

第8章 (Part 1) 符号表

- 1 符号表的作用和地位
- 2 符号的主要属性及作用
- 3 符号表的组织
- 4 符号表的管理

1 符号表的作用和地位

在编译程序中符号表用来存放语言程序中出现的有关**标识符的属性信息**，这些信息集中反映了标识符的**语义特征**属性。

在词法分析及语法在分析过程中**不断积累和更新**表中的信息，并在词法分析到代码生成的各阶段，**按各自的需要从表中获取**不同的属性信息。

收集符号属性 编译程序扫描说明部分收集有关标识符的属性，并在符号表中建立符号的相应属性信息。

上下文语义合法性检查的依据 同一个标识符可能在程序的不同地方出现，而有关该符号的属性是在这些不同情况下收集的。通过符号表中属性记录可进行相应上下文的语义检查。

作为目标代码生成阶段地址分配的依据 每个符号变量在目标代码生成时需要确定其在存储分配的位置。首先要确定其被分配的区域；其次是根据变量出现的次序，决定该变量在某个区中所处的具体位置。

2 符号的主要属性及作用

- ① 符号名
- ② 符号的类型
- ③ 符号的存储类别
- ④ 符号的作用域及可视性
- ⑤ 符号变量的存储分配信息
- ⑥ 符号的其它属性

① 符号名

可以是一个变量、函数或过程的名字，是唯一标示。

② 符号的类型

③ 符号的存储类别

一种是用关键字指定，如static，register；

另一种方式是根据定义变量说明在程序中的位置来决定，如外部变量、内部变量。

④ 符号的作用域及可视性

一般来说一个变量的作用域就是该变量可以出现的场合，也就是说在某个变量作用域范围内该变量是可引用的，这就是变量可视性的作用域规则。

⑤ 符号变量的存储分配信息

一个编译程序有两类存储区，即静态存储区和动态存储区。

⑥ 符号的其它属性

- (1) 数组内情向量
- (2) 记录结构型的成员信息
- (3) 函数及过程的形参

在整个编译期间，对于符号表的操作大致可归纳为五类：

- 对给定名字，**查询**名字是否已在表中；
- 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，**填写或更新**它的某些信息；
- **删除**一个或一组无用的项。

