

L2实验报告——代码REVIEW

实现的功能：

- 将输入的.cmm文件进行语法分析
- 若文件中的代码没有语法错误，则输出语法树
- 如果出现词法错误，则只输出语法错误的信息。

实现方法：

- 在Lab1的基础上，修改lexical.l中的代码，为每次匹配添加建立当前词法单元node的动作
- 补充syntax.y文件代码
- 根据cmm语法，在syntax中添加文法定义，并添加对应的动作来建立语法树，并添加错误恢复以输出错误信息
- 语法树详解
 - Node* createNode(char* name, char* text)
 - 分配内存空间，建立node，node信息包含name(语法单元、词法单元名称)、text(yytext)、lineno(yylino)、*parent(父节点，初始化时设为NULL)、*child[] (子节点，初始化时设为NULL)、childsum(子节点个数，初始化时设为0)
 - void addChild(int childsum, Node* parent, ...)
 - 添加子节点，因为文法中的子节点个数不固定，这里采用可变参数，使用va_start获取可变参数列表的第一个参数地址，循环使用va_arg获取参数值并赋值给parent node 的 child node
 - 子节点添加完后补充父节点的lineno(child[0]->lineno)和childsum(函数第一个参数 childsum)
 - void printTree(Node *parent, int blank)
 - 打印语法树
 - blank用于输出空格，每次调用+2
 - 如果childsum!=0，说明是语法单元。从父节点开始，先打印父节点信息，然后根据 childsum循环调用printTree递归地打印child节点的信息
 - 如果childsum==0，说明是词法单元。根据要求
 - 如果当前结点是一个词法单元，则只要打印该词法单元的名称，而无需打印该词法单元的行号。
 - 如果当前结点是词法单元ID，则要求额外打印该标识符所对应的词素；
 - 如果当前结点是词法单元TYPE，则要求额外打印说明以该类型为int还是float；
 - 如果当前结点是词法单元INT或者FLOAT，则要求以十进制的形式额外打印该数字所对应的数值；
 - 词法单元所额外打印的信息与词法单元名之间以一个冒号和一个空格隔开。

分别处理打印信息

有趣现象

- 在错误恢复中，如果error的位置不恰当，会导致输出的错误信息变多或变少或不变，且行号会发生变换

一些bug

- 还是error的问题，因为没有测试用例，所以debug的时候不知道从哪下手，用计科的测试用例debug到心态爆炸(也跟当时对error的机制不理解，基本在瞎写，后来老师讲了后就好了)
- 优先级的问题，减号和取负要有区分
- 建node的时候name不要写错
- error加错了会导致语法树输出不了
- ...