

编译原理 实验一 实验报告

181250117

秦锐鑫

设计与实现

实现的功能

1. 附录 A 7.1.1 中给出的词法单元
2. 识别八进制和十六进制
3. 识别浮点数
4. 识别注释
5. 检查出词法错误
6. 有词法错误时只输出错误信息，没有错误时打印出所有词法单元

实现方式

正则表达式匹配

在 lexical.l 中进行正则表达式匹配

主要在于空白符的匹配

十进制无符号整型，八进制整型和十六进制整型的匹配

浮点型匹配，科学计数法浮点型匹配

单行注释和多行注释的匹配

```
WS [ \r\n\t\f\v]+
INT_DEC 0|([1-9][0-9]*)
INT_OCT 0([0-7]+)
INT_HEX 0[Xx][0-9A-Fa-f]+
FLOAT [0-9]+\.[0-9]+
FLOAT_E ([0-9]*\.[0-9]+|[0-9]+\.[0-9]*)[E|e][+-]?[0-9]+
COMMENT_LINE "//".*\n
COMMENT_MUL "/*"([^\*]|\\*)+[/\*])*\*+/"
```

两次遍历

在 main.cpp 中定义 int 型变量 error_flag 与 second_flag 都初始化为 0，然后进行遍历。

当 second_flag 为 0，即第一次遍历时，不打印正确的词法单元，若出现错误，则 error_flag 置为 1；若没有出现错误，则 error_flag 仍为 0。

第一次遍历结束后，没有出现错误，即 error_flag 仍为 0，则将 second_flag 置为 1，进行第二次遍历，打印出所有词法单元，否则直接结束。

印象深刻的bug

bug1 匹配优先级

转换规则部分（正则表达式 {动作}）

将 ID 放置在前面，会有警告

```
./src/lexical.l:95: warning, rule cannot be matched  
./src/lexical.l:104: warning, rule cannot be matched
```

bug2 换行符

不同系统下的默认换行符不同，linux中文件的换行符为 `/r/n`，而windows下文件的换行符为 `/n`。