

# 编译原理 Lab8 实验报告

姓名：熊丘桓

学号：201250127

邮箱：[eaglebear@smail.nju.edu.cn](mailto:eaglebear@smail.nju.edu.cn)

## 1. 实现功能

本次实验完成了以下功能：

1. 一维数组作为函数参数

## 2. 实验设计

本次实验主要修改了 `visitLVal()` 和 `visitLValExp()` 等方法。

```

1  @Override
2  public LLVMValueRef visitLValExp(SysYParser.LValExpContext ctx) {
3      LLVMValueRef lValPointer = this.visitLVal(ctx.lVal());
4      if (arrayAddr) {
5          arrayAddr = false;
6          return lValPointer;
7      }
8      return LLVMBuildLoad(builder, lValPointer, ctx.lVal().getText());
9  }
10
11 @Override
12 public LLVMValueRef visitLVal(SysYParser.LValContext ctx) {
13     String lValName = ctx.IDENT().getText();
14     LLVMValueRef varPointer = currentScope.resolve(lValName);
15     LLVMTypeRef varType = currentScope.getType(lValName);
16     if (varType.equals(i32Type)) {
17         return varPointer;
18     } else if (varType.equals(intPointerType)) {
19         if (ctx.exp().size() > 0) {
20             LLVMValueRef[] arrayPointer = new LLVMValueRef[1];
21             arrayPointer[0] = this.visit(ctx.exp(0));
22             PointerPointer<LLVMValueRef> indexPointer = new PointerPointer<>(arrayPointer);
23             LLVMValueRef pointer = LLVMBuildLoad(builder, varPointer, lValName);
24             return LLVMBuildGEP(builder, pointer, indexPointer, 1, "&" + lValName);
25         } else {
26             return varPointer;
27         }
28     } else {
29         LLVMValueRef[] arrayPointer = new LLVMValueRef[2];
30         arrayPointer[0] = zero;
31         if (ctx.exp().size() > 0) {
32             arrayPointer[1] = this.visit(ctx.exp(0));
33         } else {
34             arrayAddr = true;
35             arrayPointer[1] = zero;
36         }
37         PointerPointer<LLVMValueRef> indexPointer = new PointerPointer<>(arrayPointer);
38         return LLVMBuildGEP(builder, varPointer, indexPointer, 2, "&" + lValName);
39     }

```

```
40 } 
```

### 3. 实验困难

---

笔者完成实验代码编写后，遇到了 OJ 的报错：

```
1 | lli-13: lli: out.ir:44:1: error: expected instruction opcode } ^
```

笔者与同学交流发现，在返回值为 `int` 的函数末尾添加 `ret i32 0` 语句即可通过 OJ 的所有测试用例。笔者推测测试用例中存在没有正确返回返回值的函数。