<<编译原理>>

图书基本信息

书名：<<编译原理>>

13位ISBN编号：9787040133677

10位ISBN编号：7040133679

出版时间：2003-1

出版时间：高等教育出版社

作者：陈意云

页数：381

<<编译原理>>

前言

本书介绍编译器构造的一般原理、基本设计方法和主要实现技术，可作为高等学校计算机科学及相关专业的教材。

虽然只有少数人从事构造或维护程序设计语言编译器的工作，但是编译原理和技术对高校学生和计算机软件工程技术人员来说仍是重要的基础知识之一。

本书能使读者对程序设计语言的设计和实现有深刻的理解，对和程序设计语言有关的理论有所了解，对宏观上把握程序设计语言来说，能起一个奠基的作用。

本书的学习还有助于读者快速理解、定位和解决在程序调试与运行中出现的问题。

对软件工程来说，编译器是一个很好的实例（基本设计、模块划分、基于事件驱动的编程等），本书所介绍的概念和技术能应用到一般的软件设计之中。

大多数程序员同时也是语言的设计者，虽然只是一些简单语言（如输入输出、脚本语言）的设计者，但学习本书仍有助于提高他们设计这些语言的水平。

编译技术在软件安全、程序理解和软件逆向工程等方面有着广泛的应用。

作为教材，本书有如下一些特点：　　（1）在介绍语言实现技术的同时，强调一些相关的理论知识，如形式语言和自动机理论、语法制导的定义和属性文法、类型论和类型系统等。

它们是计算机专业理论知识的一个重要部分，在本书中结合应用来介绍这些知识，有助于学生较快领会和掌握。

（2）在介绍编译器各逻辑阶段的实现时，强调形式化描述技术，并以语法制导定义作为翻译的主要描述工具。

<<编译原理>>

内容概要

本书介绍编译器构造的一般原理和基本实现方法，主要内容包括词法分析、语法分析、语义分析、中间代码生成、代码优化和目标代码生成等。
除了介绍命令式编程语言的编译技术外，本书还介绍面向对象语言和函数式编程语言的实现技术。
本书还强调一些相关的理论知识，如形式语言和自动机理论、语法制导的定义和属性文法、类型论和类型系统等。
　本书取材广泛新颖、图文并茂，注意理论联系实际。
本书可作为高等学校计算机科学及相关专业的教材，也可供计算机软件工程技术人员参考使用。

<<编译原理>>

书籍目录

章节摘录

在语言设计的历史上，安全性考虑不足是出于效率上的原因，最典型的是C语言。
C以处理低级构造的灵活性著称，它的设计为了保证编程的灵活性而牺牲了安全性，它鼓励在安全的边缘进行编程。
这使得C的程序效率很高，但是也使得它们缺乏安全性。
为了排除程序中的安全漏洞，需要大量的调式工作。
对于安全性不足的语言，可以通过运行时的检查来提高安全性，但是运行检查的代价有时是非常昂贵的。
另外，即使是做了细致静态分析的语言，获得彻底的安全性也需要运行时的代价。
例如，一般来说，数组界检查是不可能完全在编译时删除的。
从另一个角度看，为获得安全性所花的代价是值得的。
安全性使得程序出现执行错误就会停止执行，这可以减少调试的时间。
而且安全性保证运行时结构的完整性，因而使得无用存储单元收集（garbagecollection，俗称垃圾收集，见第10章）可以完成。
而无用存储单元收集大大降低代码开发的时间和代码的规模，虽然它需要一点运行时的代价。
于是，从历史上看，安全和不安全语言之间的选择可能最终与开发时间和执行时间之间的权衡相关。
但是，随着经济和社会越来越依赖于信息技术，关键的信息基础构造——运输、通信、金融市场、能源分配和卫生保健等，都将非常危险地依赖于不是一个管理机构范围内的计算基础和资源。
在我们越来越依赖于这些信息基础构造的同时，提高系统对恶意攻击和有漏洞软件破坏的抵抗能力显得越来越重要。
例如，对于用C编写的系统来说，用缓冲区溢出来对它进行的攻击，已占目前各种攻击的50%，另外还有基于C程序格式串的安全漏洞而进行的攻击等等。
因此，当前在安全和不安全语言之间的选择还与对系统安全性的要求相关。

<<编译原理>>

版权说明