不同种类的表格所涉及的操作往往也是不同的

3 符号表的组织

总体组织：

第一种： 把属性种类完全相同的那些符号组织在一起。

第二种： 把所有语言中的符号都组织在一张符号表中。

第三种折衷方式是根据符号属性相似程度分类组织成若干张表，每张表中记录的符号都有比较多的相同属性。

假设有下列三类符号及其所需之属性：

第一类符号	属性1	属性2	属性3
-------	-----	-----	-----

第二类符号	属性1	属性2	属性4
-------	-----	-----	-----

第三类符号	属性2	属性5	属性6
-------	-----	-----	-----

方法1：按属性分类组织的符号表

符号	属性值1	属性值2	属性值3

第一类符号之符号表

符号	属性值1	属性值2	属性值4

第二类符号之符号表

符号	属性值2	属性值5	属性值6

第三类符号之符号表

方法2：单一组织的符号表

符号	属性值1	属性值2	属性值3	属性值4	属性值5	属性值6

方法3：折衷方式组织的符号表

符号	属性值1	属性值2	属性值3	属性值4

第一，二类
符号之符号表

符号	属性值2	属性值5	属性值6

三类符号符号表

方法3的改进：属性复合组织的符号表

符号	属性值1	属性值2	属性值34

第一，二类
符号之符号表

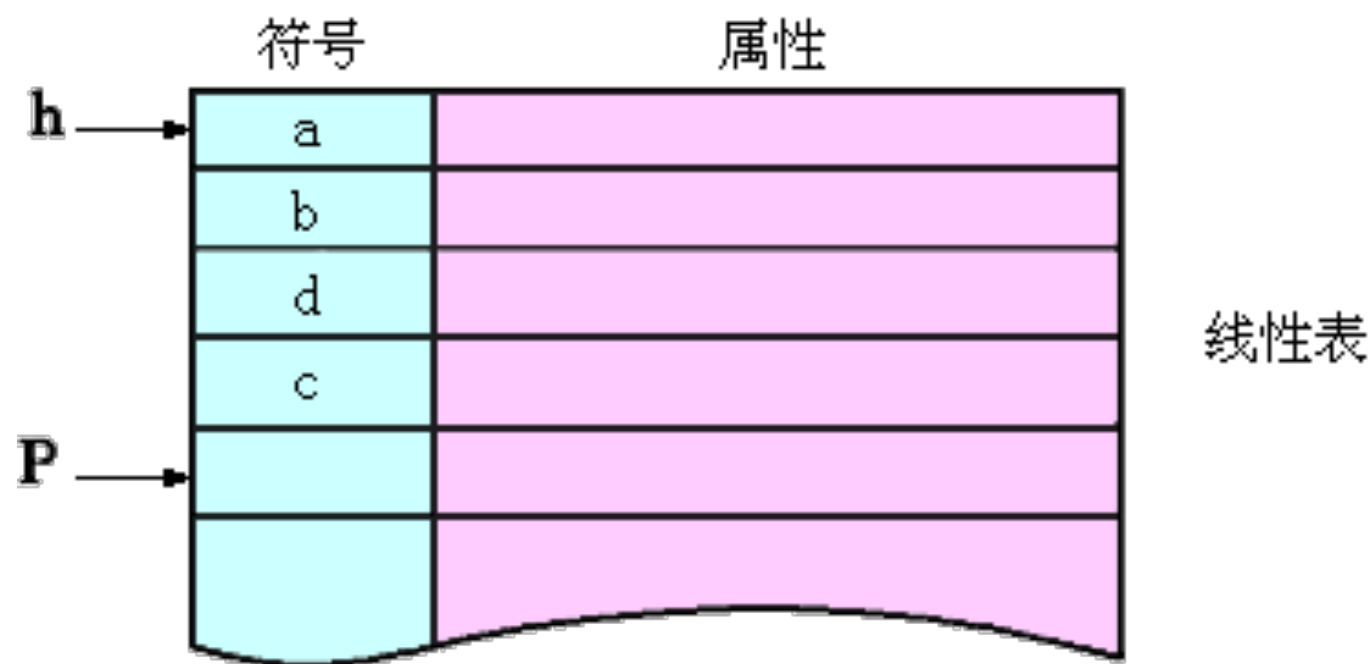
符号表项的排列

① **线性组织** 这种方法规定符号表中表项按它的符号被扫描到的先后顺序登录。

例如有一程序中出现符号的情况如下：

```
.....  
...a..... //第一次出现a的地方  
.....b..... //第一次出现b的地方  
...a..... //第二次出现a的地方  
.....d..... //第一次出现d的地方  
...c..... //第一次出现c的地方  
.....b..... //第二次出现b的地方  
.....
```

线性组织的符号表



[Forward](#)

② 排序组织及二分法

语言中的每一个符号，都是由一个或几个ASCII或EBCDIC代码字符拼写而成的，每一个符号在机器内都是由这种字符代码串来表示。排序组织的符号表，就是在符号表中的表项按其符号的字符代码串（可以看成是一个整数值）的值的大小从大到小（或从小到大）排列的。

排序组织的符号表

	符号	属性
h →	a	
	b	
	c	
	d	
P →		

排序表

③ 散列组织

一个符号在散列表中的位置，是由对该符号（即字符代码串）进行某种函数操作（通常称为“杂凑函数”）所得到的函数值来确定的。所得到的函数值与该表项在表中位置的对应关系，是通过对函数值的“求整”以及相对于表长的“求余”得到的。

