

编译原理实验一实验报告

使用版本

1. GNU Linux Release: Ubuntu 22.04, 5.10.16.3-microsoft-standard-WSL2
2. GCC version: 11.4.0
3. GNU Flex version: 2.6.4
4. GNU Bison version: 3.8.2

程序功能及其实现

程序实现了以下功能：

1. 词法分析 能够检查出 `C--` 词法中未定义的字符以及任何不符合 `C--` 词法单元定义的字符，实现方式：依照手册内容加入必要的依赖文件和变量声明后，将对应的正则表达式输入到 `Flex` 源文件中即可。
2. 语法分析 能够构造出一个程序的语法树，并检查出 `C--` 语法中不符合 `C--` 语法定义的语句，并且在发现错误之后进行错误恢复，实现多个语法错误的检测，实现方式：依照手册内容将对应的文法输入到 `Bison` 源文件中，并且消除二义性与冲突，然后在适当位置加入 `error` 符号并且修改相应的 `yyerror()` 函数即可。
3. 过滤注释 能够识别 `//` 和 `/*...*/` 形式的注释，并且在其不符合定义的时候可以对其给出相应的错误提示，实现方式：如果在词法分析过程中，如果程序检测到 `//` 就会不断通过 `input()` 获取字符直到读取到换行符为止。如果程序检测到 `/*` 就会不断通过 `input()` 获取字符，直到读取到 `*/` 为止。
4. 打印语法树 如果被分析的程序没有语法和词法错误，将会在分析结束后按照先序遍历的方式打印出对应的语法树。实现方式：在词法分析和语法分析生成相应树的结点并保存其相应的值、类型、位置等信息，语法分析结束后再对树进行先序遍历并依据手册要求打印相应内容。

程序如何编译

编译指令如下，最后得到 `parser` 即为目标程序

```
flex lexical.l
bison -d syntax.y
gcc main.c syntax.tab.c SyntaxTree.c -lf1 -o parser
```