



《编译技术》

作业八：运行环境与符号表

学 号 22920212204396

姓 名 黄子安

2024 年 5 月 23 日

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。

过程参数传递的方式有**传地址**、**传值**、**传结果**三种方式

传地址：调用者将实参的地址传递给被调用者。每个形参在被调用者的数据区中需要一个存放实参地址的单元。被调用者对形参的所有引用都会作用在实参上

传值：调用者将实参的值计算出来并传递给被调用者。每个形参在被调用者的数据区中需要一个存放实参值的单元并会对实参的值进行一份拷贝。被调用者对形参的所有引用都是对存放值的形参单元的引用，不会影响到原来实参

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。

(1) 作用域

extern：变量在整个程序的所有文件中都可见。可以在一个文件中声明变量，在另一个文件中定义变量。

extern static：变量在文件内可见，作用域限制在声明它的文件内，不被其他文件所见。

auto：局部变量，作用域在定义它的代码块内，离开该代码块后变量失效。

local static：局部静态变量，作用域在定义它的代码块内，但生命周期贯穿整个程序执行过程。

register：局部变量，作用域在定义它的代码块内，建议将变量存储在 CPU 寄存器中以提高访问速度，但不保证一定存储在寄存器中。

(2) 内存分配方式

extern：在程序的**全局数据区**分配存储空间。

extern static: 在程序的**全局数据区**分配存储空间，但仅在定义它的文件内可见。

auto: 在**栈**上分配存储空间，生命周期随代码块的进出而分配和释放。

local static: 在程序的**全局数据区**分配存储空间，生命周期贯穿整个程序执行过程。

register: 在**寄存器**中分配存储空间，如果寄存器不足，则在**栈**上分配。

(3) 符号表中的存储类属性和语义检查

extern: 检查变量是否在其他文件中定义。

extern static: 检查变量是否仅在文件内使用，是否与其他文件的变量冲突。

auto: 检查变量是否在局部代码块内使用。

local static: 检查变量是否在局部代码块内使用，以及其生命周期是否符合要求。

register: 检查变量是否作为寄存器变量使用，是否存在寄存器分配冲突。

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.
```

(1) 传值：对形参的修改不会影响到原来的实参，所以 a 的值为 2

(2) 传地址分析：调用过程 $p(a+b, a, a)$ 时，实际参数 $a+b$ 的值为 5（但作为传地址没有意义，因为表达式不是变量）， a 和 a 的地址传递给 y 和 z 。p 过程执行时， x 的值为 5（不影响其他变量）， y 的地址是 a 的地址，执行 $y := y + 1$ 相当于 $a := a + 1$ ，所以 $a = 2 + 1 = 3$ ， z 的地址是 a 的地址，执行 $z := z + x$ 相当于 $a := a + x$ ，所以 $a = 3 + 5 = 8$ ，最后输出 $a = 8$