

编译原理

成员	学号
吴一汶	3017218110
侯雨茜	3017218092
田承霖	3017218105
刘焱宁	3017218100

1) 符号说明

单词符号	种类编码	助记符
关键字	1	\$KEYWORD
算符	2	\$FUNCTOR
界符	3	\$IDENTIFIERS
标识符	4	\$BOUNDARY
常数	5	\$CONSTANT
其他	6	\$UNKNOWN

2) 使用说明

- 测试的环境

Ubuntu 16.04

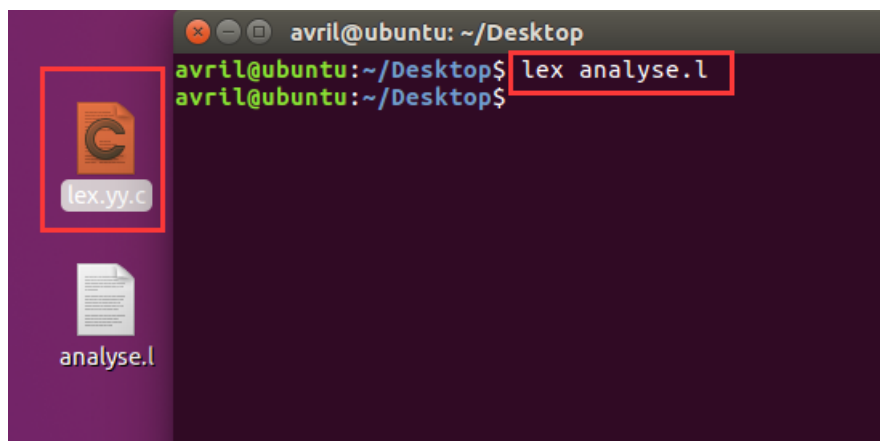
gcc 5.4.0

flex 2.6.0

- 操作说明

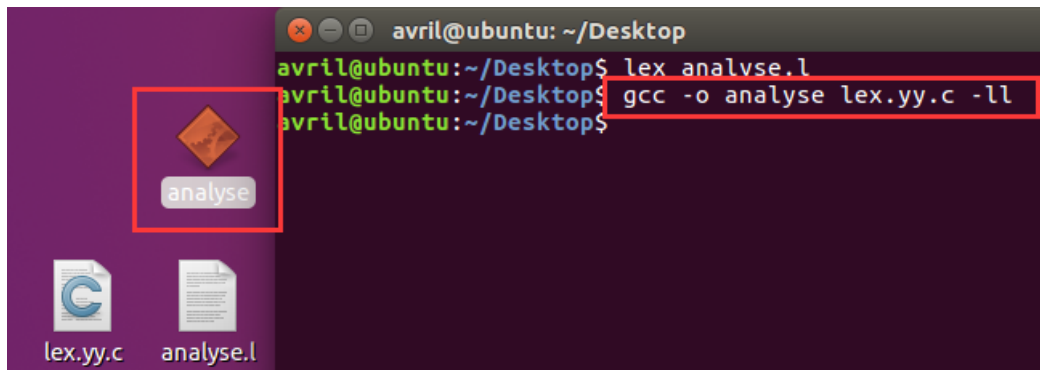
1. lex编译编写的 analyse.l，生成对应的lex.yy.c文件

```
lex analyse.l
```



