

1. 程序实现功能

将全局变量和条件语句翻译为LLVM IR。

2. 如何实现

2.1 全局变量

修改visitVarDef和visitConstDef函数，判断一下currentScope是不是globalScope。如果是的话，则进行全局变量的初始化。全局变量的初始化和局部变量初始化过程类似，但调用的api不同，关键代码如下：

```
// int类型
LLVMValueRef pointer = LLVMAddGlobal(module, i32Type, name);
LLVMSetInitializer(globalVar, value);

// 数组类型
LLVMTypeRef vectorType = LLVMVectorType(i32Type, size);
LLVMValueRef vectorPointer = LLVMAddGlobal(module, vectorType, name);
PointerPointer<Pointer> arguments = new PointerPointer<>(size);
for(int i=0; i<size; i++) arguments.put(i, value);
LLVMValueRef valueRef = LLVMConstVector(arguments, size);
LLVMSetInitializer(vectorPointer, valueRef);
```

2.2 条件语句

条件语句语法如下：IF L_PAREN cond R_PAREN stmt (ELSE stmt)?，其中cond为cond: exp | cond (LT | GT | LE | GE) cond | cond (EQ | NEQ) cond | cond AND cond | cond OR cond。

首先重写visitCond函数。如果cond是exp的话，说明cond就是exp，或者是一个比较的表达式两边的值，则返回visit(ctx.exp()); 如果不是exp的话，先得到左右两个cond返回的LLVMValueRef: left和right，然后根据各个情况生成比较指令tmp_ = LLVMBuildICmp(builder, LLVMIntPredicate, left, right, "tmp_")，最后将结果扩展成32位并返回。

然后重写visit条件语句的函数。创建三个基本块trueBlock、falseBlock和entryBlock。通过visit(ctx.cond())得到比较的结果，再将结果和0比较。如果不为0，则跳到trueBlock，反之跳到falseBlock。然后先通过LLVMPositionBuilderAtEnd(builder, trueBlock)指令，让后续指令生成在trueBlock块内，然后visit(ctx.stmt(0))，visit结束之后通过LLVMBuildBr(builder, entryBlock)跳到entryBlock。falseBlock同理。最后调用LLVMPositionBuilderAtEnd(builder, entryBlock)让后续指令都生成在entryBlock中。

3. 印象深刻的bug

3.1 normaltest2

报错：lli: lli: out.ir:27:3: error: instruction expected to be numbered '%1' %0 = srem i32 %frn7, 5 ^

解决：百度了一下这段报错：

另外值得一提的是临时寄存器的命名，在LLVM的Module里面，clang 默认生成的虚拟寄存器是按数字顺序命名的，LLVM 限制了所有数字命名的虚拟寄存器必须严格地从 0 开始递增，且每个函数参数和基本块都会占用一个编号。所以，在我们刚才手写的字节码里，寄存器是以 %1 开始的，因为包含该段指令的基本块隐式地占用了寄存器编号0。

如果将原函数改写如下：

```
define dso_local i32 @main(){
    %0 = sub i32 0, 15;
    %1 = add i32 %0, 0;
    %2 = mul i32 %1, 2;
    ret i32 0;
}
```

进行可视的字节码到二进制字节码的转换：

```
$ llvm-as test00.ll -o test00.bc
llvm-as: test00.ll:7:5: error: instruction expected to be numbered '%1'
    %0 = sub i32 0, 15;
```

然后发现我在算模的时候写的是LLVMBuildSRem(builder, zuo, you, ""), 没有给这个变量命名，可能和别的命名冲突了，改为LLVMBuildSRem(builder, zuo, you, "sRem")就正确了。

3.2 normaltest13

报错：Exception in thread "main" java.lang.ClassCastException: class SysYParser\$LvalContext cannot be cast to class SysYParser\$NumContext (SysYParser\LvalContext and SysYParser\$NumContext are in unnamed module of loader 'app')

解决：数组的下标可能是表达式。

3.3 hardtest1和hardtest3

报错：Exception in thread "main" java.lang.NullPointerException at Main\$MyVisitor.visitStmtAssign(Main.java:158) at Main\$MyVisitor.visitStmtAssign(Main.java:46)

解决：全局变量数组声明时没有初始化。改为都初始化为0之后就正确了。