



山东大学  
SHANDONG UNIVERSITY

编译原理

# 第八章 符号表

授 课 教 师 : 余仲星  
手 机 : 15866821709 (微信同号)  
邮 箱 : zhongxing.yu@sdu.edu.cn

## 第八章 符号表

- 8.1 符号表的作用和地位
- 8.2 符号表的主要属性及作用
- 8.3 符号表的组织
  - 8.3.1 符号表的总体组织
  - 8.3.2 符号表项的排列
  - 8.3.3 关键字域的组织

# 第八章 符号表

## □ 8.1 符号表的作用和地位

## □ 8.2 符号表的主要属性及作用

## □ 8.3 符号表的组织

- 8.3.1 符号表的总体组织
- 8.3.2 符号表项的排列
- 8.3.3 关键字域的组织

# 8.1 符号表的作用和地位

## □ 符号表的作用

- 收集符号属性;
- 上下文语义的合法性检查的依据, 如变量重复定义、标号检查等;
- 作为目标代码生成阶段地址分配的依据。

## □ 对符号表的操作

- 对给定名字, 查询此名是否已在表中;
- 往表中填入一个新的名字;
- 对给定名字, 访问它的某些信息;
- 对给定名字, 往表中填写或更新它的某些信息;
- 删除一个或一组无用的项。

# 第八章 符号表

## □ 8.1 符号表的作用和地位

## □ 8.2 符号表的主要属性及作用

## □ 8.3 符号表的组织

- 8.3.1 符号表的总体组织
- 8.3.2 符号表项的排列
- 8.3.3 关键字域的组织

## 8.2 符号表的主要属性及作用

### □ 符号名

- 标识符可以是变量、函数、过程、类的名字，**一般不允许重名**；
- 在一些允许**重载**的语言中，函数名、过程名是**可以重名**的，需要通过参数个数和类型进行区分。

### □ 符号的数据类型

- **基本类型**，如整型、实型、字符型、布尔型、位组型等；
- **扩充类型**，如数组类型、记录结构类型、对象类型等。

## 8.2 符号表的主要属性及作用

❑ **符号的存储类别**：是语义处理、检查和存储分配的重要依据，还决定了符号变量的作用域、可见性、生命周期等问题。

### ➤ 关键字指定

- ✓ Fortran中用COMMON定义公共存储区变量，SAVE定义函数或过程的内部静态存储变量；
- ✓ C语言用static定义属于文件或函数内部的静态存储变量。

### ➤ 根据位置确定

- ✓ C语言函数体外默认为公共存储变量，函数体内默认为私有存储变量。

## 8.2 符号表的主要属性及作用

### □ 符号的作用域及可见性

- 一般来说，定义该符号的位置及存储类关键字决定了该符号的作用域。
- C语言全局与局部的同名变量

```
int a;  
  
int func()  
{  
    float a;  
    ...a...    // 引用float a  
    ....a...   // 引用int a  
}
```

- 分程序结构，即过程/函数的嵌套定义，某层可以看到的变量是在本层定义或各外层中最内层定义的该变量。



## 8.2 符号表的主要属性及作用

### □ 符号的存储分配信息

- 静态存储区：如Fortran的Save语句和C语言的static语句定义的变量；
- 动态存储区：根据变量的局部定义和分程序结构，设置动态存储区来适应这些局部变量的生存和消亡。

## 第八章 符号表

- 8.1 符号表的作用和地位
- 8.2 符号表的主要属性及作用
- 8.3 符号表的组织
  - 8.3.1 符号表的总体组织
  - 8.3.2 符号表项的排列
  - 8.3.3 关键字域的组织

