

实验报告—step5

2018011340 计 83 郭峥岩

一、 实验内容

1. 整体框架

本次实验中 step5 的整体架构与之前的框架相比最重要的是多了一个栈帧的概念，由于本次实验中设计到的都是局部变量，因此只需要在栈帧中在函数调用的时候进行存储即可。

2. Step5 的功能实现

(1) 实现程序的语法规则的改动：

不再是单一的一条 return 语句，同时定义赋值语句，对原语法产生式进行修改即可。

(2) IR 类的更改和产生：

增加几个新的 IR 类：irFrameAddr：用于管理栈帧地址。
irBranch：用于跳转指令；irLabel：用于产生行的 label（对后面的 if-else 等语句也会使用，在此先进行定义以打好基础）

重点是在 ir 代码的生成是定义一个管理内存的类，负责在新定义一个新的变量的时候存储其在内存中的相对位置。以及使用的时候查找相对位置。

在此处定义的时候可以实现防止变量的重复定义以及使用之前使用过的变量的报错处理。

(3) 汇编代码的产生：

汇编代码的产生即按照实验指导书的内容将其从 IR 指令逐步翻译成汇编指令即可。

为了方便之后的步骤的使用，此处可以进行代码结构的调整和存储。

二、 思考题

1. 描述程序运行过程中函数栈帧的构成，分成哪几个部分？每个部分多用空间最少是多少？

解答：

本次实验的情况：

栈帧分为计算表达式所使用的运算栈，当前可用所有的局部变量和返回地址、老的栈帧信息三部分：

计算表达式所用的运算栈所用空间大小为 0

局部变量所用空间最小为 0

当前可用所有局部变量和返回地址，老的栈帧基址等信息最少为 8 字节。

另一种情况：其余如下所示：（假设一个内存地址所占的内存空间是 x 个字节）

-返回地址： x

-上一个栈帧的地址（可选）：0，如果使用： x

-存储的寄存器：0，如果没有使用到的或者要保存的就不存

-局部变量：0

-当前过程的输入参数：0

2. 如果MiniDecaf也允许多次定义同名变量，并规定新的定义会覆盖之前的同名定义，请问在你的实现中，需要对定义变量和查找变量的逻辑做怎样的修改？

解答：在定义一个新的变量的时候如果遇到同名的局部变量则不再报错，对于同名的变量可以添加另一个区分符号，即变量名不再作为标识的唯一 ID，并且把这个变量在后边出现的都添加上这一区分符号，直到再次被重新定义，需要更新区分符号。经过这样的处理，就可以将其看做不同的变量，而其他的逻辑则不受影响。

三、 参考资料

助教所写的代码：md-dzy branch

四、 总结

本次代码过程所出现的问题粗心把 sw 写成 lw 导致出错。最后改正。

总体而言进行的修改比较多。