
编译原理 — Lab 1

姓名：郑樊巍

学号：171860658

日期：2020 年 2 月 23 日

1 基本功能

本次实验完成了基本要求和要求 1.1, 要求 1.2, 要求 1.3。程序通过读入代码文件，进行词法分析和语法分析后，对存在词法错误或语法错误的文件，会将报错信息输出（stderr）；对于词法语法正确的文件，输出其 AST(Abstract Syntax Tree)。其中，词法和语法的定义遵从 C- 文法。

对于正确的代码文件，程序输出的 AST 完全符合输出要求。其中，十/八/十六进制数被正确的识别，浮点数和科学计数法被正确的计算，注释被正确的处理。运算优先级正确，同时保证了不存在语法冲突。

对于存在错误的文件，程序按照要求输出错误信息和行号。值得一提的，程序详细的区分了不同类别的错误，给用户人性化的错误提示。为了防止错误信息冗杂，程序限制每行只允许报一个错误，且词法错误优先于语法错误。

对于 A 类错误（词法错误），具体有如下几类

- 基本功能：Mysterious character ” ”
- 基本功能：Illegal id；非法命名变量
- 要求 1.1：Illegal octal/hexadecimal number；检测到错误的八/十六进制数
- 要求 1.2：Illegal float point number；检测到错误的浮点数或科学计数法表达，错误的科学计数法表达**不包括**可能的 DOT 语法（如 struct1.e-2）
- 要求 1.3: Unterminated comment；块注释未结束，对于嵌套块语句，在语法分析给出错误信息。

对于 B 类错误（语法错误），除了给出错误行号以外，程序还会额外出现错误的具体地点（Unexcepted near ”xxx”）。通过错误恢复，尽可能识别以下几类错误；对于难以鉴别具体类型的错误，程序依然会给出最简单的 Syntax error 报错。

- Missing ”]/)”: 数组定义错误，函数定义/调用错误可以较好的检测
- Missing ”;”: 缺失分号（这种错误多被报为其它错误类型）
- Something wrong in global definition: 全局变量/函数定义错误
- Something wrong in definition of struct: 结构定义错误
- Something wrong before ”,”: 逗号前存在错误

- Wrong statements in block: 语句块中存在错误
- Something wrong in local definition: 局部定义错误
- Initial value not given: 定义赋值未给初值
- Expression error: 表达式错误

2 编译方法

实验环境为

- GNU Linux Release: Ubuntu 16.04, kernel version 4.15.0-45
- GCC version 5.4.0;
- GNU Flex version 2.6.0
- GNU Bison version 3.0.4

编译方法: 通过框架代码的 Makefile 生成二进制文件 parser。

3 实验亮点

除了实现基本功能和附加的三个要求之外, 在本次实验中还有如下亮点:

1. 错误处理详尽妥当: 可以在基本内容中看到, 本次实验对各种错误类型支持较好。
2. 代码结构清晰, 可维护性好; 同时通过设计的测试代码的测试: 代码中有详细的注释, 遵循一定格式规范, 接口设计妥当。

```

Lab1
├── .git.....git 相关文件
├── .gitignore
├── Code
│   ├── Makefile
│   ├── comman.h.....定义了测试用的宏等内容
│   ├── main.c.....主入口
│   ├── lexical.l
│   ├── syntax.y
│   ├── error.h.c.....错误时调用
│   └── ast.h.c.....AST 相关
├── Test.....内含多个测试文件, 分别测试不同功能
├── parser.....二进制文件
├── report.....您正在阅读的报告
└── README
  
```