假设选定的杂凑函数fhash，对符号代码值杂凑运算之后得到杂凑值是Vhash，可表示为：

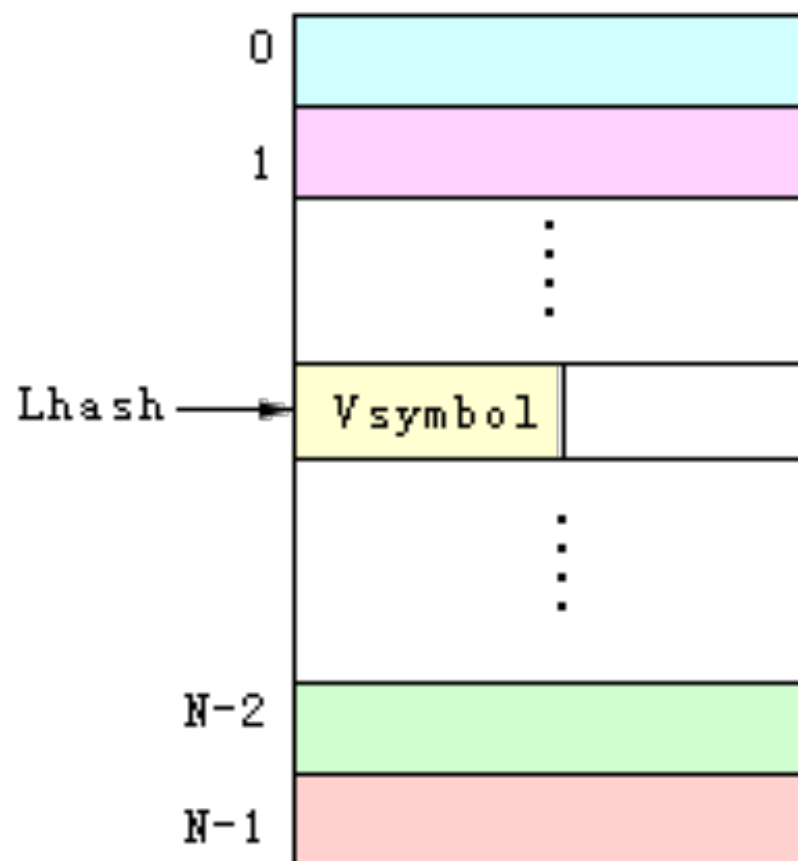
$$Vhash = fhash(\text{< 符号代码值 >})$$

散列表的表长通常是一个定值N，设符号的散列位置为Lhash，则有

$$Lhash = \text{mod}(Vhash, N)$$

散列组织的符号表

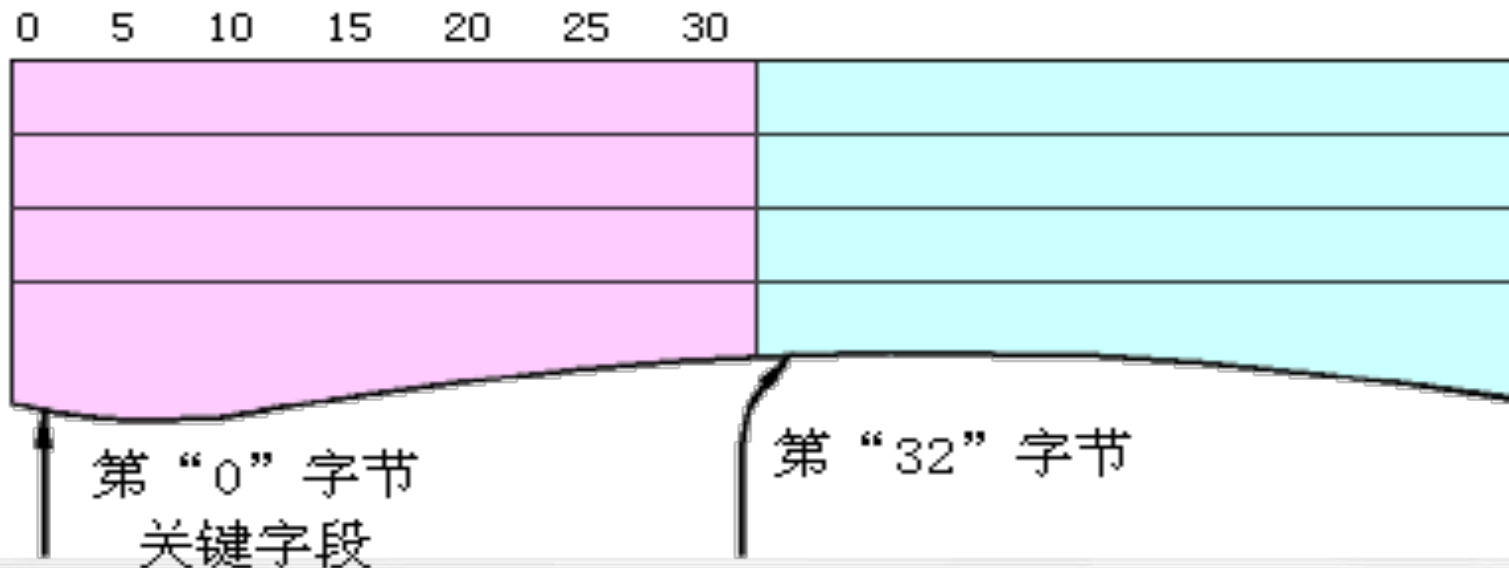
符号表 (表长 v)