## 8.3.1 符号表的总体组织

### □ 主要问题

- 不同种类的符号，属性信息有差异。

### □ 第1种组织方式：构造多个符号表，具有相同属性种类的符号组织在一起

- 优点：每个符号表中存放符号的属性个数和结构完全相同；
- 缺点：一遍编译程序同时管理若干个符号表。

符号	属性1	属性2	属性3

符号	属性1	属性2	属性4

符号	属性2	属性3	属性5	属性6	属性7	属性8

## 8.3.1 符号表的总体组织

□ 第2种组织方式：把所有符号都组织在一张符号表中

- 优点：管理集中单一；
- 缺点：增加了空间开销。

符号	属性1	属性2	属性3	属性4	属性5	属性6

## 8.3.1 符号表的总体组织

□ **第3种组织方式**：根据符号属性相似程度分类组织成若干张表

- **优点**：减少了空间开销；
- **缺点**：增加了表格管理的复杂性。

符号	属性1	属性2	属性3	属性4

第1、2种符号

符号	属性2	属性3	属性5	属性6	属性7	属性8

第3种符号

## 第八章 符号表

- 8.1 符号表的作用和地位
- 8.2 符号表的主要属性及作用
- 8.3 符号表的组织
  - 8.3.1 符号表的总体组织
  - 8.3.2 符号表项的排列
  - 8.3.3 关键字域的组织

## 8.3.2 符号表项的排列

### □ 1、线性组织

- 优点：插入快，空间效率高；
- 缺点：查询慢，时间效率差。

....a.....  
.....b....  
....f.....  
....e.....  
.....d....  
...c.....

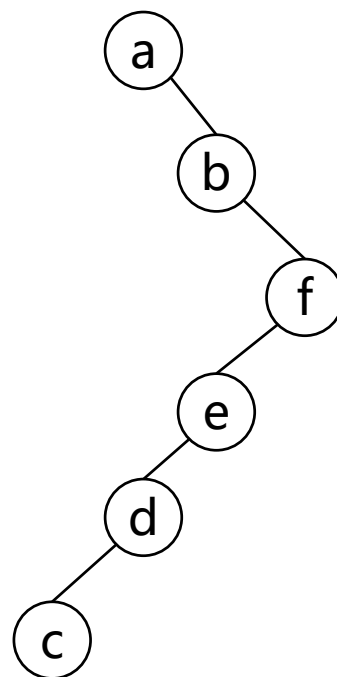
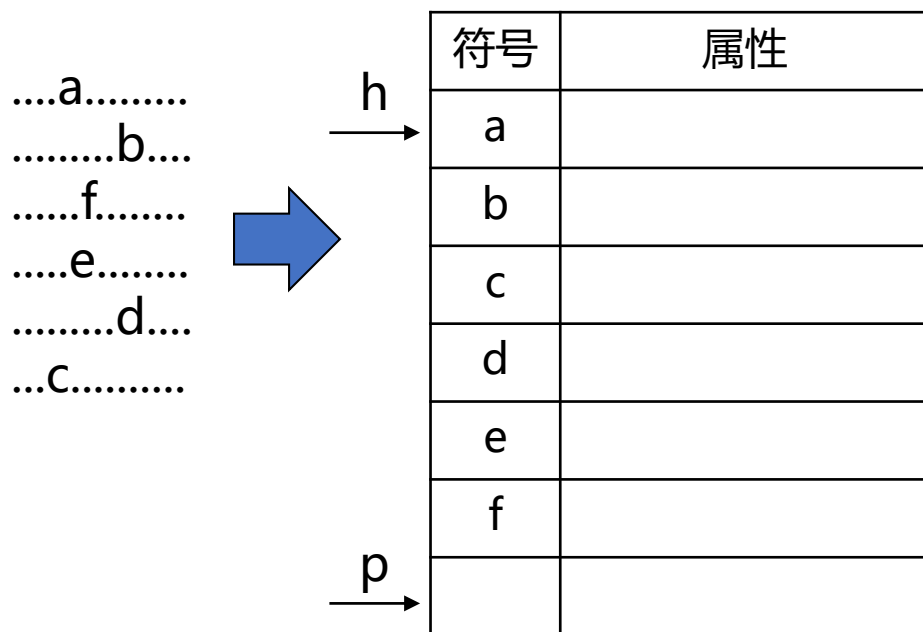


