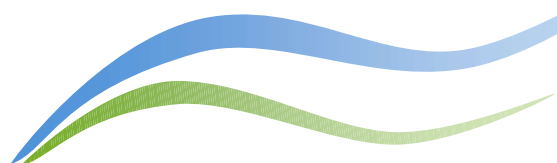


LOGO

第九章

PL/0符号表



广东工业大学计算机学院



符号表的功能

- ❖ 符号表中所登记的信息在编译的不同阶段都要用到。例如：在词法分析及语法分析过程中不断更新表中的信息，
- ❖ 另外，在词法分析到代码生成的各阶段则会从表中获取不同的属性信息。
- ❖ 符号表的功能主要归结为以下几个方面：
 - ❖ ① 收集符号属性
 - ❖ ② 作为上下文语义的合法性检查的依据
 - ❖ ③ 作为目标代码生成阶段地址分配的依据



1. 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。

❖ 例如：编译程序分析到下述两个说明语句：

```
int A;  
float B[5];
```

❖ 则在符号表中收集到关于符号**A**的属性是一个整型变量，关于符号**B**的属性是具有**5**个浮点型元素的一维数组。



2. 作为上下文语义的合法性检查的依据

❖ 通过符号表中属性记录可进行相应上下文的语义检查。

❖ 例如：在一个C语言程序中出现

```
int i[3,5]; // 定义整型数组i
```

...

```
float i[4,2]; // 定义实型数组i，重定义冲突
```

...

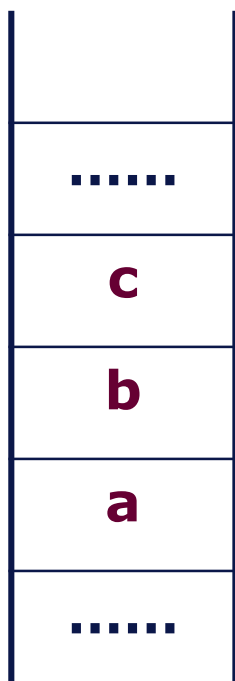
```
int i[3,5]; // 定义整型数组i，重定义冲突
```

❖ 从本例还可以看出，无论在后面两句中*i*的其它属性与前一句是否完全相同，只要标识符名重定义，就将产生重定义冲突的语义错误。



3. 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中的位置（主要是相对位置,由存储类别和被定义的位置决定！）。



- ❖ 要确定一个变量在目标代码生成时的地址，需要：
- ❖ 首先要确定其被分配的区域。例如，在C语言中有公共区、文件静态区、函数静态区、函数运行时的动态区等。
- ❖ 其次是根据变量出现的次序。通常使用在该区域中相对于区头的相对位置来决定该变量在某个区中所处的具体位置。

动态区头位置



PL/O符号表的建立

- 一、内容
- 二、结构
- 三、语法规则



PL/O符号表的建立

PL/O编译文本是在处理变量说明VAR和常量说明CONST以及过程说明部分时开始建立符号表table，并且按照标识符在源程序出现的先后顺序来建表。

例如：说明部分的处理任务就是对每个过程的说明对象造符号表，填写所在层次、标识符的属性和分配的相对位置等。标识符的属性不同时，所需要填写的信息也有所不同。登录信息是调用ENTER过程完成的。（见书**381**）



PL/O符号表结构

	NAME	KIND	LEVEL/VAL	ADR	SIZE
TXo→	A B C D E P	CONSTANT CONSTANT VARIABLE VARIABLE VARIABLE EPROCEDUR	35 49 LEV LEV LEV LEV	DX DX+1 DX+2 过程p的入口（待填）	4
TX→	G ...	VARIABLE ...	LEV+1 ...	ADR: DX ...	



PL/0符号表定义

```
enum object {constant,variable,procedure};
```

```
struct tablestruct {char name[a];
```

```
    enum object kind;
```

```
    int val;
```

```
    int level;
```

```
    int adr;
```

```
    int size;}
```

```
struct tablestruct table[txmax];
```

//txmax为100，为符号表的容量



PL/0符号表内容解释

- (1) 标识符名: NAME
- (2) 类型: KIND ①CONSTANT(常量)
②VARIABLE(变量)
③PROCEDURE(过程)
- (3) 值或层次: VAL或LEVEL
- (4) 地址: ADR
- (5) 大小: SIZE



PL/0符号表内容解释

❖ 值或层次: **VAL**或**LEVEL**

(1)若类型为CONSTANT:存放常量的值;

(2)若类型为VARIABLE 或PROCEDURE: 存放所属分程序的层次, 主程序的层次为0; 在主程序中定义的内容层次为1, 依次类推。

(或者说P过程所在层次如果为lev, 则在P过程中定义的变量名程序则为lev+1)