关键字域的组织

符号表的关键字域就是符号本身，它可以是语言的保留字，操作符号或标识符（包括变量名，函数名，记录结构标志等）。

如C语言的关键字段长度可以是32个（其中31个是存放名字，余一个是存放字符串结束标志“□”，图示如下。

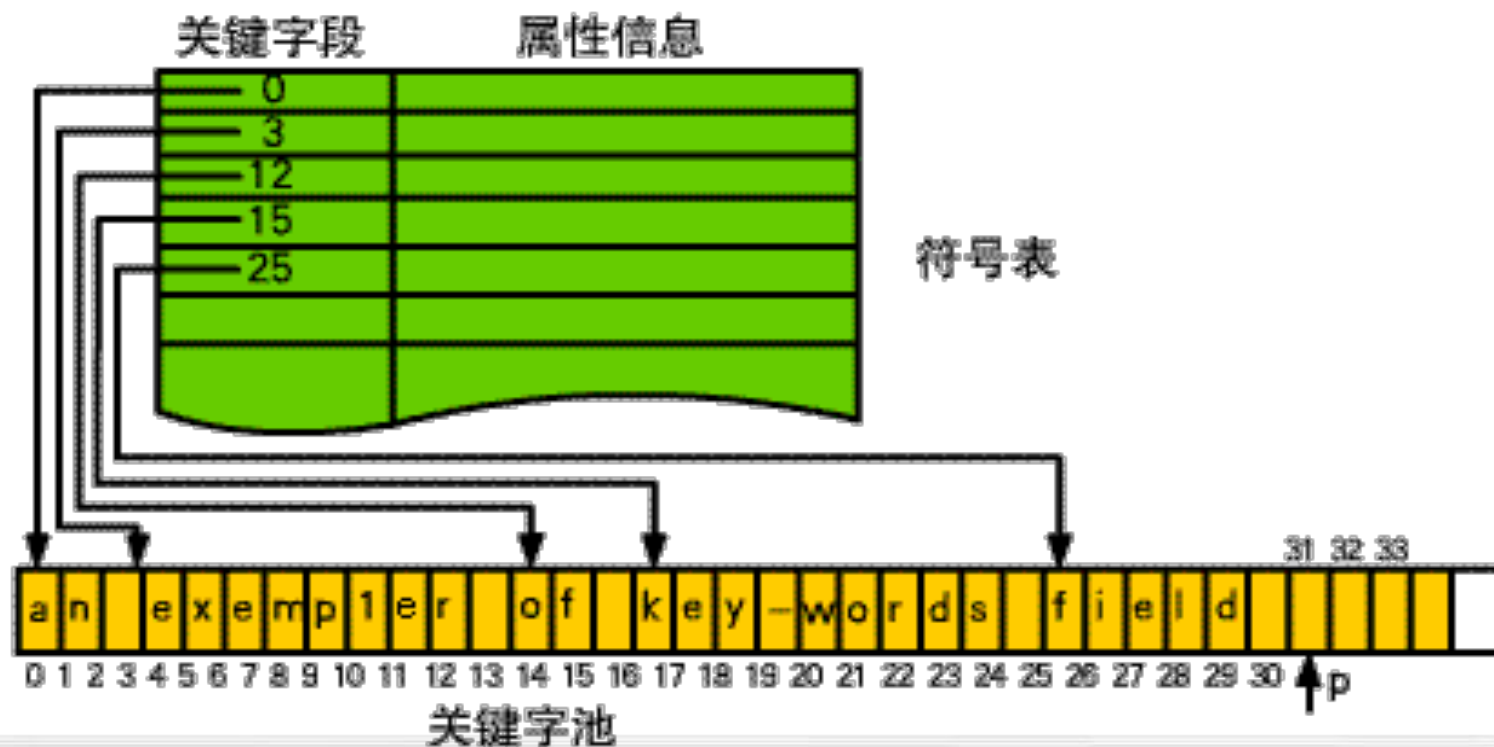


例如，有一组标识符：

关键字池

an
exemplar
of
key-wrds
field

关键字段的组织结构如下图（图9.10）。



其它域的组织

1、等长属性值域组织

可以取相应的数据类型描述属性值。

如，基本数据类型可用3个bit表示：

char	000
short	001
int	010
long	011
unsigned	100
float	101
double	110

函数形参链属性描述，见p217，图9.11。

结构成员链属性描述，见p218，图9.12。

其它域的组织

2、不等长属性值域的组织

