

编译原理

北方工业大学信息学院
School of Information Science and Technology,
North China University of Technology
束劼
shujie@ncut.edu.cn
瀚学楼1122, 88801615

第八章 符号表

第八章 符号表

- 本章目录
- 8.1 符号表的组织与作用
- 8.2 整理与查找
- 8.3 名字的作用范围
- 8.4 符号表的内容

3

第八章 符号表

- 大纲要求
- 掌握：符号表的组织与作用，名字的作用范围。
- 理解：符号表的内容。
- 了解：符号表管理的基本方法。

4

8.1 符号表的组织与作用

8.1 符号表的组织与作用

- 符号表

包含**两部分**：一部分是名字(标识符)，另一部分是此名字的有关信息。

有关信息包含种属、类型等，用于语义检查、产生中间代码以及最终生成目标代码等不同阶段。

8.1.1 符号表的作用

8.1.1 符号表的作用

- 符号表的作用

每当遇到一个名字都要查符号表。发现新名字，或者发现已有名字的新信息，则要修改符号表，填入新名字和新信息。符号表包含**两大栏**：名字栏(主栏)和信息栏。

	名字栏(NAME)	信息栏(INFORMATION)
第1项	关键字1	
第2项	关键字2	
...		
第n项	关键字n	

8.1.1 符号表的作用

- 符号表的作用

对符号表的操作可分为5类(增、删、2查、改):

- ① 对给定名字, **查询**此名是否已在表中 **查**
- ② 往表中**填入**一个新的名字 **增**
- ③ 对给定名字, **访问**它的某些信息 **查**
- ④ 对给定名字, 往表中**填写或更新**它的某些信息 **改**
- ⑤ **删除**一个或一组无用的项 **删**

9

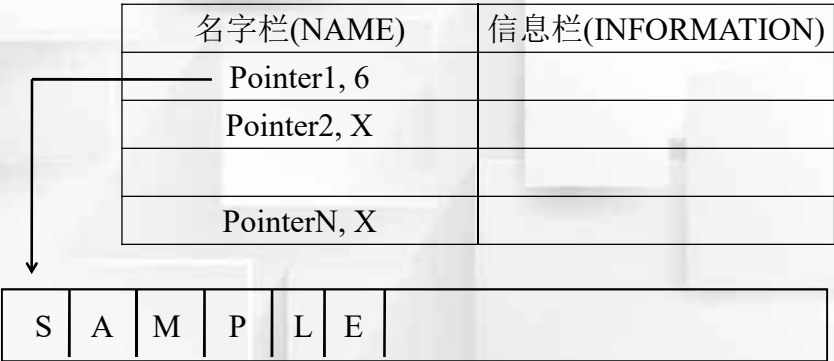
8.1.2 符号表的组织方式

8.1.2 符号表的组织方式

- 符号表的组织方式
 - 最简单的**组织方式：让各项各栏所占的存储单元的长度都是固定的。例如，标识符长度规定不超8个字符，不到8字符的用空白补足。
 - 其他**组织方式：如果对标识符的长度不加限制，则需要为所有标识符建立一个独立的数组，在符号表的主栏中放一个指示器和一个整数，或仅放一个指示器，在数组中的标识符前放一个整数。这里的指示器是标识符在数组中的位置，整数是此标识符的长度。

8.1.2 符号表的组织方式

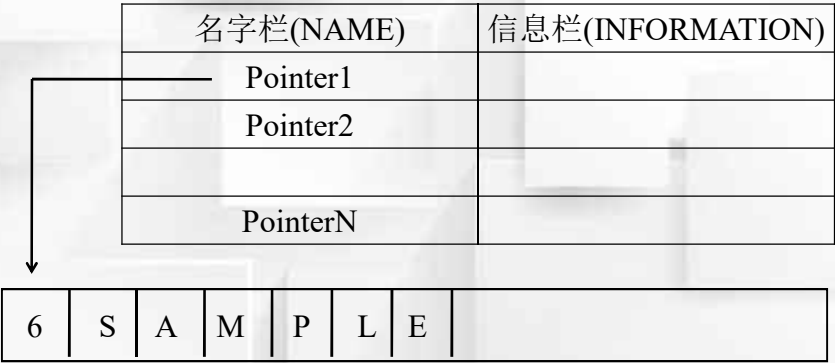
- 符号表的组织方式
 - 其他组织方式的示意图一



8.1.2 符号表的组织方式

- 符号表的组织方式

其他组织方式的示意图二



8.1.2 符号表的组织方式

- 符号表的组织方式

对信息栏的组织方式：类似对主栏的组织方式，另外，可以把某些特殊属性登记在别的地方，并在信息栏附设一个指示器，指向存放特殊属性的地方。

例如，对数组维度的存储，一般会专门开辟一个信息表区，称为数组信息表(内情向量表)。与数组相关的信息全部存入此表。

8.1.2 符号表的组织方式

- 符号表的存储方式

一张可容纳N项的符号表在存储器中，可用下列两种形式存储(每项K个字)：

- ① 把每一项置于连续的存储单元中(K个)，从而得到 $K \times N$ 个字的表。按符号表行顺序存储。
- 把整个符号表分成M个子表，每个子表含N项，如果某个子表 T_i 中某一项的字数为 K_i ，则符号表中的某一项的字数为 $\sum K_i$ 。后续举例说明。

