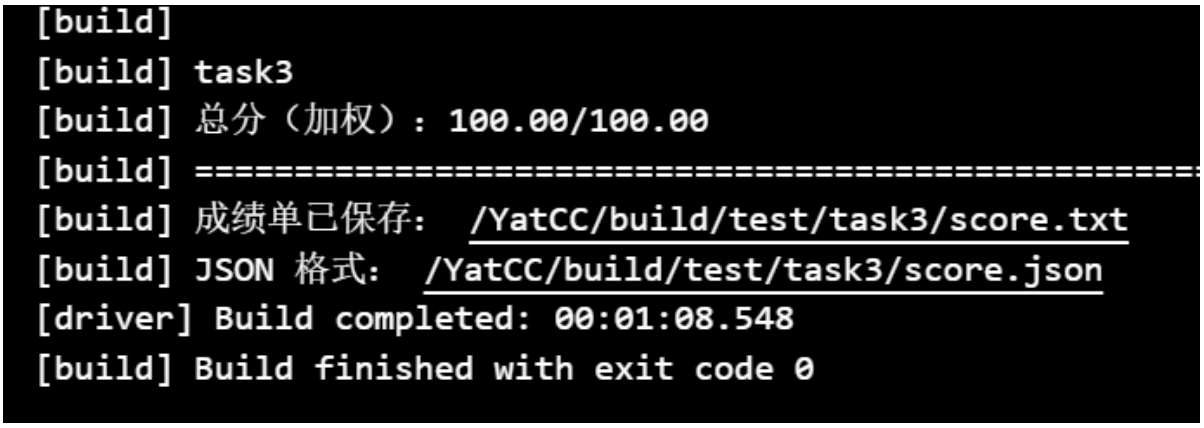
编译原理实验报告

task2——中间代码生成器

实验评分截图



实验过程

一、语句处理声明（EmitIR.hpp）

```
#include "asg.hpp"
#include <llvm/IR/IRBuilder.h>
#include <llvm/IR/LLVMContext.h>
#include <llvm/IR/Module.h>

class EmitIR
{
//...

  //============================================================================
  // 表达式
  //============================================================================

  llvm::Value* operator()(asg::Expr* obj);

  llvm::Constant* operator()(asg::IntegerLiteral* obj);

  llvm::Value* operator()(asg::BinaryExpr* obj);

  llvm::Value* operator()(asg::ImplicitCastExpr* obj);

  llvm::Value* operator()(asg::DeclRefExpr* obj);

  llvm::Value* operator()(asg::UnaryExpr* obj);

  llvm::Value* operator()(asg::ParenExpr* obj);

  llvm::Value* operator()(asg::InitListExpr * obj);
```

```
    llvm::Value* operator()(asg::CallExpr* obj);
    // TODO: 添加表达式处理相关声明

    //============================================================================
    // 语句
    //============================================================================

    void operator()(asg::Stmt* obj);

    void operator()(asg::CompoundStmt* obj);

    void operator()(asg::ReturnStmt* obj);

    void operator()(asg::DeclStmt* obj);

    void operator()(asg::ExprStmt* obj);

    void operator()(asg::IfStmt* obj);

    void operator()(asg::WhileStmt* obj);

    void operator()(asg::BreakStmt* obj);

    void operator()(asg::ContinueStmt* obj);

    // TODO: 添加语句处理相关声明

    //============================================================================
    // 声明
    //============================================================================

    void operator()(asg::Decl* obj);

    void trans_init(llvm::Type* ty, llvm::Value* val, asg::Expr* obj);

    void operator()(asg::VarDecl* obj);

    void operator()(asg::FunctionDecl* obj);

    // TODO: 添加声明处理相关声明
};
```

二、 EmitIR类的实现（EmitIR.cpp）

这是本次实验的核心任务，确保生成的 LLVM IR 能够正确执行。由于代码太长这里不做展示。

实验心得

实验实在太难了TT！！！！！！！！！！

通过本次实验，我掌握了如何使用 LLVM API 生成中间代码，理解了中间代码生成需严格遵循源语言语义，确保 IR 的正确性优先于代码优化；学会了调试和优化生成的 LLVM IR，确保其正确性和高效性；理解了 EmitIR 类的接口和功能。

希望下次的实验能简单些...