	符号	属性
h →	a	
	b	
	f	
	e	
	d	
	c	
p →		

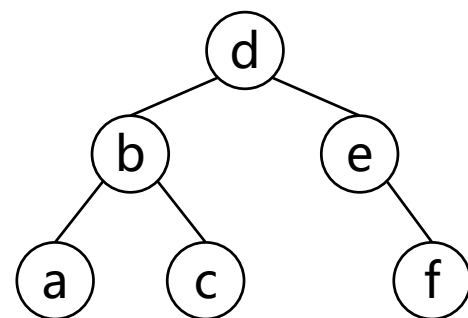
## 8.3.2 符号表项的排列

### □ 2、排序组织及二分法

- 优点：查询效率高，空间效率高；
- 缺点：插入效率低，算法复杂一些。



二叉树



平衡二叉树



## 8.3.2 符号表项的排列

### □ 3、Hash表

- 优点：插入、查询效率都高；
- 缺点：空间效率有所降低。

■ 直接定址法：  $H(\text{key}) = (a * \text{key} + b) \% m$ ，其中  $m$  是哈希表的长度。

– 例：  $H(\text{key}) = \text{key} \% m$ ，其中  $m = 10$ 。

1, 30, 34

0	1	2	3	4	5	6	7	8	9
	1								
30	1								
30	1			34					

## 8.3.2 符号表项的排列

- **直接定址法**:  $H(\text{key}) = (a * \text{key} + b) \% m$ , 其中 $m$ 是哈希表的长度。
  - 例:  $H(\text{key}) = \text{key} \% m$ .
- **数字分析法**: 取中间某些有区分度的数字。
  - 例: 身份证作为 $\text{key}$ , 同一个地区的可以取生日开始的8+3位。
- **平方取中法**: 如果关键字的每一位都有某些数字重复出现频率很高的现象, 可以先求关键字的平方值以扩大差异, 取中间数位作为最终存储地址。
  - 例:  $\text{key}=1234$   $1234^2=1522756$  取2275作hash地址
  - $\text{key}=4321$   $4321^2=18671041$  取6710作hash地址。
- **数字折叠法**: 如果数字的位数很多, 可以将数字分割为几个部分, 取他们的叠加和作为hash地址。
  - 例:  $\text{key}=123\ 456\ 789$ , 折叠 $(123 + 456 + 789) \% 1000 = 491$ 。
- **除留余数法**:  $H(\text{key}) = \text{key} \% p$  ( $p \leq m$ ,  $m$ 为表长)

## 8.3.2 符号表项的排列

■ **直接定址法**:  $H(\text{key}) = (a * \text{key} + b) \% m$ , 其中 $m$ 是哈希表的长度。

– 例:  $H(\text{key}) = \text{key} \% m$ .

Key: 1, 30, 34, 50, 77, 60, 44, 37

0	1	2	3	4	5	6	7	8	9
	1								
30	1								
30	1			34					

## 8.3.2 符号表项的排列

■ **开放定址法解决哈希冲突**：如果 $H(\text{key}_i) = H(\text{key}_j)$ ，则 $H_i = [H(\text{key}) + d_i] \% m$ ，其中 $d_i$ 有三种取法：

- 线性探测再散列： $d_i = c * i$
- 平方探测再散列： $d_i = 1^2, -1^2, 2^2, -2^2, \dots$
- 随机探测在散列（双探测再散列）： $d_i$ 是一组伪随机数列

$H(\text{key}) = \text{key} \% m$ ，其中 $m = 10$ ，取 $d_i = i$

Key: 1, 30, 34, 50, 77, 60, 44, 37

0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9
										1	30	1	50		34					
	1									2	30	1	50		34			77		
30	1									3	30	1	50	60	34			77		
30	1			34						4	30	1	50	60	34	44		77		
30	1	50		34						5	30	1	50	60	34	44		77	37	