通常使用第2种存储形式，以便合理使用存储空间

15

8.2 整理与查找

8.2.1 线性表

8.2.1 线性表

- 线性表
- 线性表中每一项的先后顺序是按**先来者先填**的顺序安排的，编译程序**不做任何整理次序的工作**。指针 **AVAILABLE** 总是指向空白区的首地址，即线性表中已填信息最后一项的下一项开始。当需要查某个名字时，就从该表的第一项开始顺序查找，若一直查到 **AVAILABLE** 还未找到这个名字，则说明这个名字不在表中。
- 往往使用**反序查找**，因为程序员习惯对新定义的名字立即使用。

8.2.1 线性表

- 线性表
- **查找时间：**对一张含 n 项的线性表来说，平均需要做 $n/2$ 次的比较，效率较低。
- 为了提高效率，给每项附加一个指示器，该指示器总是按“最新最近”访问原则把所有项连接成一条链。该链的第一个元素时最新最近被查询过的项，第二个元素是次新次近被查询过的项，等等。
- 含有这种链的线性表叫做**自适应线性表**。

19

8.2.2 对折查找与二叉树

8.2.2 对折查找与二叉树

- 对折查找与二叉树

这是提高查表速度的方法之一。

- ① 首先，把各项按名字的大小顺序整理排列。名字的大小为名字的内码二进制值。值小的在前面，值大的在后面。
- ② 其次，使用**对折法**查找。即把SYM和中项比较，若相等，则查到。若小于，则在前半段中查找。若大于，则在后半段中查找。在前、后半段查找时，也使用对折法比较查找。

21

8.2.2 对折查找与二叉树

- 对折查找与二叉树

查找时间：最多只须作 $1+\log_2 N$ 次比较，也叫对数查找法

问题：符号表是变动的，每添加一个新的项，需要重新排列，极费时间。

解决方法：把符号表组织成一颗二叉树。令每项为一个结点，每个结点带两个指示器栏，即左枝与右枝。要求任何结点p的右枝结点的值要小于结点的p的值，左枝的值则要大于结点p的值。

22

8.2.2 对折查找与二叉树

- 对折查找与二叉树

二叉树查找时间：与 $\log_2 N$ 成比例。

二叉树优势：大量减少排序的时间

二叉树劣势：效率低于对折查找法，且存储空间比对折查找法消耗更大。

23

8.2.3 杂凑技术

8.2.3 杂凑技术

- 杂凑技术

线性表的优劣势：填表快，查表慢

对折法的优劣势：填表慢，查表快

杂凑技术：综合两者优势的一种方法

25

8.2.3 杂凑技术

- 杂凑技术

一张含N项的符号表，构造一个地址函数H，对任何读入SYM中的名字， $H(SYM)$ 的值在0到N-1之间。例如，用 SYM/N 取余的函数作为地址函数H，则名字09置于表中第9项，名字34置于表中第0项，等等。

26

8.2.3 杂凑技术

- 杂凑技术

对地址函数H的要求有两点：

- ① 计算要简单、高效；
- ② 函数值能比较均匀的分布在0到N-1之间。例如，取N为质数， $H(\text{SYM}) = \text{mod}(\text{SYM}/N)$ 就是一个比较理想的函数。

其他要求，解决**地址冲突**的问题。杂凑技术通常使用一张**杂凑(链)表**通过间接方式查填符号表。所有相同杂凑值的符号名连成一串，便于线性查找。因此，符号表中多添加一栏写入链接地址。

27

8.3 名字的作用范围

8.3 名字的作用范围

- 在FORTRAN中，变量、数组、语句函数的名字的作用范围是它们所处的程序段。

因此，同一个标识符，具有不同性质，要求分配不同的存储空间。

通常使用**最近嵌套作用域**规则，对每个过程指定一个唯一的编号，该编号看成是名字的一个组成部分。在符号表中，局部名字用一个二元组：<名字，过程编号>。编号可以通过识别语义规则，用语法制导翻译的方法实现。

