

Principles of Compilers

Experiment: Stage-5

Chenghua Liu*

liuch18@mails.tsinghua.edu.cn

Department of Computer Science

Tsinghua University

目录

1	实验内容及过程	2
1.1	step11	2
2	思考题	3
2.1	step11	3

*刘程华, 学号 2018011687

1 实验内容及过程

1.1 step11

step11 的目标是支持数组：数组的声明，数组的下标操作。

借助实验指导书，回顾整个框架流程，主要在以下部分进行修改：

- 词法

中括号。

- 语法

新增语法树结点 `ArrayRef` 和 `ArrayExpr`，分别代表数组元素引用和数组元素求值。修改声明语句使其可以接受数组类型并解析其维度信息，修改表达式 `primary` 非终结符使其接受 `ArrayExpr`。

- 语义

检查数组声明时各个维度的大小均为正数，数组引用时下标数量和数组维度数量相同。

- TAC

新增 `ALLOC` 指令，用于分配局部数组。step10 中的 `addr` 被用来当作数组符号的基地址。如果在使用数组的时候，发现 `addr` 为空，对于全局数组通过 `LOAD_ADDRESS` 加载，对于局部数组通过 `ALLOC` 分配。数组元素的实际位置通过一系列乘法和加法进行运算，并保存在 TAC 指令的 `addr` 属性中，如果被 `ArrayExpr` 使用则通过 `LOAD` 加载数据，若被 `Assignment` 使用则通过 `STORE` 储存数据。

- 目标代码

生成时若遇到 `ALLOC` 则直接扩展当前栈空间，用扩展后的栈空间和 `fp` 寄存器计算局部数组的基地址。全局数组通过 `.bss` 段声明。

2 思考题

2.1 step11

1. C 语言规范规定，允许局部变量是可变长度的数组（Variable Length Array, VLA），在我们的实验中为了简化，选择不支持它。请你简要回答，如果我们决定支持一维的可变长度的数组（即允许类似 `int n = 5; int a[n];` 这种，但仍然不允许类似 `int n = ...; int m = ...; int a[n][m];` 这种），而且要求数组仍然保存在栈上（即不允许用堆上的动态内存申请，如 `malloc` 等来实现它），应该在现有的实现基础上做出那些改动？

提示：不能再像现在这样，在进入函数时统一给局部变量分配内存，在离开函数时统一释放内存。你可以认为可变长度的数组的长度不大于 0 是未定义行为，不需要处理。

答：

我们需要增加一个动态偏移量，函数开始时栈指针的偏移操作不考动态偏移量，生成各种基于栈顶寻址的指令和函数结尾释放占空间时需要考虑动态偏移量。该动态偏移量在生成指令的过程中动态更新，不会影响发生修改之前的指令，但会影响发生修改之后的指令。完成动态偏移量的增加之后，我们进行动态数组的分配，需要计算所需空间然后计算栈顶偏移并修改动态偏移量。