2. gcc 编译 lex.yy.c 文件，生成词法分析器，需要增加-ll选项

```
gcc -o analyse lex.yy.c -ll
```



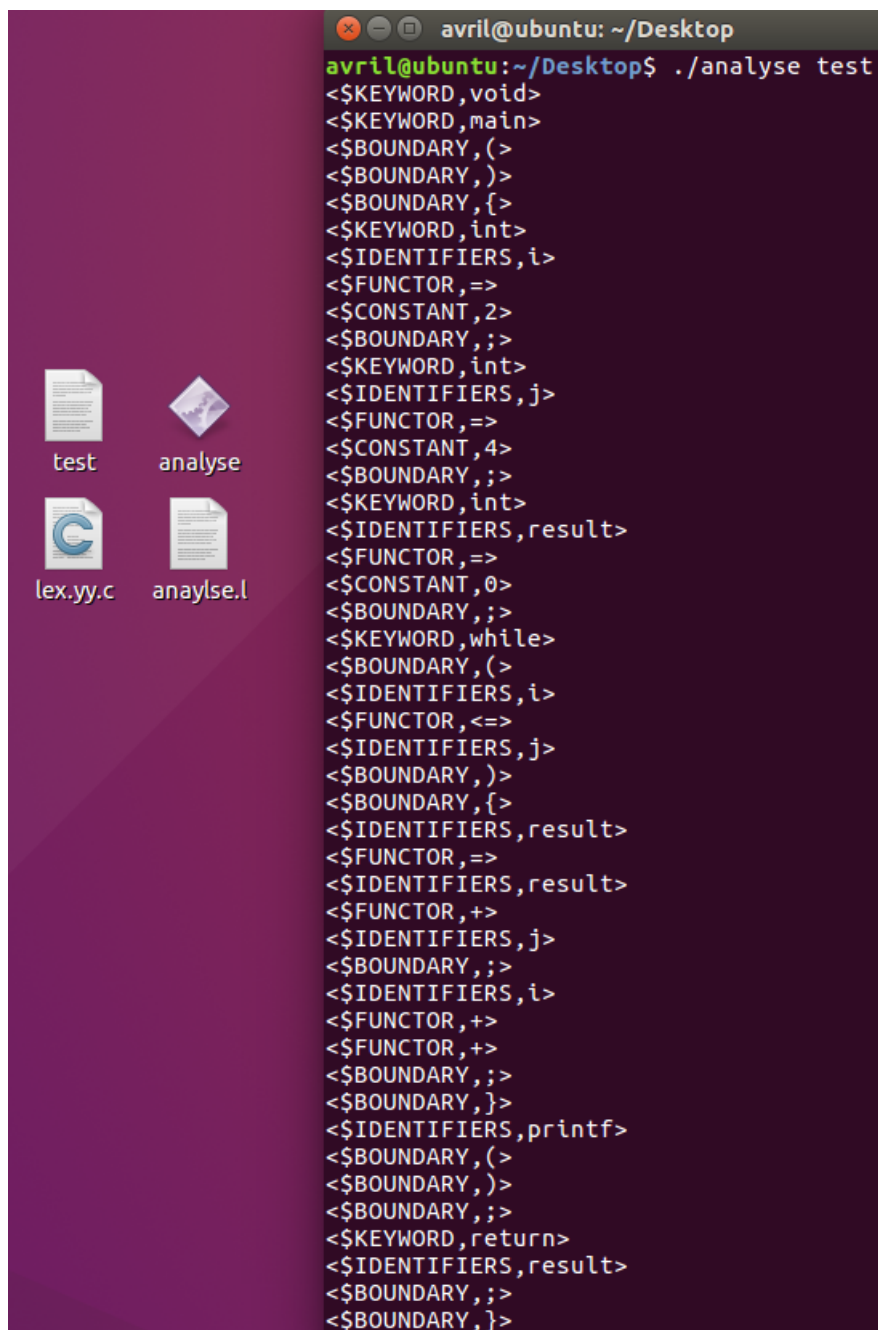
3. 新建测试文件test，输入测试样例

```
void main(){
    int i=2;
    int j=4;
    int result=0;
    while(i <=j){
        result=result+j;
        i++;
    }
    printf(); //注意,在 C- -中,这不是一个打印操作
    return result;
}
```



4. 使用词法分析器分析测试文件

```
./analyse test
```



```
avril@ubuntu: ~/Desktop
avril@ubuntu:~/Desktop$ ./analyse test
<$KEYWORD,void>
<$KEYWORD,main>
<$BOUNDARY,( >
<$BOUNDARY,) >
<$BOUNDARY,{ >
<$KEYWORD,int>
<$IDENTIFIERS,i>
<$FUNCTOR,=>
<$CONSTANT,2>
<$BOUNDARY,;>
<$KEYWORD,int>
<$IDENTIFIERS,j>
<$FUNCTOR,=>
<$CONSTANT,4>
<$BOUNDARY,;>
<$KEYWORD,int>
<$IDENTIFIERS,result>
<$FUNCTOR,=>
<$CONSTANT,0>
<$BOUNDARY,;>
<$KEYWORD,while>
<$BOUNDARY,( >
<$IDENTIFIERS,i>
<$FUNCTOR,<=>
<$IDENTIFIERS,j>
<$BOUNDARY,) >
<$BOUNDARY,{ >
<$IDENTIFIERS,result>
<$FUNCTOR,=>
<$IDENTIFIERS,result>
<$FUNCTOR,+>
<$IDENTIFIERS,j>
<$BOUNDARY,;>
<$IDENTIFIERS,i>
<$FUNCTOR,+>
<$FUNCTOR,+>
<$BOUNDARY,;>
<$BOUNDARY,>
<$IDENTIFIERS,printf>
<$BOUNDARY,( >
<$BOUNDARY,) >
<$BOUNDARY,;>
<$KEYWORD,return>
<$IDENTIFIERS,result>
<$BOUNDARY,;>
<$BOUNDARY,>
```

3) 其他说明

- 该词法分析器除了识别所给定范围的关键字、算符、界符、标识符、常数外，其他默认为无法识别 (Unknow)，其中关键字的范围是：while、break、continue、for、if、else、float、int、void、return、main；算符的范围是：+、-、*、/、=、>=、==、<=；界符的范围是：(、)、{、}、:；标识符，常数的定义与 C 语言相同。
- 对于注解中的内容，该词法分析器会忽略，不做分析。
- 涉及到的特殊情况：
形如 .3、3.、3.f、.3f、000003、.3e-2f 等都会被识别为常数
形如 _或多个_ 会被识别为标识符