

单词检索统计系统

问题描述

给定一个文本文件，要求统计给定单词在文本中出现的总次数，并检索输出某个单词出现在文本中的行号、在该行中出现的次数以及位置。

项目要求

本项目的设计要求可以分为三个部分实现：其一，建立一个文本文件，文件名由用户用键盘输入；其二，给定单词计数，输入一个不含空格的单词，统计输出该单词在文本中的出现次数；其三，检索给定单词，输入一个单词，检索并输出该单词所在的行号、该行中出现的次数以及在该行中的相应位置。

功能实现及代码分析

1、进入项目

进入项目，引导用户输入新建文件，进行单词词频统计或检索。在无法找到用户指定文件时，报错并退出操作。支持用户在未创建新文件的情况下对已存在文件进行词频统计和单词检索。

```
int main(void) {
    cout << "***          单词检索统计系统          ***" << endl;
    cout << "===== " << endl;
    cout << "***          请选择要执行的操作：          ***" << endl;
    cout << "***          1.新建文件          ***" << endl;
    cout << "***          2.单词计数          ***" << endl;
    cout << "***          3.单词检索          ***" << endl;
    cout << "***          4.退出程序          ***" << endl;
    cout << "===== " << endl;

    while (1) {
        cout << "\n请输入要执行的操作： ";
        int ops;
        cin >> ops;
        switch (ops) {
            case 1: { //新建文件
                cout << "请输入要新建的文件名： ";
                string fileName;
                cin >> fileName;
                create(fileName);
                break;
            }
            case 2: { //单词计数
                cout << "请输入要打开的文件名： ";
                string fileName;
                cin >> fileName;
                openFile.open(fileName);
                if (openFile.is_open()) { //判断文件是否正常打开
                    count(); //统计单词
                    openFile.close();
                } else {
                    cout << "找不到这个文件！ " << endl;
                }
                break;
            }
            case 3: { //单词检索
                cout << "请输入要打开的文件名： ";
                string fileName;
                cin >> fileName;
                openFile.open(fileName);
                if (openFile.is_open()) { //判断文件是否正常打开
                    query(); //检索单词
                    openFile.close();
                } else {
                    cout << "找不到这个文件！ " << endl;
                }
                break;
            }
        }
    }
}
```

```

        case 4: { //退出程序
            exit(0);
        }
        default:
            cout << "请输入正确的操作!" << endl;
            break;
    }
}
}
}

```

2、新建文件

新建文件，逐行输入内容，以#结束输入。

```

void create(string fileName) {
    cout << "输入#结束" << endl;
    ofstream fout(fileName);
    while (1) {
        cout << "请逐行输入文本: ";
        char t = getchar();
        if (t == '#') break; //输入终止条件
        string line;
        getline(cin, line);
        fout << line + '\n';
    }
    fout.close();
}

```

3、单词计数

根据用户输入确定要查询的单词，逐行读入，将所有单词通过split函数切分后剔除','和'.'并放入wordTable单词表中，然后遍历单词表统计要查询的单词，输出统计结果。

```

void count() {
    cout << "请输入要检索的单词: ";
    string word;
    cin >> word;

    string line; //每次读入一行句子
    int wordCount = 0, iCount = 0; //所有单词量
    while (getline(openFile, line)) {
        vector<string> wordTable;
        split(line, wordTable, ' ');
        wordCount += wordTable.size();
        for (auto i:wordTable) {
            if (i[i.size() - 1] == '.' || i[i.size() - 1] == ',') { //去除末尾','和'.'
                i.erase(i.end() - 1);
            }
            if (i == word) {
                iCount++;
            }
        }
    }

    cout << "单词 " << word << " 出现了" << iCount << "次" << endl;
}

```

4、单词检索

通过find函数逐行检索所要检索的单词，并返回位置。在处理单词时，去掉结尾的空格、'\n'、','和'.'。

```

void query() {
    cout << "请输入要检索的单词: ";
    string word;
    cin >> word;
    string line;
    int iCount = 1, lineNum = 1, pos = 0;
    while (getline(openFile, line)) { //逐行检索
        while ((pos = line.find(word, pos)) != string::npos
            && (line[pos + word.size()] == ' ' || line[pos + word.size()] == '\n'
            || line[pos+word.size()]=='.' || line[pos+word.size()]==',')) {

```

```

        cout << "单词 " << word << " 第" << iCount++ << "次出现在第"
            << lineNum << "行第" << pos << "列。" << endl;
        pos += word.size();
    }
    lineNum++;
    pos = 0;
}
cout << "单词 " << word << " 总共出现" << --iCount << "次。" << endl;
}

```

5、辅助函数

获取单词

通过快慢标记的方法，分离出每行中的单词，并将其存入wordTable，即单词总表。

```

void split(const string &s, vector<string> &wordTable, char temp) { //分词
    int first, second = 0;
    first = s.find(temp);
    while (string::npos != first) {
        wordTable.push_back(s.substr(second, first - second));
        second = first + 1;
        first = s.find(temp, second);
    }
    if (second != s.length()) { //找到单词
        wordTable.push_back(s.substr(second));
    }
}

```

用例演示

```

**          单词检索统计系统          **
=====
**          请选择要执行的操作：        **
**          1.新建文件                  **
**          2.单词计数                  **
**          3.单词检索                  **
**          4.退出程序                  **
=====

```

```

请输入要执行的操作： 1
请输入要新建的文件名： test.txt
输入#结束
请逐行输入文本： Google is everywhere.
请逐行输入文本： It is on phones, TVs, and is even inside cars.
请逐行输入文本： It is now taking a big step to expanding its Assistant to even more hardware today.
请逐行输入文本： You can ask it what time it is.
请逐行输入文本： #

```

```

请输入要执行的操作： 2
请输入要打开的文件名： test.txt
请输入要检索的单词： is
单词 is 出现了5次

```

```

请输入要执行的操作： 3
请输入要打开的文件名： test.txt
请输入要检索的单词： is
单词 is 第1次出现在第1行第7列。
单词 is 第2次出现在第2行第2列。
单词 is 第3次出现在第2行第25列。
单词 is 第4次出现在第3行第2列。
单词 is 第5次出现在第4行第27列。
单词 is 总共出现5次。

```

```

请输入要执行的操作： 4

```

```

Process finished with exit code 0

```