

<<编译原理及编译程序构造>>

图书基本信息

书名：<<编译原理及编译程序构造>>

13位ISBN编号：9787564140960

10位ISBN编号：7564140968

出版时间：2013-2

出版时间：东南大学出版社

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<编译原理及编译程序构造>>

内容概要

《普通高校计算机专业精品教材系列:编译原理及编译程序构造(第2版)》介绍编译理论基础及其实现方法,强调语言的形式化定义、编译技术的各种概念及实现过程的具体方法。

介绍过程以算法为核心,力求简单明了地反映编译的基础知识。

从形式语言理论角度讨论词法分析和语法分析技术,为计算机软件工作者开发大型软件打下良好基础。

《普通高校计算机专业精品教材系列:编译原理及编译程序构造(第2版)》以理论联系实际为宗旨,内容深入浅出,重点突出,并结合构造el语言的编译程序介绍一种常用而又简单的编译方法。

<<编译原理及编译程序构造>>

书籍目录

1引论 1.1程序设计语言与编译 1.2编译程序概述 1.2.1词法分析 1.2.2语法分析 1.2.3中间代码生成 1.2.4优化 1.2.5目标代码生成 1.2.6表格与表格管理 1.2.7出错处理 1.2.8遍 1.3编译程序生成 1.4编译程序构造 习题 2编译基础知识 2.1字母表与符号串 2.1.1符号串集合的运算 2.1.2符号串的前缀、后缀及子串 2.1.3字母表的闭包与正闭包 2.2文法与语言的关系 2.2.1文法的直观概念 2.2.2文法与语言的形式定义 2.3文法构造与文法简化 2.3.1由语言构造文法的例子 2.3.2文法的简化 2.3.3构造无E产生式的上下文无关文法 2.4语法树与文法的二义性 2.4.1语法树 2.4.2文法的二义性 习题 3词法分析 3.1正规文法和有限自动机 3.1.1正规文法、正规集与正规式 3.1.2有限自动机 3.1.3正规式与有限自动机之间的关系 3.1.4正规文法与有限自动机 3.2词法分析程序 3.2.1预处理与超前搜索 3.2.2扫描器的输出格式 3.2.3扫描器的设计 3.3词法分析程序的自动生成 3.3.1 LEX语言 3.3.2 LEX编译程序的构造 习题 4自上而下语法分析 4.1下推自动机 4.2自上而下分析法的一般问题 4.2.1消除左递归 4.2.2消除回溯——预测与提左因子 4.3预测分析程序与LL(1)文法 4.3.1求串a的终结首符集和非终结符A的随符集 4.3.2构造预测分析表 4.3.3状态表 4.4递归下降分析法 习题 5优先分析法 5.1简单优先分析方法 5.1.1基本思想 5.1.2有关文法的一些关系 5.1.3优先矩阵的构造算法 5.1.4简单优先分析算法 5.2算符优先分析法 5.2.1算符优先分析技术的引进 5.2.2算符优先文法及优先表的构造 5.2.3算符优先分析的若干问题 5.3优先函数 习题 6 LR分析法及分析程序自动构造 6.1 LR分析器 6.2 LR(0)项目集族和LR(0)分析表的构造 6.2.1 LR(0)项目集规范族的构造 6.2.2 LR(o)分析表的构造算法 6.3 SLR分析表的构造 6.4规范LR分析表的构造 6.4.1构造LR(1)项目集规范族的算法 6.4.2构造LR(1)分析表的算法 6.5 LALR分析表构造 6.5.1基本思想 6.5.2构造LALR分析表的算法 6.6二义文法的应用 6.7分析表的自动生成 6.7.1终结符和产生式的优先级 6.7.2结合规则 6.7.3 LR分析表的安排 习题 7语法制导翻译并产生中间代码 7.1概述 7.2简单算术表达式和赋值语句的翻译 7.2.1四元式 7.2.2赋值语句的翻译 7.2.3类型转换 7.3布尔表达式的翻译 7.3.1布尔表达式在逻辑演算中的翻译 7.3.2控制语句中布尔式的翻译 7.4控制语句的翻译 7.4.1标号和转移语句 7.4.2 IF语句的翻译 7.4.3 WHILE语句的翻译 7.4.4 REPEAT语句的翻译 7.4.5循环FOR语句的翻译 7.4.6分情语句的翻译 7.4.7复合语句的翻译 7.5数组元素及其在赋值语句中的翻译 7.5.1数组及其下标变量地址的计算 7.5.2数组元素引用的中间代码形式 8运行是数据区的管理 9代码优化 10目标代码生成 附录 参考文献