PL/O符号表内容解释

❖ 地址：ADR

(1) 简单变量或常量：该量在数据区所占单元的绝对地址或相对地址；这里采用DX表示给本层局部变量分配的相对存储位置，每说明完一个变量后DX加1。(由于过程活动记录的头3个单元用于存放控制信息，局部变量依次存放其后，见书P200)

(2) 过程：该过程的分程序的入口地址（需要返填）



PL/0符号表内容解释

❖ 大小: SIZE (过程标识符才需要含有这个信息)

过程: 该过程局部变量的个数 (需要返填) +3
(DX+m, DX=3, m为过程中局部变量的数目)



PL/O符号表内容解释

❖ PL/O符号表举例

✧ 例：右边程序的说明
部分分析后的符号表如下所示

```
CONST A=35, B=49;  
VAR C, D, E;  
PROCEDURE P;  
    VAR G, X, Y, Z;
```

NAME: A	KIND: CONSTANT	VAL: 35		
NAME: B	KIND: CONSTANT	VAL: 49		
NAME: C	KIND: VARIABLE	LEVEL: LEV	ADDR: DX	
NAME: D	KIND: VARIABLE	LEVEL: LEV	ADDR: DX+1	
NAME: E	KIND: VARIABLE	LEVEL: LEV	ADDR: DX+2	
NAME: P	KIND: PROCEDUR	LEVEL: LEV	ADDR:	SIZE: 7
NAME: G	KIND: VARIABLE	LEVEL: LEV+1	ADDR: DX	
.....				



PL/0符号表内容解释

✧ PL/0符号表举例2

✧ 例：右边程序在处理到/*here*/时的符号表如下所示

NAME	KIND	VAL / LEVEL	ADDR	SIZE
a	CONSTANT	25		
x	VARIABLE	LEV	DX	
y	VARIABLE	LEV	DX+1	
p	PROCEDUR	LEV		4
r	PROCEDUR	LEV		5
x	VARIABLE	LEV+1	DX	
s	VARIABLE	LEV+1	DX+1	
t	PROCEDUR	LEV+1		4

```
const a=25;
var x,y;
procedure p;
  var z;
  begin
    .....
  end;
procedure r;
  var x, s;
  procedure t;
    var v;
    begin
      .....
    end;
  begin /*here*/
    .....
  end;
begin
  .....
end.
```



PL/O符号表内容解释

❖ 注意事项

- (1) **LEVEL**: 过程说明语句PROCEDURE的层次与该过程中的局部变量的层次相差1。
- (2) **ADR**: 在PL/O中使用一个运行栈，而运行栈的前3个单元即0、1、2单元分别为SL:静态链；DL: 动态链；RA: 返回地址；所以从第3个单元开始给变量分配单元。因此，每个过程中变量的相对起始地址在Block内置初值 $DX:=3$ 。（增加一个变量则再加1）



PL/O符号表管理

1.符号表操作的相关参数

TX为表头索引（为当前过程里相应的符号表的表头位置，见例）。

登录(在符号表中插入一项)

```
void enter(enum object k, int* ptx, int lev, int* pdx)
```

```
/* k: 名字种类 const,var or procedure
```

```
ptx: 名字表尾指针的指针
```

```
lev: 名字所在的层次
```

```
pdx: 当前应分配变量的相对地址，分配后增加1
```

