

Lab5实验报告

201250141 刘屿

实验思路

1. 添加作用域来定义符号表，在进出 program, funcDef 与 block 时进行切换，符号表保存变量名与对应的指针 LLVMValueRef 间的映射
2. 添加函数与局部变量的翻译
3. 添加左值的翻译
4. 添加左值表达式与函数调用表达式的翻译
5. 完成左值与表达式的翻译后，再翻译赋值与返回语句，会很简单

精巧的设计

- 符号表中保存的是每个变量声明时通过 LLVMBuildAlloca 分配的指针
- 增加全局变量 hasReturn，用来判断函数体中有没有出现 return 语句
 - 避免 void 类型函数没写返回语句，可以在退出 funcDef 时手动添加
- 保证 exp 返回的 LLVMValueRef 不存在指针
 - 例如在翻译左值表达式时，lVal子节点返回了指针，只需要对指针使用 LLVMBuildLoad 返回即可
- 数组初始化时，将 buildGEP, store 操作抽出为单独的函数 buildGEP(int elementCount, LLVMValueRef varPointer, LLVMValueRef[] initArray)

遇到的困难及解决办法

- 不了解 LLVMBuildGEP 如何使用
 - 在 GitHub 上查找代码如何使用
- 对于某个操作不清楚 LLVM 如何实现
 - 使用 Clang 生成中间代码来查看应该如何实现，再查阅手册找到对应函数