LAB1 实验报告

1. 文件结构

---lexical.l 词法分析
---main.c 传入参数，读取并且打印
---Makefile
---syntax.y 语法分析
---tree.c 语法树的实现
---tree.h 语法树声明

2. 语法单元或者词法单元数据结构

```
typedef struct node{
    int line; //行号
    char* name; //单元名称
    NodeType type; // 单元类型
    union {
        unsigned int intval;
        float floatval;
        char str[maxline];
    }; //词法单元的对应值或者其余对应的字符串
    struct node* child;//孩子节点
    struct node* next;//兄弟节点
} Node;
```

3. 词法分析

对于每一个被识别的词法单元，都存入 yyval 的 node 成员之中，调用 newtoken 创建一个新的 Node*成员，并且返回对应的成员。

4. 语法分析

在每一个表达式后紧跟一个功能式，采用 create 树节点的方式一步步建立树的结构，默认的方式是创建节点，将产生式右边所有的节点添加为该节点的孩子，如果发生错误处理直接调用 yyerror。

5. 错误恢复

词法错误：
不符合定义的符号，直接出错
不合法的浮点数或者八进制十六进制数
语法错误：
（）与【】的匹配同步问题
主要是 Exp 的表达式出错问题
{}的匹配问题
;的处理问题

6. 问题及感想

词法的识别、正则表达式的错误导致程序识别方面的错误
错误处理不周全，%locations 的位置添加问题，重写 yyerror 出现的错误。

感想：错误恢复可能性太多，实现起来不太严谨，不方便使用。
Bison 的很多变量没有说明明白，需要自己都源代码查看含义，两个工具组合使用很微妙。