

# Difference Introduction of Compiler Project3

A10515001 李大祥  
Due:2017-06-15-10:00

## Between Project1 and Project2

### 1.scanner.l 部分

project2 中与 project1 的文法部分相同，但在 project2 中删除了第三部分的 print symbol table 功能。

### 2.makefile 部分

把 gcc 与 yacc(bison)命令添加到 makefile 里面，简化测试流程。

生成的程序的名称从 scanner 改为 project2。

### 3.yacl.y 部分

新增了 yacc 部分的文法，接受 project1 传来的 token，并在 expression 后面 print 出来，同时新增了 symbol table 的 columns。

## Between Project2 and Project3

### 1.scanner.l 部分

修改 token 类型属性名字，只是对命名的修改，并不影响文法。

在第三部分把 yywrap() 和 yyerror(const char \*str) 重新放回了这里，这样是为了精简 yacl.y 档案。

### 2.makefile 部分

在实验室机器上发现 makefile 里面 yacc 命令与 bison 命令结果不同，所以重新修改成 bison。

生成的程序的名称从 project2 改为 go\_compiler。

在删除命令 rm 中新增了 .iasm、go\_compiler、\*.class 三种。

### 3.yacl.y 部分

#### 3.1 yacl.y 文法部分

(a) 删除了很多没有用到的文法，比如乘除、大于等于这些文法。

(b) 在相关的 expression 后面新增了对 expression 的“翻译”，这个“翻译”是指翻译成 jasm code。这些“翻译”后的结果会暂存在 jasm array 中，等待最后一起写入一个 .iasm 类型的文件里。

(c) 在 yacl.y 的第一部分，新增了类似 jasm、temp\_parameter、is\_print 这些变数。jasm array 是用来保存所有“翻译”完成的 jasm code，类似 temp\_parameter 这种变数是为了暂存某些 jasm code，is\_print 这种变数是为了在“翻译”过程中提供区别目前状态的控制信号。

(d) 在 yacl.y 的第一部分，还增加了两个函数，合并了一个 clear funtion。lookup\_nodeth funtion 是为了在少条件的情况下查询 symbol table 变数，lookup\_const function 则是为了专门查询 symbol table 中的 const。

#### 3.2 yacl.y 第三部分 C 语言程式部分

新增了用 C 语言中的 FILE \*fpt 来保存生成的 jsam code。

新增了用 system() funtion 来执行 linux command 以便于修改和快速测试。

新生成的 jsam file 与.class file 名称都是 go\_test 。

新增了 symbol table 的 columns，主要是增加了 Fun\_index 来区分不同函数中的相同层次的相同名称 variable。