对于不等长属性值的属性域，一般不把所有属性值都存放在符号表项的某个域内，而另设相关空间存放属性值。

如，数组内情向量属性值，见p219，图9.13。
不等长结构索引组织的符号表，见p222，图9.15。

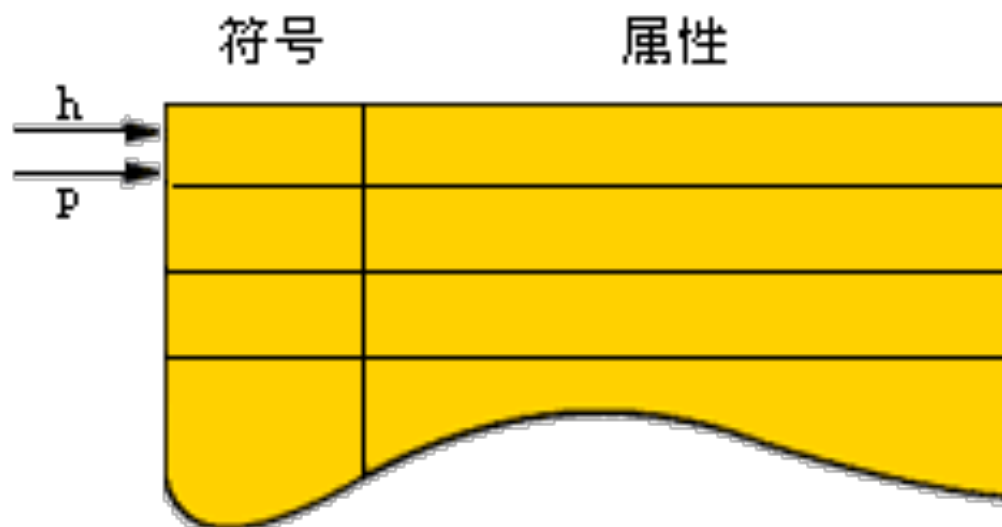
下推链域的组织

下推链域组织要求在进入一个内层结构并发生重名标识符定义时，需把当前符号表中外层的该符号表项下推到下推链中而在符号表被下推的表项处建立内层的同名标识符的表项。见p223，图9.18。

4 符号表的管理

1 符号表的初始化

① 符号表的表长是渐增变化的情况



渐增符号表初态

[lookback](#)

② 符号表的表长是确定的情况

定长符号表初态

定长符号表初态

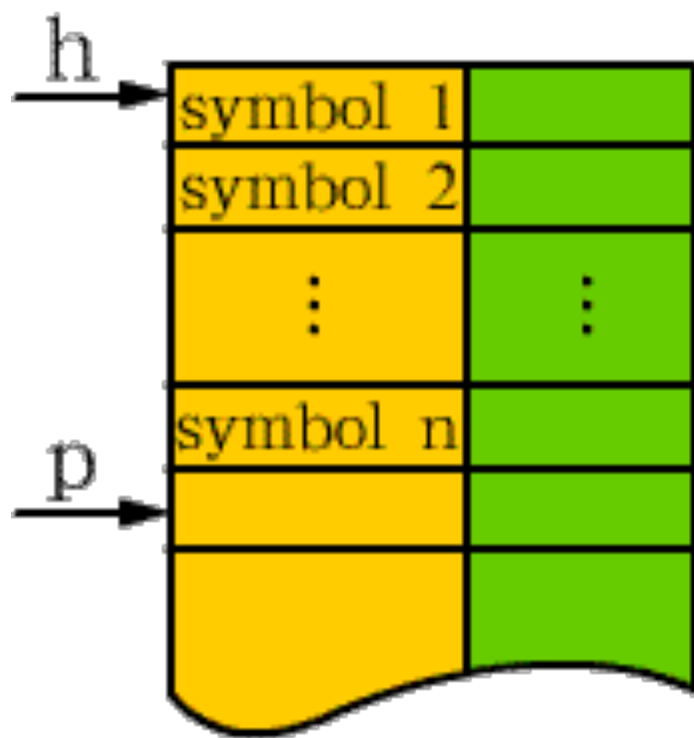
	符号	属性
0	┌	
1	┌	
2	┌	
3	┌	
	⋮	⋮
N-2	┌	
N-1	┌	

