

**2018年大学生科技立项**

**结 题 报 告**

**（B类科技作品的设计与创作作品）**

**项目名称：《**基于JAVA的编译器开发**》**

**项目负责人：蒲宝剑**

**项目成员：王东山、段晓旭、申强、吕娜**

**负责人学院、班级：计算机学院，软件1603班**

**2018年11月**

|  |
| --- |
| 1. 研究目的及意义   编译器是将高级语言程序翻译成处理器可以识别的机器语言。因此，编译器是联系应用开发与实际运行环境的重要支撑软件。本项目组将要选用Java语言开发编译器，是由于Java语言对正则表达式和集合运算等形式语言有较好的支持，并且在Windows, Linux，android等系统上具有较好的移植性。本项目组选用用RISC-V模型机作为目标处理器，即通过设计RISC-V的编译器，可以将C语言程序翻译成在RISC-V处理器上的运行的目标程序，达到快速开发部署的目的。  RISC-V 处理器指令集于2014年正式推出，是一个开放的、标准的、现代的、没有专利问题和历史包袱的全新指令集。RISC-V指令集并且以BSD许可证发布。任何公司和个人都可以在自己的产品中免费使用，而修改也无需开源。RISC-V不仅能让公司收益，也能够让大学和研究机构更好地研究新的处理器技术和架构。  虽然Linux下已经有GCC-RISCV编译器工具软件，使用时只需make就可以编译了。但本项目组立足现状着眼未来，开发基于Java语言的编译器可以更好地适应自主设计的RISC-V处理器，更好地优化目标代码，能够更好地提供多处理器互联解决方案，对应用更好地提供支撑。即为用户提供最为优质便捷的编译器产品，并且开放开源，以用户需求定制解决方案。    图1 系统架构设计图  如图1所示，我们在项目的整体设计方案上采用分层开发模式。从顶层的应用开发到底层的硬件支持提供一整套支撑方案。在整个系统的发展规划中，各层均可独立开发，通过标准接口，下层为近邻上层提供支持。整个项目硬件方面除  了设计处理器CPU外，能够支持标准总线协议，包括中断控制、总线仲裁，将键盘、显示器、存储器等常用外设连接进来。整个项目在软件方面设计专用的基于Java开发的C语言编译器，可以支持运行操作系统。操作系统设计将核心层与应用层分离。设计专用的核心函数库，支持界面管理、OpenGL图形库、标准C语言函数库等。  在工程化方面，我们先以“试验田”的方式，在小范围内进行推广，我们计划先在高校内部推广，主要依靠同学和老师的使用和反馈。建立完善的培训体系，并为在校学生电子应用竞赛提供免费支持。在不断开发和应用中，对产品进行完善，并适当考虑今后的商业化。即首先在高校的教师和学生免费使用，并提供专业支持。通过小规模的使用，发现和找出问题，再进行改进，然后商业化。另外，本项目可在军工领域使用，对国家具有很高的战略价值，因此预计会受到国家政策方面的优惠。 |
| 1. 和本课题有关的国内外研究现状分析（包括发展水平和存在的问题等）   编译器开发具有极高的实用价值和意义，高级语言编译器的性能决定了基于JAVA 语言平台所开发出的软件质量，所以国内外很多大学的科研和技术人员也在积极地开展这方面的技术探索和项目实践，他们大多是以特定的软件项目为背景来进行一些与编译器开发相关或类似的研究分析，他们的研究目标大多是基于某种实验型高级语言的编译器开发和优化改进。  最近十年以来，国外关于编译器设计的发展动态主要体现在：首先，编译器采用了大量的更加复杂的算法，主要用于推断或简化程序中的信息，这又与更为复杂的程序设计语言的发展结合起来，其中经典的有用于函数语言变异的Hindley-Milner类型检查的统一算法，其次，编译器已经越来越成为基于窗口的可视化交互开发环境，连接程序，调试程序以及项目管理程序等，已经成为了事实上的编译器行业标准，另一方面，尽管国内外的专家学者们近年来在编译原理领域进行了大量的研究，但是基于编译器设计原理在近20年中都没有多大变化，它现在成为计算机科学课程中的中心环节之一。 |

