)，从而得到一个简单的 C 语言子集文法,该文法采用扩展的 Backus 范式表示。为了便于学生理解和实现，实验设计五种不同难度的文法，分别命名为 a--、a-、a、a+和 a++。考虑到篇幅所限，本文只展示最基础的 a--文法，如图 3 所示。

从图 3 可知,a--文法仅仅包含最基本的函数定义和 return 语句，只能形成最基本的函数单元。在此基础上，对文法不断进行丰富，最后得到一个 C 语言子集文法 a++,该文法具备多种数据类型(int 和 float)、多种分支结构(如 if-else)和多种运算操作(算术运算、逻辑运算和关系运算等)。

（2） 词法分析

对于学员选定的文法类别，该实验要求学员理解软件工具 Flex(快速词法分析器生成器)，补全相应 lexer.l 文件中的模式动作。然后利用 Flex 生成对应

文法的词法分析器，借助生成的词法分析器完成对于给定输入程序的分析工作，具体的输入输出形式如图 2 所示。

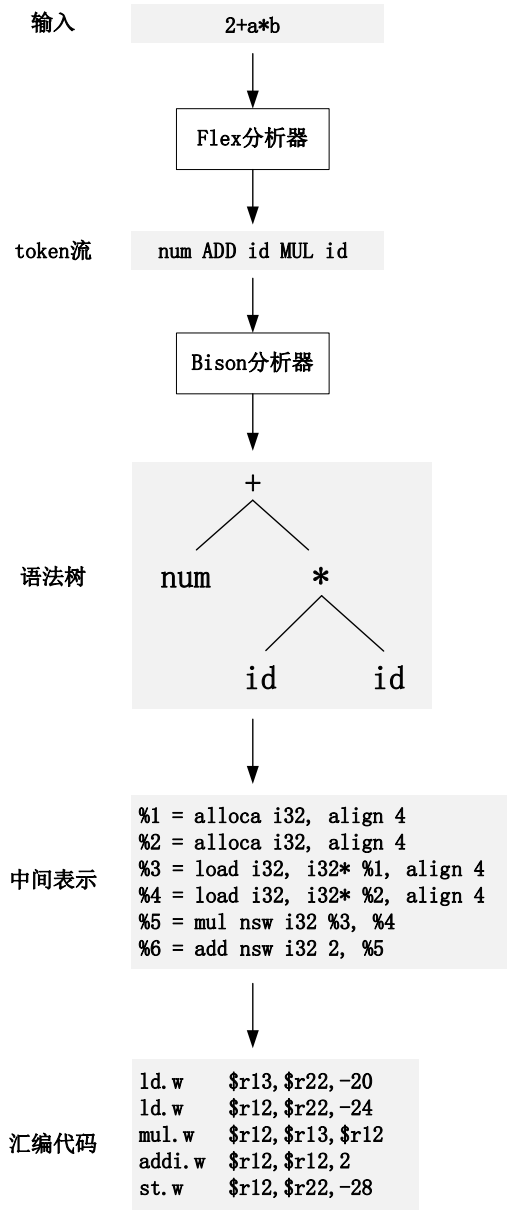

图 2 编译流程图

编译单元	CompUnit → [CompUnit] FuncDef
函数定义	FuncDef → FuncType **Ident** '(' ')' Block
函数类型	FuncType → 'void' | 'int'
语句块	Block → ' { ' { BlockItem } ' } '
语句块项	BlockItem → Stmt
语句	Stmt → ' return ' [Exp] ' ; '
表达式	Exp → AddExp
加减表达式	AddExp → MulExp
乘除表达式	MulExp → IntConst

图 3 a--文法

（3） 语法分析

该实验要求学员学习并理解文法生成器 Bison。在此基础上，通过对 Flex 生成的 token 流进行分析(如图 2 所示)，利用 Bison 生成相应的语法分析树。借助 Flex 和 Bison 完成对选定文法的测试用例分析工作。

（4） IR 生成

该实验以 LLVM IR 作为编译器的中间表示形式。本文所述的 IR 并不是标准 LLVM IR，实验方案在 LLVM IR 的基础上进行相应裁剪，只保留与本实验体系文法相关的基本 IR 指令和特性。相应地，学员只需要学习和理解 LLVM IR 的基础知识，包括静态单赋值格式(SSA)、IR 常用的指令类型及其表示形式等知识。

参考 LLVM 生成 IR 的过程和逻辑，通过遍历 Bison 生成的抽象语法树，在给定的代码框架上要求学员生成基于 LLVM IR 格式的中间表示(如图 2 所示)，并使用 LLVM 相关工具对生成的 IR 进行正确性测试。