29

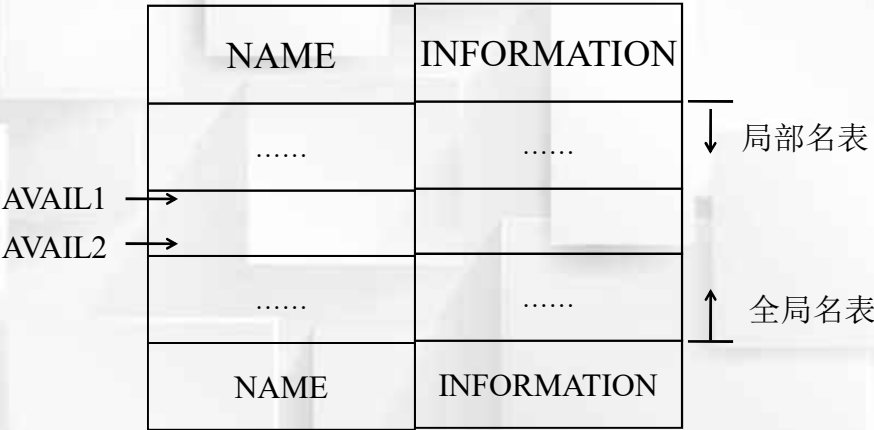
8.3.1 FORTRAN的符号表组织

8.3.1 FORTRAN的符号表组织

- FORTRAN中由一个主程序段和若干过程段组成。
- 对一遍扫描的编译程序，可以逐段产生其目标代码。当一段处理完毕，它所有的局部名均无须保存在符号表中，需要保留在符号表中的只有全局名。
- 局部名表区是一个可重复使用的区域。

8.3.1 FORTRAN的符号表组织

当现行局部段处理完毕，AVAIL1重置，指向局部名区的第一项

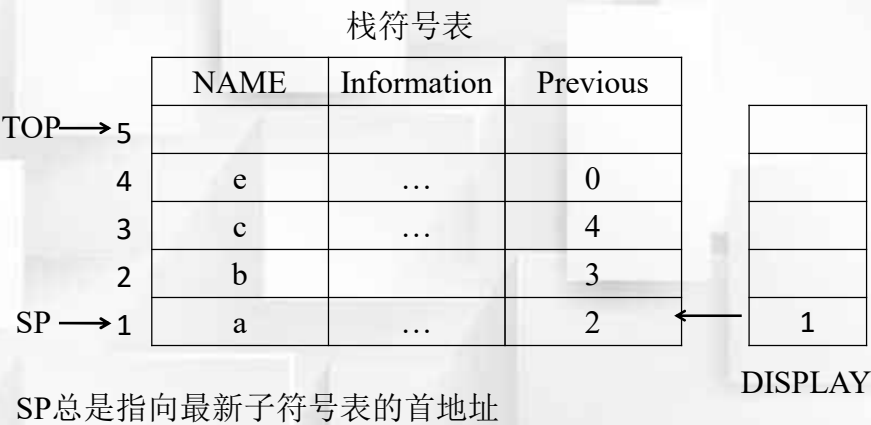


8.3.2 Pascal的符号表组织

8.3.2 Pascal的符号表组织

- Pascal的特点是具有过程结构。在一个过程中被说明了的名字，被认为是局部于该过程的。
- Pascal的过程结构是嵌套的，内层过程可以引用外层过程中说明过的名字。因此，每进入一层，要为该层建立一张子符号表。查表时，先从子符号表查起，逐层往外查询。
- 另外需要建立一个显示(DISPLAY)层次关系表，称为过程的嵌套层次表。作用是为了描述过程的嵌套层次，指出当前正在活动着的各嵌套的过程相应的子符号表，在栈符号表中的起始位置。

8.3.2 Pascal的符号表组织



8.4 符号表的内容

8.4 符号表的内容

- 变量名、数组名和过程名的信息栏中的信息

变量、数组：类型、种属、长度、相对数、内情向量(数组)、联系记录结构与其分量的方式、形式参数标志等等。

过程：是否为程序的外部过程?若为函数，类型是什么?其说明是否处理过?是否递归?等等

37

8.4 符号表的内容

- 对于使用单一符号表的简单语言，符号表填入新项的工作可以由词法分析程序来完成。
- 对于某些复杂的语言，在同一过程段里允许同一标识符识别各种不同对像。要采用多种符号表，并让语法-语义分析程序负责查填工作。

38

第八章 小结

第八章 小结

- 8.1 符号表的组织与作用
- 8.2 整理与查找
- 8.3 名字的作用范围
- 8.4 符号表的内容

Coursework

- 8.1 什么是符号表？符号表有哪些重要作用？
- 8.2 符号表的组织方式有哪些？它的组织取决于哪些因素？