

**文献摘要**

|  |  |
| --- | --- |
| 学 院 | 计算机科学与技术学院 |
| 专 业 | 计算机科学与技术 |
| 班 级 | 计算机1401 |
| 姓 名 | 李开心 |
| 指导教师 | 林泓 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2018 | 年 | 3 | 月 | 4 | 日 |

阅读15篇文献（含2篇外文），每篇文献写200-300字文献摘要

文献1：

Christopher W.Fraser，David R.Hanson．王挺，黄春，等．可变目标 C 编译器：设计与实现[M]．北 京：机械工业出版社，2016.11．

摘要：

本书系统地介绍了可变目标ANSIC编译器lcc的设计方法和实现技术。lcc编译器是一个具有产品级质量的用于研究的C编译器，在UNIX界广为流行。本书结合lcc的具体实现，详细讲述了存储管理、符号表、词法分析、语法分析、中间代码生成、优化、目标代码产生等编译程序的各个部分。本书特别针对3种目标机器（MIPSR3000、SPARC、Intel386）介绍了代码生成器的设计。通过学习上述内容，读者可以深入了解产品级编译程序设计中的许多关键技术，对于如何设计和实现一个实用的编译器有具体、真切的认识。

文献2：

Brian W.Kernighan，Dennis M.Ritchi．徐宝文，李志．C 程序设计语言[M]．北京：机械工业出版社， 2004.1．

摘要：

C语言的限制少，通用性强，使用方便且效率高，因此C语言的使用十分广泛。本书是由C语言的设计者Brian W. Kernighan和Dennis M. Ritchie编写的著作，全面、系统地讲述了C语言的各个特性及程序设计的基本方法，包括基本概念、类型和表达式、控制流、函数与程序结构、指针与数组、结构、输入与输出、UNIX系统接口、标准库等内容。学习完前6章就可以了解C语言的基本结构及语法，并可以编写一些小程序。从第7章开始介绍C语言的标准库，以及UNIX系统接口，对编写大型C语言程序提供了支持。附录中有C语言的定义，如词法规则、语法符号、表达式、声明、语句等。还有对标准库的更加全面而简洁的描述。

阅读15篇文献（含2篇外文），每篇文献写200-300字文献摘要

文献3：

Mayur Pandey．LLVM Cookbook[M]．Birmingham：Packet Publishing，2015.5．

摘要：

本书以任务驱动的方式编写基于LLVM 的编译器前端、优化器、后端。有丰富的实例来讲解LLVM 的架构，并详细地讲述了如何使用LLVM 来编写编译器。

本书第一章介绍了LLVM的设计与使用，详细地描述了LLVM的模块化设计和LLVM工具。第二章通过详细的实例来讲述如何编写一门语言的前端。第三章介绍了JIT的支持以及如何扩充前端。第四章开始介绍后端优化以及代码生成。实现一门语言时，可以只需编写编译器前端，后端工作交由LLVM来完成，简化编译器的实现过程。通过这本书，可以学会许多编译器的技术，了解编译选项以及编译过程。

文献4：

王爽．汇编语言[M]．北京：清华大学出版社，2013.9

摘要

汇编语言是各种CPU提供的机器指令的助记符的集合，人们可以用汇编语言直接控制硬件系统进行工作。本书全面介绍了8086汇编语言的各个方面，从指令的含义与使用，到内存、CPU、寄存器等计算机底层知识的讲解。

第一章介绍了汇编语言的相关知识、计算机内部总线、各种存储芯片等。第二章到第十章讲述了如何编写汇编语言，如各种汇编指令的使用，以及程序中各个段的意义。第十一章到十七章则对计算机内部的原理进行了更加深入的讲述，如内中断如何产生、端口如何进行读写、使用BIOS进行键盘输入与磁盘读写等。通过阅读本书，可以学习如何使用汇编语言，并且能够对计算机内部工作原理有更加深入的了解。

阅读15篇文献（含2篇外文），每篇文献写200-300字文献摘要

文献5：

Bryant,Radal E．Computer Systems: A Programmer’s Perspective[M]．London: Pearson，2015.6

