

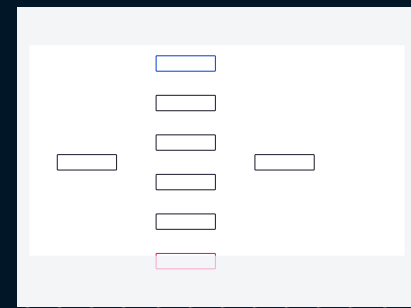
C To LLVM

小组成员：王澳 张智

语法支持

- `include`头文件
- 基础类型支持（包括`int`, `double`, `char`, `void`）
- 函数声明定义
- 结构体声明定义
- 赋值语句
- 变量定义语句（包括基础类型，数组，结构体等）
- 条件语句（包括`if`, `else if`, `else`）
- 循环语句（包括`for`, `while`）
- 表达式（包括各种运算，变量及函数调用等）

模块划分



词法分析 & 语法分析

```
prog: (include)* (declaration | statement)*;

include: '#include' '<' HEADER '>';

declaration: functionDeclaration | structDeclaration;

functionDeclaration:
    vType vId '(' paramsDefinitionPattern ')' '{' statement* '}';

structDeclaration:
    vStruct '{' (structMemberDeclaration)+ '}' ';';

structMemberDeclaration:
    (vType | vStruct) (vId | vArray) (',' (vId | vArray))* ';';

statement:
    variableDefinitionStatement
    | arrayDefinitionStatement
    | structDefinitionStatement
    | assignStatement
    | conditionStatement
    | whileStatement
    | forStatement
    | returnStatement
    | breakStatement
    | continueStatement
    | funcStatement;
```

Antlr Grammar File

语义分析

- 使用Visitor遍历模式
- 处理Scope作用域链
- 处理局部变量和全局变量
- 错误处理

中间代码生成

```

; ModuleID = ""
target triple = "x86_64-pc-linux-gnu"
target datalayout = "e-m:e-i64:64-f80:128-n8:16:32:64-S128"

declare i32 @"printf"(i8* %.1", ...)

declare i32 @"gets"(i8* %.1")

declare i32 @"strlen"(i8* %.1")

define i32 @"main"()
{
    main.body:
    %"a" = alloca i32
    store i32 1, i32* %"a"
    %".3" = getelementptr inbounds [18 x i8], [18 x i8]* @".string1", i32 0, i32 0
    %".4" = call i32 @printf(i8* %.3")
    ret i32 0
}

@".string1" = constant [18 x i8] c"Hello, C To LLVM\0a\00"
```

```
#include <stdlib.h>
```

```
int main()
{
    int a=1;
    printf("Hello, C To LLVM\n");
    return 0;
}
```

样例演示

sort ▼

Compile

Run

Loading...

Loading...

特点和难点

- 采用SpiderMonkey的AST结构，易于扩展
- 语法支持丰富，包括结构体的使用等等
- 生成LLVM中间表示代码，易于平台移植，效率高

Thank you!