

实验三 词法分析

17计二许红凯 320170941570

任务介绍:

根据给定源语言的构词规律，从任意字符串中识别出所有的合法单词符号，并以等长的二元组形式输出

输入:

字符串形式的源程序

输出:

单词符号所构成的串(流)，单词以等长的二元组形式呈现

题目:

设计一个程序，根据给定源语言的构词规则，从任意字符串中识别出该语言所有的合法的单词符号，并以等长的二元组形式输出。注意：

1. 附录中介绍了一个基于C语法规则设计的源语言LittleC和一个基于Palcal语法规则设计的源语言LittleP,可以作为参考
2. 学生可以自行挑选或设计榆中源语言，以此为基础来完成本实验和后续实验。该语言的实现应该满足附录B的要求
3. 该程序应该设计为至少2个模块：驱动模块和工作模块。驱动模块包含了程序的入口和出口，主要负责输入、输出处理并调动工作模块;工作模块负责具体的分割、识别、归类等工作。这样做的好处是只要模块间的接口(需要传递哪些数据，数据的结构)设计合理,后续实验中做语法分析器时就可以直接调用此处的工作模块，而不需要改动太多代码

代码思路:

1. 分为5个类别：关键字、标识符、分隔符、运算符、常量
2. 输入字符串形式的源程序后，按照每个字符判断是否为分隔符、运算符、常量
3. 对于关键字和标识符，先识别为合法字符串，再判断是否为关键字，若不是则为标识符

程序功能说明

1. 根据给定源语言的构词规律，从任意字符串中识别出所有的合法单词符号，并以等长的二元组形式输出。例如：输入 {int a = 1; }，输出（分隔符 {）（关键词 int）（标识符 a）（运算符 =）（常量 1）（分隔符 ;）（分隔符 }）
2. 识别若干个（或一个）连续字符构成的合法字符串，再判断该字符串为“关键字”还是“标记符”
3. 识别一个字符是否为“分隔符”
4. 识别一个字符是否为“运算符”
5. 识别一个字符是否为“常量”

重点说明

1. 在识别“分隔符”、“运算符”、“常量”时，只需对当前一个字符进行判断识别。例如：

```
//判断是否为“运算符”
int MatchCountChar(char *str,int i){
    for(int j = 0; j < 8; j++){
        if(str[i] == countChar[j]){
            printf("%s %c\n",name[3],str[i]);
        }
    }
    return i;
}
```

2. 在识别“关键字”和“标识符”时，需先判断连续若干个（或当前一个）字符是否为合法字符串，例如：在 {int a = 1; } 中先识别出合法字符串int。再判断该合法字符串是否为“关键字”，否则为“标识符”。最后再进行输出。

```
char str1[30];
for(int m = 0; m < strlen(str1); m++){
    str1[m] = '\0';
}

int j = 0;
//判断首字母是否符合
if( (str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z') || str[i] == '_'){
    str1[j] = str[i];
    //判断后续字符
    while( (str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z') || str[i] == '_'){
        str1[j++] = str[i++];
    }
    str1[j] = '\0';
}
```

//判断一个合法字符串是否为“关键字”

```
int matchKeyword(char *str){
    for(int i = 0; i < 32; i++){
        //是关键词
        if( !strcmp(str,keyword[i]) ){
            return 0;
        }
    }
    return 1;
}
```

//是关键字

```
if(j && matchKeyword(str1)){
    printf("%s %s\n",name[1],str1);
}
//是标识符
else if(j && !matchKeyword(str1)){
    printf("%s %s\n",name[0],str1);
}
```