编译原理 stage 5 报告

傅子轩 2020010742 计01

主要改动

1. `parser`中增加了对数组定义和访问的解析
2. 增加了`ast`结点`ArrayType`和`ArrayRef`，`ArrayRef`接受一个`Lvalue`和一个`Expr`，分别作为基址和偏移
3. `build_sym`中对数组声明生成对应的`ArrayType`，对`ArrayRef`分别`accept`其`arr_base`和`index`
4. `type_check`中，对全局变量、数组这类通过地址访问的变量，对`VarRef::ATTR(lv_kind)`设置为`MEM_VAR`,对于保存在寄存器上的局部变量设置为`SIMPLE_VAR`。以在翻译阶段生成不同代码。对于`ArrayRef`，其`arr_base`的类型必须为`ArrayType`，`ArrayRef`的类型为`arr_base`的元素类型.
5. `translation`中对于`MEM_VAR`类型的左值，通过`LoadSymbol`或基址加偏移计算其地址。对于这种类型的左值，`LvalueExpr`使用`Load`，`AssignExpr`使用`Store`，否则使用`Assign`或直接访问。对于局部数组声明，使用`tac::ALLOC`在栈上分配空间
6. `riscv_md`中，先遍历所有指令，找到`ALLOC`，并在`RiscvStackManager`中预留空间，并记录偏移，随后再进行翻译。ALLOC指令的翻译就是简单的`addi dest, sp, offset`

思考题

step 11

先使用一块空间记录原`sp`，申请可变数组空间时，计算所需空间大小，减小`sp`以分配这块空间，返回时直接恢复原`sp`即可。(栈上的临时变量是用`fp`寻址的，所以`sp`的改变不会影响其他代码)