**第六章-WHILE语句设计文档**

同IF语句，参照FOR语句完成WHILE语句部分

一．WHILE的语法树

返回两个语句DO和WHILE

//while

class WhileStatAST :public StatAST {

std::unique\_ptr<StatAST> While, Do;

public:

WhileStatAST(std::unique\_ptr<StatAST> While, std::unique\_ptr<StatAST> Do) :

While(std::move(While)), Do(std::move(Do)) {}

Value \*codegen() override;

};

二.WHILE的解析

首先初始化

static std::unique\_ptr<StatAST> ParseWhileStat();

根据提供的实例代码，可以获得WHILE语句的一般规范，则可以解析有

//解析 while

static std::unique\_ptr<StatAST> ParseWhileStat() {

getNextToken(); // eat the while.

// while.

auto While = ParseExpression();

if (!While)

return nullptr;

if (CurTok != DO)

return LogError("expected do");

getNextToken(); // eat the Do

if (CurTok != '{')

return LogError("expected {");

getNextToken(); // eat the {

auto Do = ParseStatement();

if (!Do)

return nullptr;

if (CurTok != '}')

return LogError("expected }");

getNextToken(); // eat the }

if (CurTok != DONE)

return LogError("expect DONE in WHILE statement");

getNextToken();//eat DONE

return llvm::make\_unique<WhileStatAST>(std::move(While), std::move(Do));

}

参照FOR，将DO语句里{}的函数块作为循环主体，WHILE语句作为判断条件，获取并返回。

三．IR代码生成

1.参照IF语句，收件获得While语句并将其处理并于0比较，得到WhileV

Value \*WhileV = While->codegen();

if (!WhileV)

return nullptr;

WhileV = Builder.CreateICmpNE(WhileV, Builder.getInt32(0),

2.参照IF语句和FOR语句，我们将

BasicBlock \*LoopBB = BasicBlock::Create(TheContext, "loop", TheFunction);

BasicBlock \*AfterBB = BasicBlock::Create(TheContext, "afterLoop", TheFunction);

3.作为循环主体和判断是否跳出循环的条件，并进行三角判断

//类似if判断

Builder.CreateCondBr(WhileV, LoopBB, AfterBB);

当While成立的时候就一直loop否则跳出循环

4.经试验和摸索之后完整代码如下

Value \*WhileStatAST::codegen() {

Function \*TheFunction = Builder.GetInsertBlock()->getParent();

BasicBlock \*LoopBB = BasicBlock::Create(TheContext, "loop", TheFunction);

BasicBlock \*AfterBB = BasicBlock::Create(TheContext, "afterLoop", TheFunction);

Value \*WhileV = While->codegen();

if (!WhileV)

return nullptr;

WhileV = Builder.CreateICmpNE(WhileV, Builder.getInt32(0),

"inLoop");

//类似if判断

Builder.CreateCondBr(WhileV, LoopBB, AfterBB);

Builder.SetInsertPoint(LoopBB);

Value \*DoV = Do->codegen();

if (!DoV)

return nullptr;

WhileV = Builder.CreateICmpNE(While->codegen(),

Builder.getInt32(0), "outLoop");

Builder.CreateCondBr(WhileV, LoopBB, AfterBB);

Builder.SetInsertPoint(AfterBB);

return Builder.getInt32(0);

}