N-表长

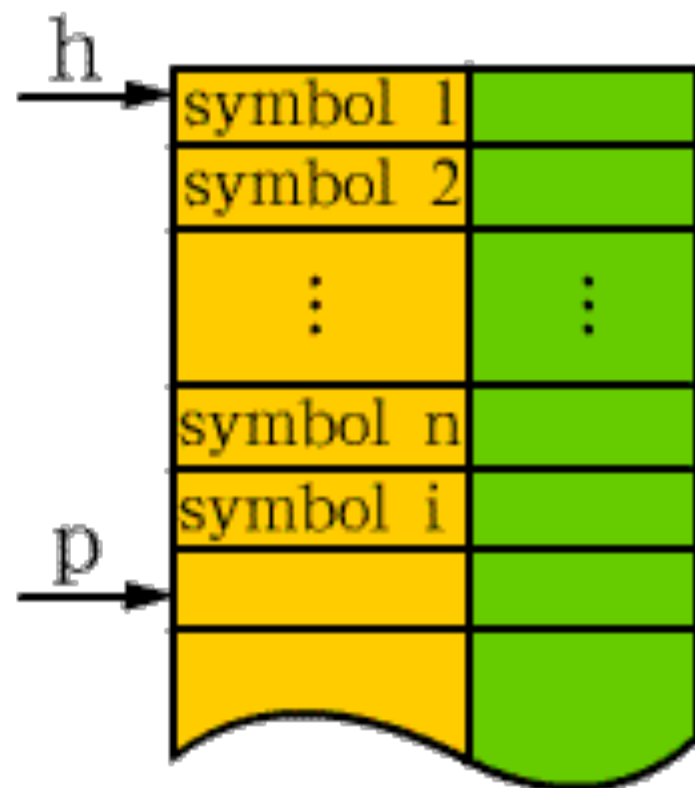
2 符号的登录

当编译程序从语言程序中获得一个标识符符号并确定该符号在符号表中尚不存在时，就要将此符号登录到符号表中。

线性符号表登录前后



登录前



登录后

排序符号表登录前后

h →	symbol 1	
	symbol 2	
	⋮	⋮
	symbol i	
	symbol j	
	⋮	⋮
p →	symbol n	

h →	symbol 1	
	symbol 2	
	⋮	⋮
	symbol i	
	symbol k	
	symbol j	
	⋮	⋮
p →	symbol n	

登陆符号 symbol k

 $\text{symbol } i < \text{symbol } k \leq \text{symbol } j$

3 符号的查找

每当编译程序从语言源程序获得一个符号，首先要确定该符号的类别。根据类别分别在相应的符号表中进行查找。

查找符号表的目的是建立或确认该符号的语义属性。

4 符号表中分程序结构层次的管理

对于具有分程序型结构的语言程序，不同层次分程序中定义的标识符号具有不同的作用域和不同的可视性规则。

通常对于具有分程序结构的语言可用两种方式组织它们的符号表：一是对每个分程序建立一个**独立的分表结构**的符号表；一是把各分程序符号组织在一张**单表结构**的符号表中。

源程序的形式

```
{ //第一层分程序
  int a;
  float b,d;
  { //第二层分程序
    int c;
    float a;
    { //第三层分程序
      int d;
      float c;
      { //第四层分程序
        float d;
        |
        a=b+c+d;
        |
      }
    }
  }
}
```