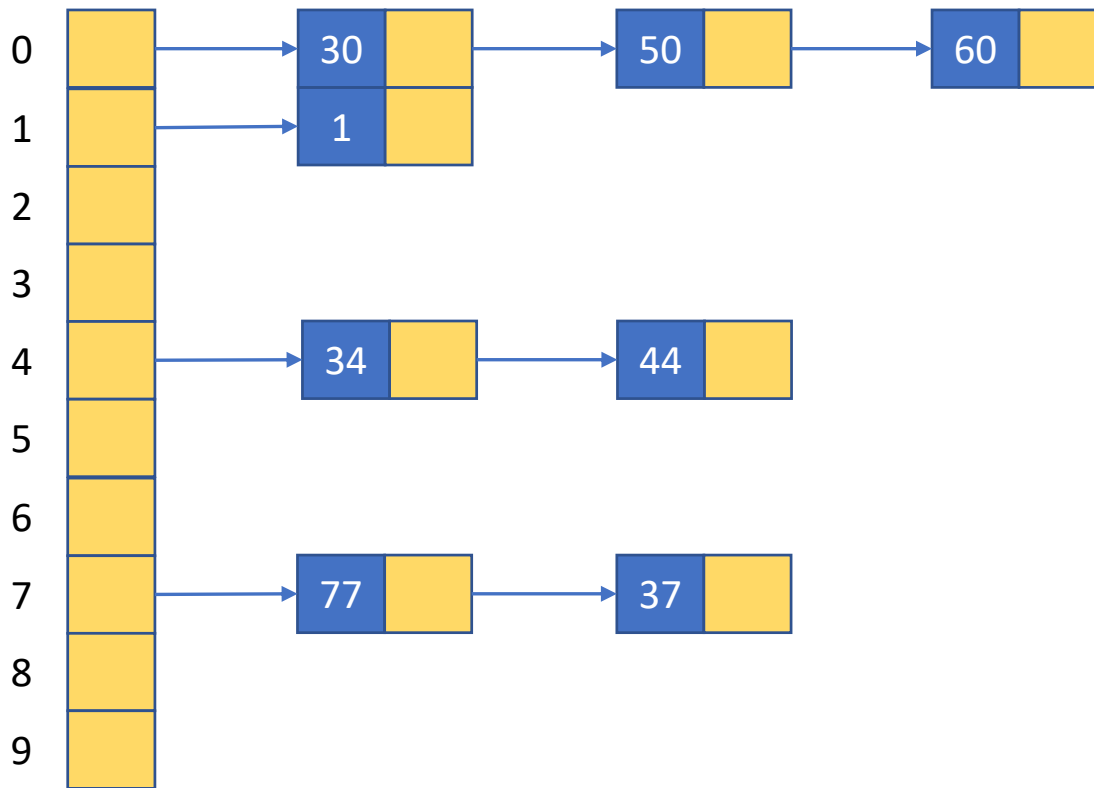
## 8.3.2 符号表项的排列

- **开放定址法**：如果 $H(\text{key}_i) = H(\text{key}_j)$ , 则 $H_i = [H(\text{key}) + d_i] \% m$ , 其中 $d_i$ 有三种取法：
  - 线性探测再散列：  $d_i = c * i$
  - 平方探测再散列：  $d_i = 1^2, -1^2, 2^2, -2^2, \dots$
  - 随机探测在散列（双探测再散列）：  $d_i$ 是一组伪随机数列
- **再哈希法**：如果 $H_1(\text{key}_i) = H_1(\text{key}_j)$ , 则使用 $H_2(\text{key}_i) = H_2(\text{key}_j)$ , 如果还冲突, 再使用 $H_3(\text{key}_i) = H_3(\text{key}_j), \dots$
- **链地址法**：将所有关键字为同义词的记录存储在同一线性链表中。

## 8.3.2 符号表项的排列

■ **链地址法**：将所有关键字为同义词的记录存储在同一线性链表中。

Key: 1, 30, 34, 50, 77, 60, 44, 37



## 第八章 符号表

- 8.1 符号表的作用和地位
- 8.2 符号表的主要属性及作用
- 8.3 符号表的组织
  - 8.3.1 符号表的总体组织
  - 8.3.2 符号表项的排列
  - 8.3.3 关键字域的组织

## 8.3.3 关键字域的组织

■ 在编译程序中，符号表的**关键字域就是符号本身**。

➤ 如有如下标识符

an

exemplar

of

key\_word

field