|  |
| --- |
| 1. 研究内容和方法   设计并实现一个简单的单处理器的编译器设计，要求能够将C语言程序编译成机器语言程序。首先编译器进行词法分析，也就是要把那些字符串分离出来；接下来进行语法分析,即在词法分析的基础上将单词序列组合成各类语法短语；然后进行语义分析，就是把各个由语法分析分析出的语法单元的意义搞清楚；最后生成的是目标文件，即可在CPU上运行的可执行文件。    图2 编译器设计结构图  (1)词法分析逐个读构成源程序的字符，把它们组成词法记号流；  (2)语法分析把词法记号流依照语言的语法结构按层次分组，以形成语法短语；  (3)语义分析阶段检查程序的语义正确性，并为以后的代码生成阶段收集信息；  (4)语法分析和语义分析后，某些编译器产生源程序的显式中间表示，可以认为这种中间表示是一种抽象机的程序。中间表示必须具有两个性质：它易于产生并且易于翻译成目标程序；  (5)代码优化阶段试图改进代码，以产生执行较快的机器代码。如果中间代码生成算法比较简单的话，它给代码优化留了很多机会；  (6)编译的最后一个阶段是目标代码生成，它生成可重定位的机器代码或汇编码。此阶段为源程序所用的每个变量选择存储单元，并且把中间代码翻译成等价的机器指令序列；  (7)符号表管理和出错管理是编译过程中的两项重要工作，它们与词法分析、语法分析、语义分析、中间代码生成、代码优化和代码生成这六个阶段相互作用；  (8)每个阶段都有可能发现源程序的错误。在发现错误后，该阶段必须处理此错误，使得编译可以继续进行，以便进一步发现源程序的其他错误。 |
| 1. 创新之处和可预期的创造性成果   我们在项目的整体设计方案上采用分层开发模式。从顶层的应用开发到底层的硬件支持提供一整套支撑方案。在整个系统的发展规划中，各层均可独立开发，通过标准接口，下层为近邻上层提供支持。整个项目硬件方面除了设计处理器CPU外，能够支持标准总线协议，包括中断控制、总线仲裁，将键盘、显示器、存储器等常用外设连接进来。整个项目在软件方面设计专用的基于Java开发的C语言编译器，可以支持运行操作系统。操作系统设计将核心层与应用层分离。设计专用的核心函数库，支持界面管理、OpenGL图形库、标准C语言函数库等。开发基于Java语言的编译器可以更好地适应自主设计的RISC-V处理器，更好地优化目标代码，能够更好地提供多处理器互联解决方案，对应用更好地提供支撑。即为用户提供最为优质便捷的编译器产品，并且开放开源，以用户需求定制解决方案。设计该编译器更为符合我们自主开发的RISC-V处理器设计需求，包括目标代码优化功能，多处理器之间访问功能，指令集范围界定等功能。 |
| 1. 采取的研究方法、技术路线   严格按照之前撰写项目需求报告的执行流程初步往下推进，首先编译器进行词法分析，也就是要把那些字符串分离出来；接下来进行语法分析,即在词法分析的基础上将单词序列组合成各类语法短语；然后进行语义分析，就是把各个由语法分析分析出的语法单元的意义搞清楚；最后生成的是目标文件，即可在CPU上运行的可执行文件。 |
| 1. 研究进度   2018.05.01—2018.05.30已经完成有关Java技术开发工具和RISC-V指令集的收集和整理。  2018.06.01—2018.06.30已经完成基于Java的编译器界面设计和数据传送。  2018.07.01—2018.08.30已经完成编译器的简单C语言语句的词法分析，并能够输出结果。  2018.09.01—2018.09.30已经完成编译器的简单C语言语句的语法分析，并能够输出结果。  2018.10.01—2018.10.30已经完成编译器的简单C语言语句的语义分析，并能够输出结果。  2018.11.01—2018.11.30已经完成编译器的简单C语言语句的中间代码生成和优化和初步的目标代码和简单的javaGUI界面加以支持。 |

|  |
| --- |
| 1. 研究成果   已经能够将基本C语言程序编译成目标处理器的机器语言程序中的中间代码部分，并对部分程序进行的优化处理。该编译器更为符合我们自主开发的RISC-V处理器设计需求，包括目标代码优化功能，多处理器之间访问功能，指令集范围界定等功能，通过对界面的设计和对此方法的分析和后期的中间代码的优化，汇编指令，机器指令的执行，使得程序能够实现基本的编译器功能。  已经能够通过输入C语言程序代码，输出其标准的中间代码，并做了部分目标代码和中间的优化过程，使得项目在不断的推进，也通过对此项目的开发，学习了开发编译器的基本技巧和现有编译器中的一些不足之处，希望能够在后期，通过团队的合作，使其得到优化和改进。 |
| 八、课题组成员及分工   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 姓 名 | 性别 | 学院 | 班级 | 项目中的分工 | 签章 | | 蒲宝剑 | 男 | 计算机学院 | 软件1603班 | 系统整体分析与设计 |  | | 王东山 | 男 | 计算机学院 | 计科1506班 | 词法和语法分析 |  | | 段晓旭 | 男 | 计算机学院 | 计科1506班 | 语义分析与中间代码生成 |  | | 申强 | 男 | 理学院 | 信息1502班 | 目标代码生成与优化 |  | | 吕娜 | 女 | 计算机学院 | 计科1605班 | 系统测试与撰写文档 |  | |