摘要：

本书是从程序员的角度来写的关于计算机系统的书，讲述应用程序员如何利用系统知识来编写出更好的程序。本书分为三个部分，全面讲解了计算机系统的方方面面。

第一部分程序结构和执行，介绍了在计算机系统中信息的表示和处理、程序的机器级表示、处理器体系结构等内容。第二部分在系统上运行程序，是对链接、异常控制流、虚拟内存的讲解。第三部分程序间的交互和通信，介绍了系统级I/O、网络编程、并发编程等概念。通过这些内容，读者可以深入理解程序内部、系统内部的工作原理，建立完整的计算机系统的知识体系。

文献6：

Alfred V.Aho，Monica S.Lam，Ravi Sethi，Jeffrey D.Ullman．赵建华，郑滔，戴新宇．编译原理[M]．北 京：机械工业出版社，2009.1．

摘要

本书全面、深入地探讨了编译器设计方面的重要主题，包括词法分析、语法分析、语法制导定义和语法制导翻译、运行时刻环境、目标代码生成、代码优化技术、并行性检测以及过程间分析技术，并在相关章节中给出大量的实例。本书不仅介绍了编译器构造的基本原理和技术，还详细介绍了一些有用的编译器构造工具，比如介绍了Lex和Yacc的工作原理和使用方法。不仅如此，本书涵盖了编译器开发方面的最新进展，每章中都提供了大量的系统及参考文献。

阅读15篇文献（含2篇外文），每篇文献写200-300字文献摘要

文献7：

Dick Grune，Ceriel J．H.Jacobs．Parsing Techniques: A Practical Guide[M]．New York：Springer-Verlag， 2008．

摘要：

本书介绍了各种解析技术，对不同的解析技术及其相互关系和实用性（包括错误恢复方法）进行了详细的讨论。与大多数书籍不同，它不仅介绍了流行的内容如Chomsky文法以及常用解析方法，还介绍了很多小众的如耦合文法、属性和前缀文法、命令文法等以及相应的解析方法。本书有48页的系统参考，包含400多个条目，提供了一个通用的上下文无关解析器，并进行了详细讨论。

文献8：

范志东，张琼声．自己动手构造编译系统：编译、汇编与连接[M]．北京：机械工业出版社，2016.8．

摘要

本书描述的编译系统不仅包含编译器的实现，还包括汇编器、链接器的实现，以及机器指令与可执行文件格式的知识。本书将描述的重点放在编译系统的关键流程上，并对工业化编译系统的实现做了适当的简化，实现了一个基于Intelx86指令集的编译系统。

本书结合程序代码的主要部分详细阐述了编译系统的实现原理和过程，并对编译器、汇编器、编译优化器涉及的关键算法、数据结构和程序实现流程，以及ELF文件的格式、Intel指令格式均作了详细的说明，并结合大量的图表，展示了编译系统工作过程中代码信息的流动和存储格式的变化。

阅读15篇文献（含2篇外文），每篇文献写200-300字文献摘要

文献9：

王博俊，张宇．自己动手写编译器、链接器[M]．北京：清华大学出版社，2015.2．

摘要：

本书讲述了一个真实编译器的开发过程，源语言是以C语言为基础进行了适当的简化定义的一门新语言，称为SC语言，目标语言是Intel x86机器语言。本书讲述了从SC语言的定义到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、Yacc等编译器自动生成工具，纯手工编写而成。前三章对SC语言进行了完整的定义，第四章到第七章讲述了SCC编译器的前端部分，第八章到第十章完成了编译器的后端部分，并实现了一个链接器用于将多个目标文件连接为一个可执行文件。

文献10：

Stanley B.Lippman．侯捷.深度探索 C++对象模型[M]．北京：电子工业出版社，2012.1．

摘要

有两个概念可以解释C++对象模型。1.语言中直接支持面向对象程序设计的部分。2.对于各种支持的底层实现机制。本书则专注于对第二个概念的介绍。第一章分析了实现类机制的几种对象模型的区别，来说明对象模型对程序的影响。第二章到第四章则是介绍类中构造函数、成员变量、成员函数在编译器内部的实现机制。第五章则讲解了对象的构造、析构、拷贝的实现方式，深入理解这些操作的含义。第六章介绍执行期语义学，第七章介绍模板、异常处理、执行期类型识别等。通过这些内容，将能消除我们队C++（及其面向对象支持）的各种错误认识你，并清楚地了解软件组件设计上的难点与运用之道。