（5） 汇编代码生成

本实验任务要求实验者根据指定的体系结构，生成对应的汇编代码。本实验体系目前采用 LoongArch 架构，需要学生理解龙芯体系结构的二进制接口 API，包括基本数据类型、寄存器内容和栈帧约定等知识。

在理解 LoongArch 体系结构的基础上，通过在给定的代码框架中完成指令选择、指令调度和寄存器分配等任务，针对中间表示产生正确的汇编代码(如图 2 所示)。最后，将生成的汇编代码在龙芯服务器上进行实际运行和测试。至此，学员将得到一个可实际使用的编译器。

3.3 拓展实验

在编译器实验的基础上，本实验体系额外设计多个不同类型的拓展实验，如编译优化、编译器测试和反编译等任务，以期进一步提高学员的综合实践能力。拓展实验面向所有学生选修，以满足不同专业与兴趣学生的需求。

（1） 编译优化

本实验基于已有的编译器实验，要求学生在此基础上实现相关的编译优化方法。考虑到实验时间有限以及编译优化的实现难度，本文选取 mem2reg、死代码删除、常量传播、函数内联以及局部子表达式删除等优化方法作为实验任务。在实现相关优化方法后，利用相应测试用例集合对编译器进行正确性和性能测试，并对结果进行统计分析。

（2） 编译器测试

从图 1 和图 2 可知，按照功能测试的原则，针对已经实现的编译器至少应有以下 3 种功能测试：

第一，生成符合文法要求的合格程序，测试编译器前端生成的抽象语法树是否正确。

第二，生成符合要求的抽象语法树，测试编译器中端生成的中间表示代码是否正确。

第三，生成符合要求的中间表示代码，测试编译器后端生成的汇编代码是否正确。

此外，对编译器整体要进行集成测试，借此评估编译器的正确性和性能。因此，针对以上功能测试和性能测试的相关要求，根据软件测试相关知识，要求学生编写相应测试用例生成方法，借此生成大量自动化测试用例，从而实现编译器自动化评测功能。

（3） 深度学习实现编译器

针对已经实现的编译器，根据编译器测试用例生成方法和测试结果，学员将可以收集成千上万个不同的<程序，汇编代码>对。将这些<程序，汇编代码>对作为数据集，要求学生选用适当的数据预处理方法、深度学习模型和优化算法，实现一个能够完成高级程序编译的深度学习模型。该模型的输入为程序代码，输出为相应的汇编代码。最后，通过测试集对实现的深度学习模型进行功能和性能评测，以此深入理解形式化方法与深度学习技术的优缺点。

（4） 反编译器实现

面向编译器生成的汇编代码，要求学员根据相关反编译分析原理(如数据流分析)，通过逆向分析重建原程序的数据结构和程序结构。在实现反编译器后，通过未公开的测试用例集对其进行功能性测试，以确保反编译器的正确性。

（5） 深度学习实现反编译器

针对实现的反编译器，根据反编译器功能测试结果，学员将可以收集多个不同的<汇编代码，程序>对。将这些<汇编代码，程序>对作为数据集，要求学生选用适当的数据预处理方法、深度学习模型和优化算法，实现一个能够完成汇编代码反编译的深度学习模型。该模型的输入为汇编代码，输出为相应的高级语言代码。最后，通过测试集对实现的深度学习模型进行评测，以检测模型的正确率。

4 实验组织与实施

4.1 实验环境与工具

本实验的配置如表 1 所示，相关的实验框架已开源在 gitee 仓库。由于本实验不涉及大型软件的调试和安装(如对 LLVM 源码的调试)，学生在笔记本电脑上即可完成相关实验任务的编码工作。

表 1 实验环境配置表

实验环境	版本
Ubuntu	22.04
gcc/g++	11.4
gdb	12.1
cmake	3.15
Git	2.34
Flex	2.6
Bison	3.8
LLVM	14.0

所有的实验任务、测试用例以及评测机都已部署在希冀平台，其可以自动化完成对学生提交代码的评测和判分。本实验针对每一种文法，分别构建相对应的测试用例，每一类测试用例的分值权重为 20%。此外，本实验体系提供超 100 页的实验阅读材料，包括实验环境配置、实验任务简介、相关工具介绍和实验评测体系等内容。

4.2 实验组织与考核方式

依据不同的教学要求和教学对象，本实验体系可灵活组织相应的必修和选修实验任务，单人或组队(不超过 3 人)合作完成实验任务。考核方式包括实验评测考核、实验报告撰写和小组答辩(可选)三部分。实验评测考核由评测机给出具体分数，并由助教对实验结果进行确认；实验报告由老师对学生所撰写的报告进行总体评价和打分；小组答辩由组长代表队伍进行实验汇报答辩，由相关专家和老师对答辩人进行提问和打分。