Level 1

Level 2

Level 3

Level 4

分程序结构的分表组织

第一层分程序符号表

int a
float b
float d

第二层分程序符号表

int c
float a

第三层分程序符号表

int d
float c

第四层分程序符号表

float d

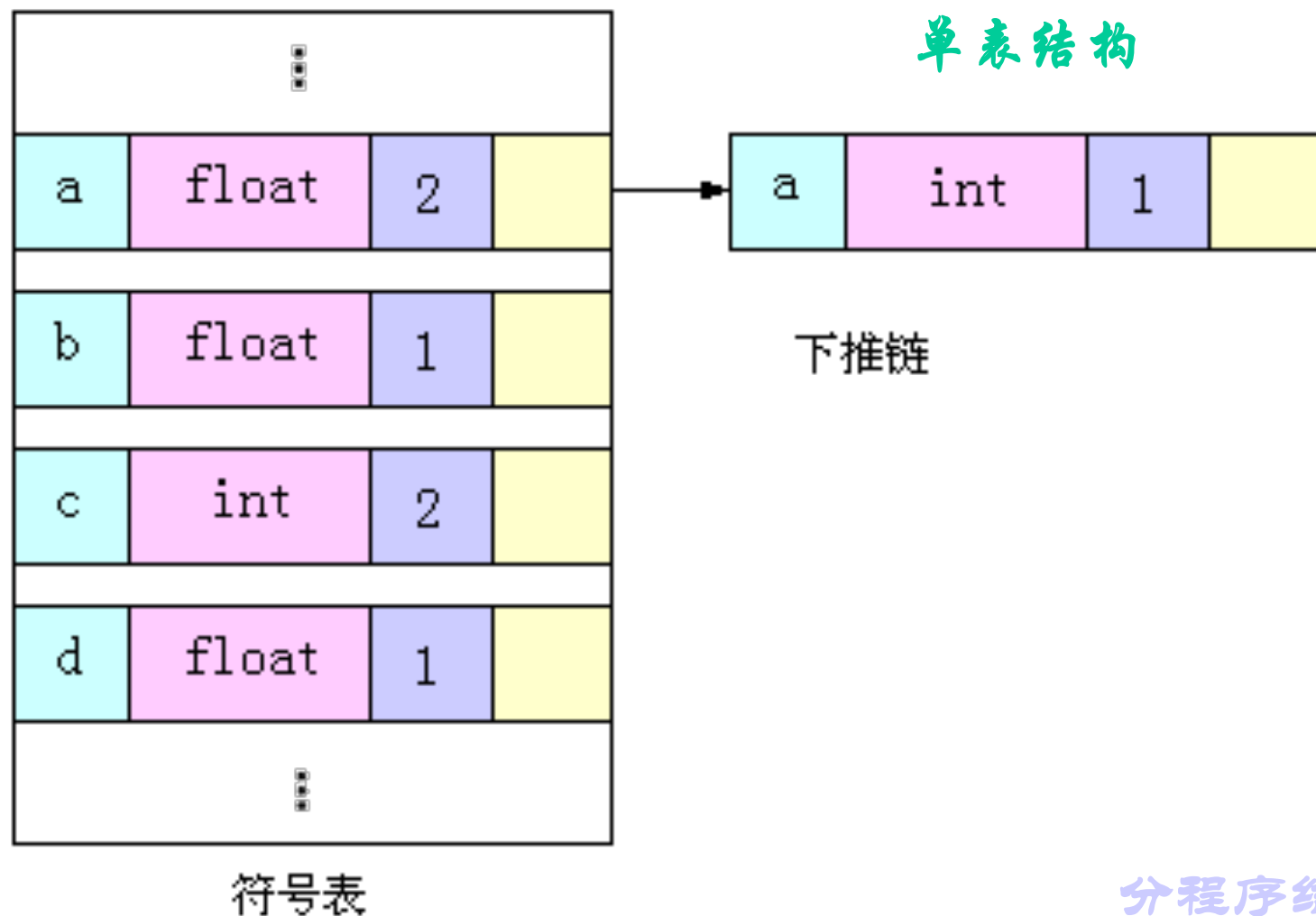
进入分程序第一层时的符号表层次属性描述

⋮			
a	int	1	
