C语言与Java语言的符号表区别和特点

# C语言

大多数编程语言都可以分成三部分：声明(declaration)，表达式(expression)，语句模块(statement). 每部分都有专门的语法来定义，在上一节中，通过语法定义了C语言的变量声明，并通过解析器成功实现了变量声明的语法解析。对于C语言中的一段函数代码，便可分割成对应于上面所说的三部分。函数声明中的函数名，返回值和输入参数例如：

int fun(int arg1, int arg2);

就可以对应于上面三部分中的声明部分。函数的主体则对应于表达式和语句模块部分。

在对代码的声明部分进行语法解析时，我们需要构建一种数据结构，以便于支持具体的代码生成，这种数据结构，就是我们接下来要研究的符号表。符号表本质上是一种数据库，用来存储代码中的变量，函数调用等相关信息。该表以key-value 的方式存储数据。变量和函数的名字就用来对应表中的key部分，value部分包含一系列信息，例如变量的类型，所占据的字节长度，或是函数的返回值。当我们的解析器读取源代码，遇到声明部分时，便给符号表添加一条记录，如果变量或函数脱离了它的作用范围时，便将他们对应的记录从表中删除。例如：

{

int variable = 0;

}

在上面代码中，进入大括号时，解析器遇到变量的声明，于是便把变量variable 的相关信息写入符号表。当解析器读取到右括号时，便把variable在符号表中的信息给删除，因为出了variable的作用范围只在括号之内。

符号表还可以用来存储类型定义(typedef)和常量声明，在词法解析的过程，词法解析器还需要和符号表交互，用于确定一个变量名是否属于一种类型定义，例如：

typedef char SingleByte

当词法解析器读取SingleByte这个字符串后，会在符号表中查询这个字符串所对应的记录，由于每个记录都有一个标志位，用来表明该字符串是否属于变量声明，于是词法解析器从记录中读取这个标志位，发现SingleByte对应的标志位被设置为1，因此词法解析器就不会把SingleByte当做普通的变量处理，而是当做关键字来处理。

符号表作为一种数据库，它必须具备以下特点：

1. 速度：由于符号表会被编译器频繁写入和读取，因此记录的写入，查询速度必须足够快。为了保证速度，整个符号表会直接存储在内存中，由此符号表的设计必须仔细考虑内存消耗。
2. 维护性。符号表几乎是编译器中，最复杂的数据结构。它的设计必须灵活可扩展，使得除了编译器外，其他应用程序或模块也能良好的访问符号表。
3. 灵活性。C语言的变量声明系统很复杂，例如它允许类型关键字的相互组合等(long int, long doube \*…), 因此符号表必须能支持各种不同的变量声明方式。
4. 重复性支持。由于对大多数编程语言而言，在不同的间套下，重复的变量名是允许的：

int variable = 0;

{

int variable = 1;

}

例如上面例子中，两个变量虽然拥有相同的名字，但却是合法的。在大括号内的变量会覆盖(shadow) 外层同名变量。因此符号表必须支持同一个key, 但却可以映射到不同的value.

1. 易删除。由于变量可能随时超出作用范围，因此一旦语法解析器发现变量失效后，必须能快速的将其从符号表中删除。

## 符号表的数据结构设计

由于哈希表的插入和删除平均耗时是O(1), 因此它能满足快速的插入和删除这一要求，如果遇到作用域不同的同名变量，他们必然被哈希到同一个位置，那么我们可以用链表把哈希到同一个地方的记录串联起来，这样就解决了支持重复性的问题。举个具体例子：

int Godot;

void waiting(int vladmir, int estragon)

{

int pozzo;

while (condition)

{

int pozzo, lucky;

}

}

在上面的代码中，Godot 和 waiting是属于第一层的变量，函数waiting的参数vladmir, estragon ,和内部变量 pozzo 属于第二层的变量。while 体内的变量 pozzo, 和 lucky 属于第三层的变量，而且两个pozzo是同名变量。于是通过链式哈希表来实现符号表的过程如下：

---------------------

作者：tyler\_download

来源：CSDN

原文：https://blog.csdn.net/tyler\_download/article/details/52437687?utm\_source=copy

版权声明：本文为博主原创文章，转载请附上博文链接！

---------------------

作者：tyler\_download

来源：CSDN

原文：https://blog.csdn.net/tyler\_download/article/details/52437687?utm\_source=copy

版权声明：本文为博主原创文 章，转载请附上博文链接！