阅读15篇文献（含2篇外文），每篇文献写200-300字文献摘要

文献11：

Scott Meyers．Effective Modern C++[M]．南京：东南大学出版社，2015.9．

摘要：

本书彻底介绍了C++11和C++14中引入的语言特性，例如auto型别推导、移动语义、lambda表达式，以及并发支持，以及如何高效地运用这些特性，从而使软件具备正确性、高效率、可维护性和可移植性。涵盖以下主题：大括号初始化、noexcept规格、完美转发，以及智能指针的make函数的优缺点。std::move、std::forward、右值引用和\*引用之间的联系。编写整洁、正确，以及高效的lambda表达式的方法。std::atomic和volatile有怎样的区别，它们分别用于什么场合，以及它们和C 的并发API有何联系。

文献12：

Scott Meyers．Effective C++[M]．北京：电子工业出版社，2011.1．

摘要

学习程序语言根本大法是一回事；学习如何以某种语言设计并实现高效程序则是另一回事。本书是一本轻薄短小高密度的“专家经验积累”，目的是告诉读者如何高效地运用C++。本书一共组织了55个条款，谈论如何强化你的程序和设计。每个条款都描述了一个写好C++的方式，且其背后都有具体的例子予以支持，例如用对象管理资源以便对象析构时资源自动释放，在operator=中处理“自我赋值”以避免自我赋值导致的资源泄露或使用空指针等问题。不仅给出结论，还给出了详尽的解释，让读者能知其然且知其所以然。

阅读15篇文献（含2篇外文），每篇文献写200-300字文献摘要

文献13：

Nicolai M.Josuttis．C++标准库[M]．北京：电子工业出版社，2012.9．

摘要：

C++标准化始于1989年，由国际标准化组织（ISO）推动。C++标准库提供了一组公共类和接口，极大地拓展了C++语言核心功能。本书涵盖了所有全新的C++11标准库组件，包括：并发、分数运算、时钟和定时钟、Tuple、新STL容器、新STL算法、新智能指针、新locale facet、随机数和分布、Type trait和使用工具、正则表达式。因为STL对实际开发十分重要，本书使用了大量篇幅对STL进行了详尽的介绍。本书还介绍了标准库中的新式C++编程风格及其影响，包括Lambda表达式、基于区间的for循环、move语义以及可变参数模板。在本书的末尾还给出了参考书目，给需要进一步研究学习C++标准库的读者提供了建议。

文献14：

Stanley B.Lippman，Josee Lajoie，Barbara E.Moo．C++ Primer[M]．北京：电子工业出版社，2013.9

摘要

C++是一种面向对象的计算机程序设计语言，支持多种程序设计风格。本书全面介绍了C++11的内容，所有的例子和讲解都遵循C++11标准。本书第一部分讲解C++基础，即C++语言的基本类型、表达式、语句、函数以及类，能让读者快速了解C++语言的基本概念。第二部分则是对C++标准库的描述，详细介绍了标准库的几大组件：IO库、顺序容器、泛型算法、关联容器，以及动态内存，C++标准库为大型C++程序的开发提供了基础。第三部分详细介绍类设计的工具，如拷贝控制、运算符重载、面向对象，以及泛型编程。最后一部分则是一些高级主题，特别是C++11新添加的特性，如Tuple、随机数等。通过仔细阅读本书，可以学会C++语言的大部分内容，并习得良好的程序设计规范。

阅读15篇文献（含2篇外文），每篇文献写200-300字文献摘要

文献15：

Mark Allen Weiss．风舜玺．数据结构与算法：C 语言描述[M]．北京：机械工业出版社，2004.1．

摘要：

本书讨论数据结构和算法那分析，目的是同时教授学生良好的程序设计技巧和算法分析能力，使得他们能够开发出高效率的程序。第二章讲解算法分析，用数学的手段分析证明算法的正确性、复杂度，从而为后续算法提供性能评估的基础。后续章节则介绍了常用的数据结构，如列表、栈、队列、树、图、散列，并对其涉及的算法进行了详细的讲解，如树的遍历算法、排序算法、图的最短路径算法等。不仅如此，还讲解了五类常用算法：贪婪算法、分治算法、动态规划、随机化算法和回溯算法。每章末尾还给出了详细的参考文献，对于需要深入了解某一内容的读者提供了建议。