词法分析与语法分析

程序功能(必做+选做1.1)

数组访问错误处理

对于实验说明中，

> 输入文件的同一行中保证不出现多个错误，你的程序需要将这些错误全部报告出来，每一条错误提示信息在输出中单独占一行。

程序对数组访问错误的处理方式和讲义中稍有不同。

```
int main()
{
    float a[10][2];
    int i;
    a[5,3] = 1.5;
    if (a[1][2] == 0) i = 1 else i = 0;
}
```

程序将报告三个错误，其中第4行有两个错误，一个是“]”缺失，一个是语句不合理。

```
Error type B at Line 4: Missing "]".
Error type B at Line 4: Unvalid statement.
Error type B at Line 5: Missing ";".
```

这是为错误恢复提供便利，如果单纯将错误认定为“]”缺失，程序将无法处理如下输入文件，因为不能确保任何“]”缺失都可以有相应的恢复策略。

```
int main()
{
    int a[10];
    int b[10];
    int c;
    c = b[a[1];
}
```

输出：

```
Error type B at Line 6: Missing "]".
```

C--语言和C语言的文法差异处理

对于C--语言中没有的标识符 `for`，`double` 等进行了恢复处理，对输入文件中类似的标识符进行识别，如下。

```
int main()
{
    int float = 1;
    float 1err = 1.0;
}
```

输出：

```
Error type B at Line 3: Unvalid name of variable.
Error type B at Line 4: Unvalid identifier.
```

实现思路

文件组织

```
Code目录下：
.
├── Makefile
├── lexical.l
├── main.c
├── parse.c
├── parse.h
└── syntax.y
```

数据结构

定义 `TreeNode` 类型结构体，建立语法多叉树。

```
typedef struct TreeNode {
    char name[32];  //词法单元
    char text[32];  //词素
    int lineno;     //行号
    int childsum;   //子节点数目
    struct TreeNode* child[MAX_CHILD_NUM];
} Node;
```

主要功能函数如下。

```
Node* createNode(char*, char*);
void insertNode(int, Node*, ...);
void printNode(char*, char*);
void printTree(Node *, int);
```

实现过程

1. **检验词法分析程序** 根据C--文法构造相应的正则表达式，并额外定义了常见错误形式的正则表达式(主要针对C--和C语言的文法差异)，充分利用了正则表达式的匹配规则，权衡了正确性了代码的可读性，如下。

    - 不合法的十进制
    - 不合法的八进制
    - 不合法的十六进制
    - 不合法的单精度浮点数
    - +=，-=，/*，*/，//

2. **添加文法的产生式** 根据C--语言文法添加产生式，并通过 `%left` 、 `%right` 和 `%nonassoc` 以及它们的顺序对终运算符的优先级(precedence)以及结合性(associativity) 进行规定，

3. **恢复解决文法歧义** 悬空else问题的解决方案参照讲义中增加 `LOWER_THAN_ELSE` 算符完成，其他二义文法均使用词法单元error进行错误恢复处理。

4. **构建并打印语法树** 先序遍历多叉树，按照讲义要求递归打印语法树节点。

代码风格

- 把功能的底层实现封装到函数中，增加代码的可读性。
- 开启宏定义 `#define PRINT_DEBUG` 进入调试模式，可以额外打印结点的词法单元类型，并在语义恢复之后输出语法树(即使存在语法错误)。

写在最后

和室友聊天说起编译原理实验的讲义，我们都觉得相对PA和OS，这本指导手册的可读性更高一些。 坦白的说，之前的两个实验使我对稍具规模的实验任务产生了自然而然的畏惧感。 不过欣喜的是，我感受到了实验任务和理论课程的直接联系非常紧密，这使我在完成实验一的过程中有不小的成就感。

最后非常感谢助教学长和许畅老师对我的问题详尽的解答！