<<编译原理及编译程序构造>>

章节摘录

版权页：插图：这种技术除了可以减少搜索时间外，还有一个优点是把两小块合并为较大块比较容易实现。

其缺点是会使内部碎片增加，因为分配的块可能大于所申请的长度。

另外，外部碎片也可能增加。

因为，虽然两块是毗邻的，但由于它们不是双胞块，所以不能合并。

尽管这样，这种技术还是比前两种技术更为有效。

8.3.2堆空间的释放与无用单元收集 1) 堆空间的释放 在程序设计语言中有堆空间的释放语句，像Pascal中的dispose (P)，PL/1中的free等语句都是释放语句。

这些语句是将指针变量P所指的存储块退回到可利用表中。

最简单的处理办法是将释放块作为新块插入到可利用表的链首位置。

但这种回收策略有个缺点，就是在程序运行一段时间后，可利用表中将含有大量小块，分配程序的搜索时间将变得过分冗长，而且有不能满足用户申请要求的危险。

一种较好的解决办法是将两个连续的小块合并成一大块。

对新释放的块，按存储地址大小在可利用表中检查是否有与它是相邻关系的块，若有就将它们合并成一大块，否则仅按地址大小插入到适当位置。

为实现此操作，可利用表必须按块的地址顺序组织，以便搜索和插入。

在Pascal中，堆空间是全局量。

设P是某过程内的局部变量，执行new (P)以后，在堆中为P分配一块空间，如果在退出该过程前没有把P空间释放掉（这不算语法错或语义错），那么在退出该过程后P所指的空间既没有用但又不能再分配给其他用户，造成资源浪费，并经常导致死机。

为了克服这个问题，有些语言的编译程序提供堆管理程序，对无用单元进行收集。

2) 无用单元收集 (garbage collection) 无用单元收集程序一般是在堆的可利用空间几乎耗尽，以至于不能满足用户申请存储区要求，或者发现可利用空间已降至某个危险点时才执行。

无用单元收集过程通常分为两个阶段：第一阶段称为作标记阶段，即对已分配的块查阅这一段时间内是否被访问过，若被访问过就作一标记；第二阶段是收集阶段，即把所有未加标记的存储块加入到可利用表中，然后消除加过标记的那些存储块的标记，包括访问标记。

这种方法可以防止死块的产生，因为加上标记的块不会被释放（表示它还在用），而没有加标记的块都会被释放到可利用表中。

这种无用单元收集技术存在一个缺点：它的开销（主要指收集时间）随可利用空间的下降而增加。

解决这个问题的途径是在可利用空间降至某个数值时就调用收集程序，使得可利用空间总保持在较佳的状态。

另外，在执行无用单元收集程序时必须中断用户程序的执行，要等到收集程序执行结束后再执行用户程序。

所以，若频繁地执行收集程序，必然影响计算机的运算速度。

因此，选择执行收集程序的时刻必须要合理。

上述的无用单元的收集只能收集死块单元（包括长期不用单元）和外部碎片，不能收集内部碎片。

解决这个问题的方法是在收集的第一阶段，标记的对象不是以存储块为单位而是以存储单元为单位。

第二阶段是既实行收集又进行内存大搬家，对每一使用块重新定位，将已用块归于一端，腾出另一端为可利用空间。

这种操作可以收集所有无用单元。

<<编译原理及编译程序构造>>

编辑推荐

《普通高校计算机专业精品教材系列:编译原理及编译程序构造(第2版)》可作为高等院校计算机专业的本科生或专科教材,也可作为硕士生入学考试及计算机软件技术人员的参考书。

<<编译原理及编译程序构造>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>