

编译原理-期末项目

期末项目

- 总分占比：40%
- 团队人数：4人
- 提交内容：代码(github)/程序设计文档(word/PDF)/答辩(PPT)
- 考核内容：
 - a. 编译器构造算法和编译程序设计方法
 - b. 需求分析、算法设计、代码编写和开发文档撰写等系统开发能力
 - c. 文献查阅、使用现代工具和利用开源软件项目资源进行编译器二次开发的能力
 - d. 团队管理组织能力、团队合作能力，强调共同参与, 分工协作
 - e. 总结、表达、展示、沟通和思辨能力

1. 编译算法实现

- **Mission:** 掌握编译器构造的经典算法
- **Task:**

请选择一种高级程序设计语言，选择课内学习的一种词法分析器构造算法、语法分析器构造算法和中间代码生成算法实现一个小语言PL/0的简单编译器(Compiler)。

- a. 要求编译器的输入为符合PL/0语言源程序，输出为中间代码（三地址代码）表示的程序；
 - b. 要求编译器是个一遍的编译程序，词法分析程序作为子程序，需要的时候被语法分析程序调用；
 - c. 要求使用语法制导的翻译技术，在语法分析的同时生成中间代码，并保存到文件中。
- **Candidate Algorithms:**
 - a. 词法分析：Thompson算法、子集法、等价状态法等
 - b. 语法分析：递归下降分析法、预测分析程序、LR分析法等
 - c. 中间代码生成：属性文法、翻译子程序等
 - PL/0语言的语法规则：
 - a. 词法规则

关键字：PROGRAM、BEGIN、END、CONST、VAR、WHILE、DO、IF、THEN。

标识符：以字母开头的、由字母和数字组成的字符串。

整数：数字开头的数字串。

算符、界符：+、-、*、/、:=、=、<>、>、>=、<、<=、（、）、；、，

b. 语法规则

<程序>→<程序首部> <分程序>

<程序首部>→PROGRAM <标识符>

<分程序>→[<常量说明>][<变量说明>]<语句部分>（注：[]中的项表示可选）

<常量说明>→CONST <常量定义>{, <常量定义>}（注：{}中的项表示可重复若干次）

<常量定义>→<标识符>=<无符号整数>

<无符号整数>→<数字>{, <数字>}

<变量说明>→VAR<标识符>{, <标识符>}

<标识符>→<字母>{<字母> | <数字>}

<语句部分>→<语句> | <复合语句>

<复合语句>→BEGIN <语句>; <语句> END

<语句>→<赋值语句> | <条件语句> | <循环语句> | <复合语句> | <空语句>

<赋值语句>→<标识符>:=<表达式>

<表达式>→[+|-]项 | <表达式> <加法运算符> <项>

<项>→<因子> | <项><乘法运算符> <因子>

<因子>→<标识符> | <常量> | (<表达式>)

<加法运算符>→+ | -

<乘法运算符>→* | /

<条件语句>→IF <条件> THEN <语句>

<循环语句>→WHILE <条件> DO <语句>

<条件>→<表达式> <关系运算符> <表达式>

<关系运算符>→= | <> | < | <= | > | >=

<字母>→a | b | … | x | y | z

<数字>→0 | 1 | … | 8 | 9

c. PL/0 源程序示例：

PROGRAM add

VAR x,y;

BEGIN

```
x:=1;

y:=2;

WHILE x<5 DO x:=x+1;

IF y>0 THEN y:=y-1;

y:=y+x;

END
```

2. 编译工具使用

- **Mission:** 掌握现代工具进行编译器二次开发

- **Task:**

请选择compiler-compiler工具，生成PL/0语言的编译器

- **Candidate Tools:**

1. Flex/Bison

- a. 词法分析器生成器 Flex (Lex的升级版) 的官网 :<https://github.com/westes/flex>
- b. 语法分析器生成器 Bison (Yacc的升级版) 的官网:<http://www.gnu.org/software/bison>

2. ANTLR (ANother Tool for Language Recognition)

- a. 一种包括词法分析、语法分析、中间代码生成、优化、目标代码生成的编译器生成工具
- b. 官网: <https://wwwantlr.org>