

1. 实验过程

在模板的基础上进行代码的编写。

1. 先从整个 `Module` 入手, `Module` 中只有两个部分, 全局变量的声明、初始化和函数的声明、实现;
2. 对于函数的实现, 主要是需要处理 `CompoundStmt`, 而这又相当于按顺序处理一个个 `Stmt`;
3. `Stmt` 有很多种类型, 分别处理:
 - a) 局部变量定义 `DeclStmt`
 - b) 二元赋值语句
 - c) `IfStmt`
 - d) `WhileStmt`
 - e) 函数调用 `CallExpr`
 - f) `Return` 语句
 - g) ...处理 `IfStmt` 和 `WhileStmt` 时又可以调用处理 `CompoundStmt` 的函数或者直接处理 `Stmt`;
4. 实现了一个函数专门用于获得表达式 `Exp` 的值, 需要处理:
 - a) 非赋值的二元运算符
 - b) 一元运算符
 - c) 整数字面量 `IntegerLiteral`
 - d) 字符串字面量 `StringLiteral`
 - e) 隐式类型转换

f) 函数调用 `CallExpr`

g) ...

该函数又可以递归调用自己，几乎所有需要用到表达式的值的地方，都可递归调用该函数。

以上就是代码的基本框架。

2. 遇到的困难以及解决方法

1. 对于形如 `[m x type]` 和 `type*` 类型的数组变量，其访问的形式似乎是不一样的。按照 Wiki 里面的教程，如果要访问 `a[1]`，则索引其实为 `[0,1]`，有两个。然而，在索引前都加个 0，对于 `[m x type]` 类型的数组变量是可以的，而 `type*` 类型的数组变量不行。

解决方法: Wiki 中访问数组的方法应该只适用于 `[m x type]` 类型的数组变量 (`%var=alloca [m x type]`)，我误解了。对于 `type*` 类型的数组变量 (`%var = alloca type*`)，不需要在索引前加 0，而 `type*` 类型的数组变量，常见于函数的参数。

3. 排行榜截图

提交详情 (#278)

测试报告

提交文件

回到作业详情

回到课程页面

测试报告

排行榜分数

performance 0.16974761796021387

score 634

提交排行

排行榜

名次	提交时间	performance	score
1	2023-04-25 22:12:03	0.3958229722848921	642
2	2023-05-09 21:48:20	0.2572473171407036	641
3	2023-05-16 19:06:03	0.20284044371482365	636
4	2023-05-16 20:34:40	0.19422272871890567	637
5	2023-05-15 11:03:58	0.183421911994549	637
6	2023-05-17 01:35:30	0.16974761796021387	634
7	2023-05-14 21:37:05	0.16967551492081845	637
8	2023-05-12 11:19:07	0.12091118321873637	610

每页个数 10 第 1/1 页