

## 第八章 运行环境与符号表

### 1、过程参数的传递方式有几种?简述“传地址”和“传值”的实现原理。

答：过程参数的传递方式有四种：传值调用(call by value)、传址调用(call by address)、传名调用以及传结果。

- 1) 传址调用的实现原理：当函数调用时，形式参数获取实际参数的内存地址。任何对形式参数的操作都是通过这个地址在内存中直接修改实际参数的值。这样，函数内部的任何更改都会反映到调用函数外部的实际参数上。这种方式的优点是可以直接修改实际参数的值，且传递成本低（仅传递地址），但缺点是可能会导致实际参数意料之外的修改。
- 2) 传值调用的实现原理：在传值方式中，当函数被调用时，创建形式参数的局部存储空间并将实际参数的值复制到新创建的局部存储空间中。函数内部对形式参数的操作只影响局部变量，不影响外部的实际参数。这种方式的优点是可以保护实际参数不被修改，缺点是如果参数体积较大，复制的成本也较高，而且无法通过函数直接修改实际参数的值。

补充：

- 3) 传名的实现原理：传名是另一种较少见的参数传递方式，它在某些特定的编程环境或语言中存在，比如在早期的 Algol 语言中。传名调用的核心是延迟计算。实际参数作为一个未求值的表达式传递到函数中，这意味着实际参数的表达式本身（而不是值）被传入函数。每当在函数体内引用该参数时，参数表达式被重新求值。这意味着如果参数表达式包含变量，而这些变量在函数调用过程中被修改，那么每次参数表达式的求值可能得到不同的结果。
- 4) 传结果的实现原理：调用者将实在参数的地址和值都传递给被调用者。每个形参在被调用者数据区中需要两个单元，分别存放实参值和实参地址。被调用者通过值单元存取值，调用结束后，返回前，按地址将结果写入实参。

### 2、C 语言中规定变量标识符的定义可分为extern, extern static, auto, local static 和register

五种存储类：

- (1) 对五种存储类所定义的每种变量，分别说明其作用域。
- (2) 试给出适合上述存储类变量的内存分配方式。
- (3) 符号表中登录的存储类属性，在编译过程中支持什么样的语义检查。

答：

(1)

- extern: 用于声明一个全局变量，其定义可能在别的文件或同一文件的不同位置。其作用域是全局的，即从声明点开始到文件末尾。
- extern static: 作用域为该定义所在的 C 程序文件。
- auto: 这是局部变量的默认存储类，作用域局限于变量所在的代码块。当代码块执行完毕，这些变量的生命周期结束。
- local static: 在函数内部声明的 static 变量，它的作用域限制于函数内部，但其生命周期贯穿整个程序执行期间。
- register: 让编译器尝试将变量存储在 CPU 的寄存器中，以便快速访问。其作用域限于变量所在的代码块。

(2)

extern: 通常在全局数据区或 BSS 段分配内存。

extern static: 每个 C 程序文件分别设置一个静态公共区。

auto: 在栈上分配内存。

local static: 在全局/静态数据区分配内存, 但仅在其所在的函数内部可见。

register: 尝试在 CPU 寄存器中存储, 如果不行就退回到栈上。

(3)

存储类属性被登记在符号表中, 帮助编译器进行如下语义检查:

- 实施标识符变量重复定义的合法性检查
- 确保变量的使用不超出其声明的作用域。
- 对于 auto 和 local static 等局部变量, 编译器需要管理它们的创建和销毁。
- 确保 extern 变量在其他编译单元中有相应的定义。
- 对于 register 类变量的特殊处理, 编译器需要检查是否有足够的寄存器可用。

3、下面的程序执行时输出的a 分别是什么?若

(1) 参数的传递办法为“传值”。

(2) 参数的传递办法为“传地址”。

```
program main (input,output);
```

```
procedure p(x,y,z);
```

```
begin
```

```
    y:=y+1;
```

```
    z:=z+x;
```

```
end;
```

```
begin
```

```
    a:=2;
```

```
    b:=3;
```

```
    p(a+b,a,a);
```

```
    print a
```

```
end.
```

答:

(1) 传值调用不影响实参的值,  $a = 2$ 。

(2) 当  $p(a+b, a, a)$  被调用时, 首先  $a+b$  被计算为 5, 传递给  $x$ 。然后,  $y$  (即  $a$ ) 被增加 1 变为 3, 接着  $z$  (即  $a$ ) 被更新为  $3+5=8$ 。所以最后  $a = 8$ 。