

实验一 预处理

17计二许红凯 320170941570

任务介绍：

在词法分析之前，对程序员提交的源程序进行预处理，剔除注释等不必要的字符，以简化词法分析

输入：

字符串形式的源程序

输出：

处理之后的字符串形式的源程序

题目：

设计一个程序，从任意字符串中剔除C语言形式的注释，包括：

- 1.形如：//...的单行注释；
- 2.形如：/*...*/的多行注释

代码思路

1. 代码有三种状态：正常、注释//、注释/**-/
2. 当前默认为正常代码状态，直到读入/后进行判断下一字符
3. 注释//到换行结束，注释/*到*/结束

程序功能说明

1. 从任意字符串中剔除C语言形式的注释，包括//注释和/**/注释
2. 识别//作为注释的开始，直到识别出换行符\n作为注释的结束
3. 识别/* 作为注释的开始，直到 */作为注释的结束
4. 从文件中一个个读取字符，输出时一个个向文件输出字符

重点说明

1. 用tag = 0,1,2表示当前识别过程中的三种代码状态，分别为：正常代码部分，//注释部分，/**/注释部分。对于tag = 0的字符，进行输出。

```
//当前为读入正常代码
if(tag == 0){
    if(temp1 == '/'){
        temp2 = fgetc(in);

        //下一读入部分为/**/注释
        if(temp2 == '*'){
            tag = 2;
        }
        //下一读入部分为//注释
        else if(temp2 == '/'){
            tag = 1;
        }
        //读入为正常代码状态时，则一个个输出
        else{
            fputc(temp1, out);
            fputc(temp2, out);
        }
    }
    else{
        fputc(temp1, out);
    }
}
```

2. 默认当前为正常代码状态即tag = 0。对于下一个读入的字符，判断是否有可能为注释。当一个注释被识别结束后，恢复tag = 0即默认状态，再去读下一个字符去进行判断识别。

```
//当前为读入/**/注释，直到读入*/表示注释结束，重新开始读正常代码
else{
    if(temp1 == '*'){
        temp2 = fgetc(in);
        if(temp2 == '/'){
            tag = 0;
        }
    }
}
```

3. 对于//注释，以换行符\n结束，但是剔除该注释时若本身有换行，则正常代码输出时也需要有换行操作，故对该\n也要输出

```
//当前为读入//注释，直到读入换行表示注释结束，重新开始读正常代码
else if(tag == 1){
```

```
        if(temp1 == '\\n'){
            tag = 0;
            fputc(temp1,out);
        }
    }
```