b	float	1	
⋮			
d	float	1	
⋮			
符号表			

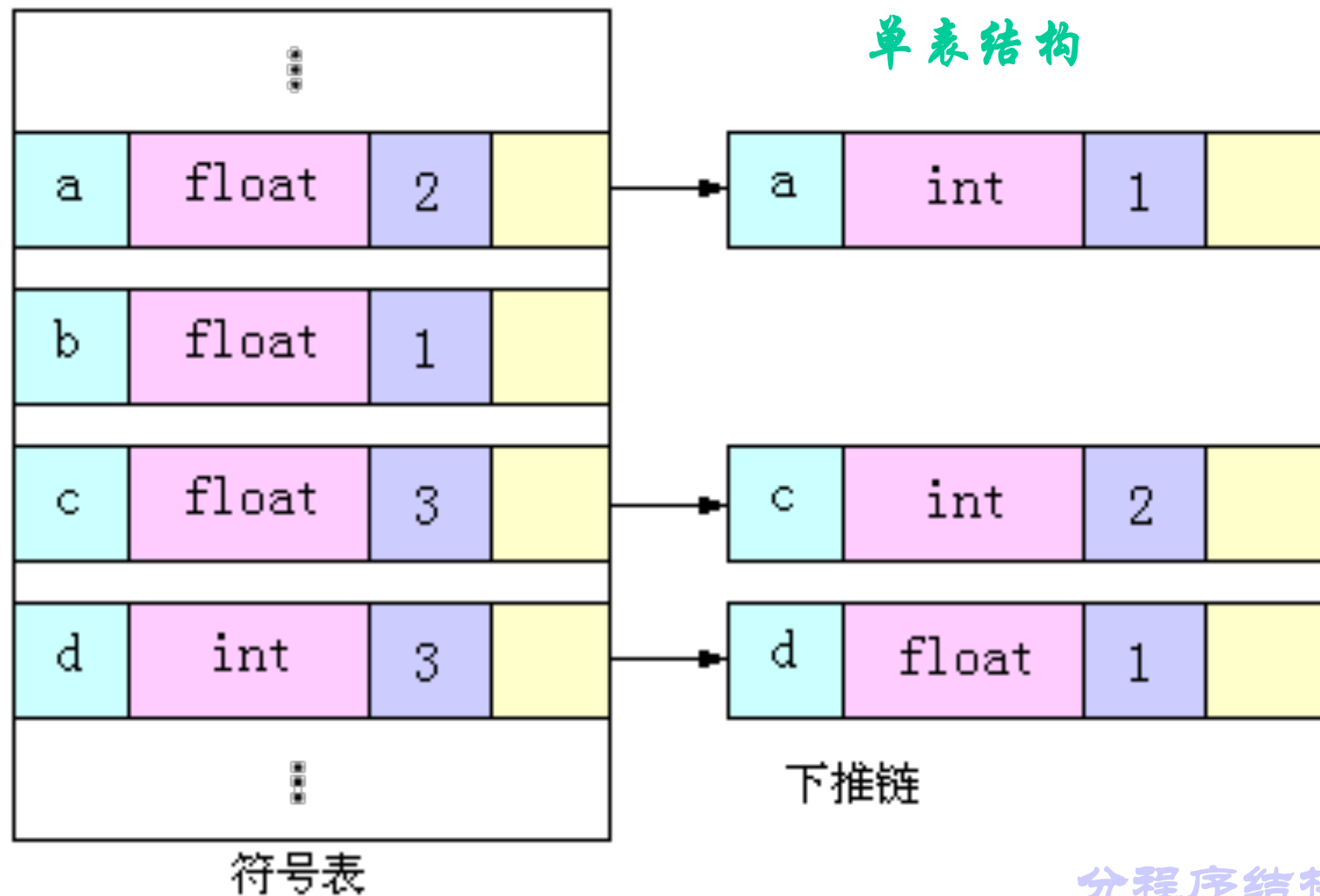
单表结构

分程序结构

进入分程序第二层时的符号表层次属性描述

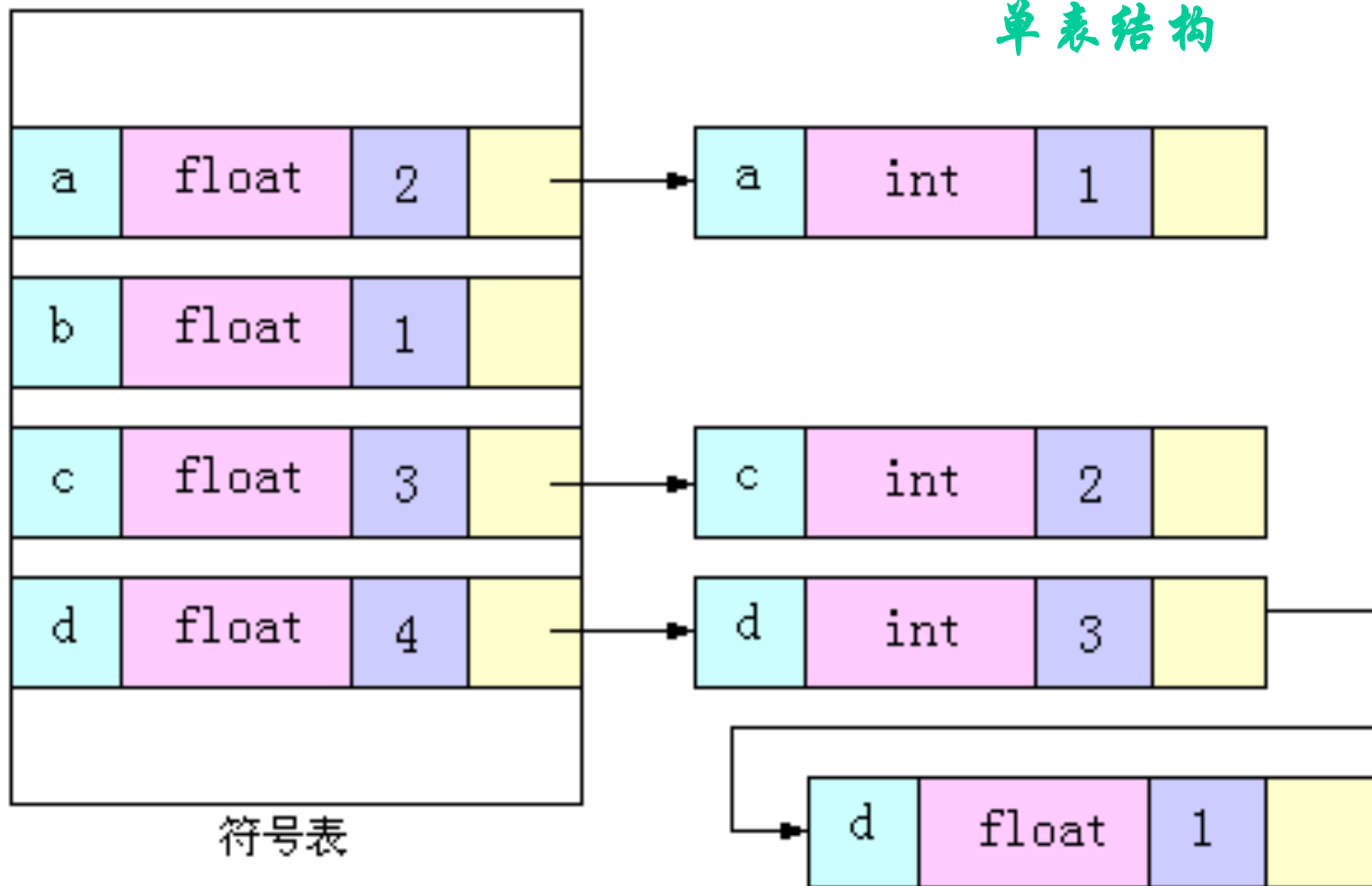


进入分程序第三层时的符号表层次属性描述



进入分程序第四层时的符号表层次属性描述

单表结构



分程序结构

练习

对照课件，阅读书上相关内容