编译原理 lab7 实验报告

201250107 李小路

实验思路：

1. 创建新的类 LLWhileStack，里面保存两个栈，分别持有 while 条件判断块和 while 块结束之后的基本块，同时提供 push 方法压栈，peek 方法找到最后进入的 while 块、entry 块。pop 弹出最后进入的块
2. 翻译 while 语句：首先通过 while 循环找到当前 while 语句对应的函数 LLVMValueRef，创建 whileCondition 基本块对应 while 后括号内的条件判断；之后创建 whileBody 块对应大括号内的内容、创建 entry 块对应 while 结束之后的语句，此时 whileCondition 和对应 entry 入栈，根据子节点 cond 判断，如果为 true 则跳转至 whileBody，否则跳转至 entry；跳转进入 whileBody 之后 visit 子节点 stmt，然后跳转回 whileCondition 再次进行条件判断是否跳出 while 循环；如果跳转至 entry 后则将对应的 while 循环与 entry 弹出栈；
3. 翻译 continue 和 break 语句：直接分别调用 while 栈的 peek 方法，continue 就跳转至 whileCondition 进行条件判断，break 则直接跳转至 entry；

印象深刻的 bug：

1. 栈中需要同时持有对应的 while 和 entry 分别用于 continue 和 break 的跳转、
2. 需要在结束一个 while 的翻译之后将该 while 出栈，不然会使得外部的 break 识别错误对应到不正确的 while 循环；同时 continue 和 break 则使用 peek 检索而不出栈；

参考：特别鸣谢 201250109 陆剑锋同学

Antlr 官网文档 All Classes (ANTLR 4 Runtime 4.11.1 API)

llvm 文档 Index (JavaCPP Presets for LLVM 15.0.3-1.5.8 API) (bytedeco.org)

`https://thedan64.github.io/inkwell///llvm_sys/core/fn.LLVMBuildZExt.html`