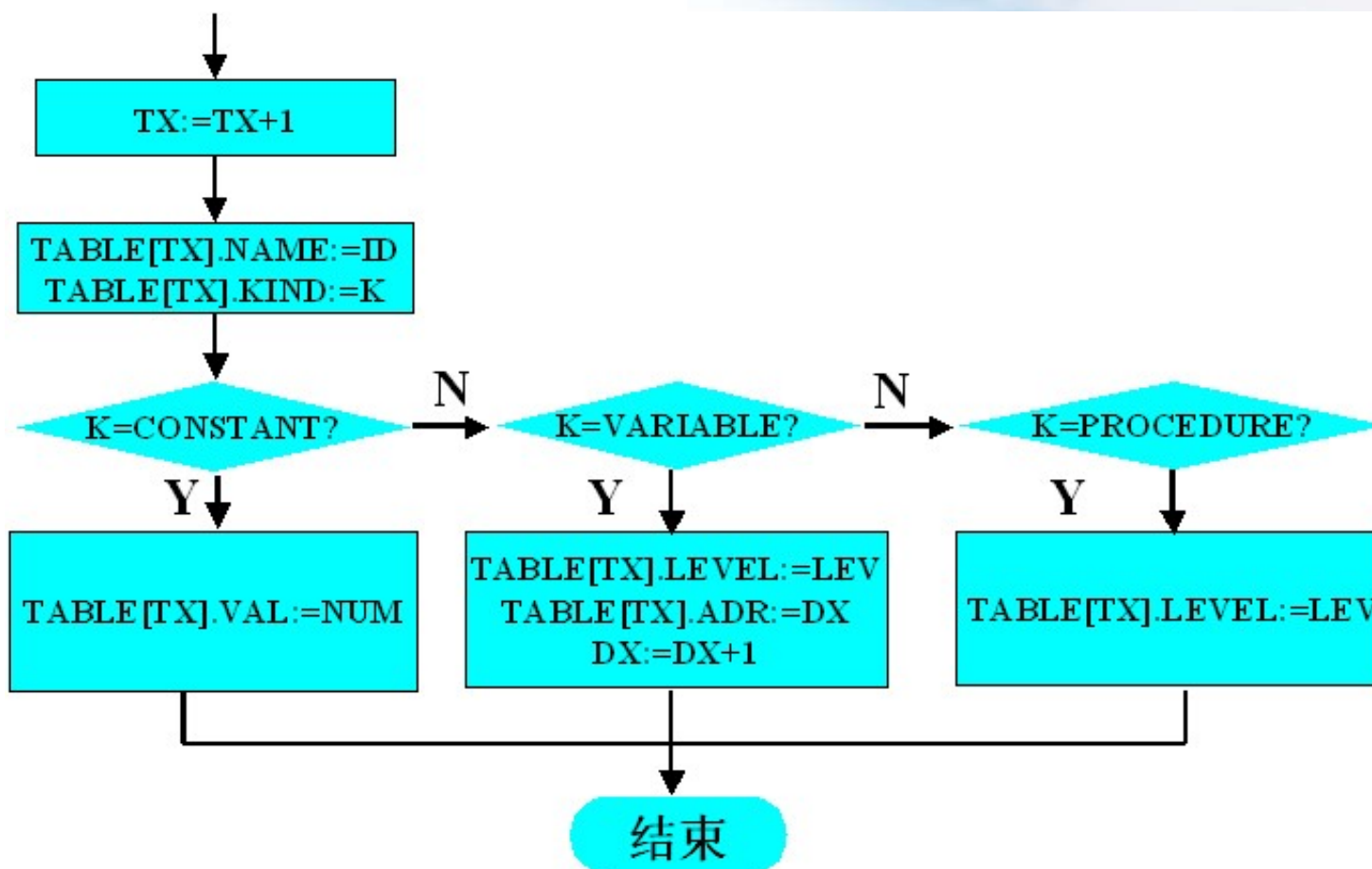
```
*/
```

见书**P440**



PL/0符号表管理

❖ enter()函数具体操作流程





PL/O符号表管理

查询

```
int position(char * idt, int tx)
```

```
/* idt: 被查标识符名字串
```

```
   tx: 符号表栈当前栈顶的位置, 表尾指针
```

```
返回所查标识符在符号表栈中的位置, 没查到则返回0*/
```

见书**P441**

思考：为什么说这里的**tx**是表尾指针？



PL/O符号表进阶

- ❖ 增加一维数组类型
/*符号表中的类型*/
- ❖ **enum object{**
- ❖ **constant,**
- ❖ **variable,**
- ❖ **procedur,**
- ❖ **array, //数组**
- ❖ **};**

两种处理方式:

1. 不修改符号表, 把数组中的元素都看作变量, 将其一一放入符号表。需增加一个变量来记录数组个数。
2. 建立一个数组信息表, 在符号表里新增一项信息**ref**, 当名字为数组类型或数组变量名时, **ref**指向该数组在数组信息表中的位置;

主观题 10分



已知一PL/O源程序为:

CONST A=15, B=28;

VAR C, D, E;

PROCEDURE P;

VAR G ; ...

试填写它的符号表中的 (1) - (8) 号空。

NAME: A	KIND: CONST	VAL (3)		
NAME: B	KIND: (1)	VAL (4)		
NAME: C	KIND: VAR	LEVEL: 0	ADR: 3	
NAME: D	KIND: (2)	LEVEL: 0	ADR: (6)	
NAME: E	KIND: VAR	LEVEL: 0	ADR: (7)	
NAME: P	KIND: PROCEDUR	LEVEL: 0	ADR: 过程的入口 (待填)	SIZE: (8)
NAME: G	KIND: VAR	LEVEL: (5)	ADR: 3	
.....	

正常使用主观题需2.0以上版本雨课堂

作答