4.3 实验效果与分析

在 2023 年秋季学期，在完成必修实验(编译器实验中的前端和中端任务)的基础上，共有 5 支队伍完成编译器实验，1 支队伍完成编译优化实验。其中，对于学员实现的编译器，每支队伍都通过功能性测试；对于编译优化实验，该队伍完成 mem2reg 的中端优化功能。对这 6 支队伍额外进行现场答辩和评分，由外校专家和本校老师进行综合评判。

通过问卷调查和现场答辩，学生们对该实验体系给予积极肯定的评价：

① 对个人能力是一次充分的锻炼，特别是对于 C++编程能力提升很大；

② 学习更多现代编译工具链，积累了大型软件的开发经验；

③ 对于编译器的实际流程有较为深刻的理解和认识。

同时，也有学生反映各种问题：

① 实验代码注释不够详细，导致代码理解有困难；

② 实验难度较大，希望给予更多时间进行编码；

③ 实验材料对于重难点知识介绍不够突出，希望能重点讲解实验难点。

4.4 实验改进方向

针对上述教学反馈，编译原理课程组将在后续的实验体系建设上进行如下改进：

第一，详细注释框架代码。在原有代码注释的基础上，增加对于代码算法的注释，从而帮助学生在宏观上把握和理解代码框架。

第二，合理设计实验时间。由于编译实验理论难度大，实践要求高，需要教师合理安排各个实验的时间长短和截止日期，以期减轻相关实验任务给学生带来的学习压力。

第三，完善实验阅读材料。在已有的实验阅读材料基础上，针对学生提出的各种问题，进行针对性改进，如：①对于环境安装和相关工具使用，额外增加视频材料，方便学员学习和理解；②对于实验的重难点部分，将对其进行详细介绍和重点讲解。

5 结束语

本文注意到现有实验教学体系中存在的不足，为了提高学员的计算机系统能力，建立一套面向编译器实现的编译实验体系。通过编译器实验，强调学生借助相关工具实现一个实际编译器；通过拓展实验，以进一步提高学员的工程实践能力。在实验实施的过程中，学员加深了对现代编译器系统的认知，而且有效促进学生对于相关工具(如 Git 版本管理和 cmake 构建工具等)以及设计模式的学习和掌握。

参 考 文 献

[1] 谭志虎， 秦磊华， 胡迪青. 面向系统能力培养的计算机专业实践教学模式[J]. 中国大学教学，2017(9)： 80-84.

[2] 陈智勇. 计算机科学与技术专业学生系统能力培养的改革与实践[J]. 计算机教育, 2019(3): 58-61.

[3] 蒋宗礼. “编译原理”课程与专业能力培养[J]. 计算机教育, 2009 (21): 4-6.

[4] Yanxiang He, Zhuomin Du, Hanfei Wang. Research on the Knowledge and Ability dual-driven Teaching Model for the Course of Compilers Principles[C] //2020 15th International Conference on Computer Science & Education (ICCSE). IEEE, 2020: 24-29.

[5] Zhang Y, Hu C, Zeng M, et al. Encouraging compiler optimization practice for undergraduate students through competition[C]//Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1. 2021: 4-10.

[6] 徐伟, 刘硕, 张昱. 基于 LLVM 驱动程序的编译原理实践教学改革[J]. 软件导刊, 2023, 22(6)：191-195.

[7] 田玲, 余盛季, 王晓斌, 等. 《编译原理》 实验教学改革及探索[J]. 实验科学与技术, 2013, 11(6): 297-299.

[8] 余芳，王晓明 赵森. 基于创新思维培养的编译原理实验教学改革 [J].大学教育，2019（12）：45-47.

[9] 余月，李凤霞，陈宇峰，等 . 计算机编译原理课程虚拟实验设计与实践 [J]. 实验技术与管理，2019，36（8）：123-126.

[10] 史涯晴. 突出编程能力培养的编译原理课程教学改革[J]. 计算机教育, 2022(9): 105-108.

[11] 吴岩. 建设中国“金课” [J]. 中国大学教育，2018（12）：6-11.

[12] 蔡朝晖，陈伟清，贺莲，等 . 面向赋能教育的计算机系统课程口袋实践体系建设 [J]. 计算机技术与教育学报，2023，11（4）：22-25.

[13] 李清安，袁梦霆，王汉飞，等. 基于 LLVM 的编译实验课程设计 [J]. 计算机教育，2019（2）：11-14.

[14] 张昱，桑榆扬 . 引入开源编译器 LLVM 的编译原理课程改革 [J]. 计算机教育，2017（2）：62-67.

[15] 刘兵，张辰，谢红侠，等 . 基于 Clang+LLVM 架构的编译原理课程教学探索 [J]. 计算机教育，2020（1）：42-49.

[16] 伽玛. 设计模式: 可复用面向对象软件的基础[M]. 机械工业出版社, 2019.