## 基于c++的统计输入字符串中单词个数程序

#include <iostream>

#include <string>

#include <cstring>

using namespace std;

int find(char (\*ss)[20], char s[], int m)

{

int flag = -1; //要返回的标志符初始为-1

for(int i = 0; i < m; i++) //传入了行号，每个都要比较

{

if(strcmp(ss[i], s) == 0)

flag = i;

}

return flag;

}

int main()

{

char str[256] = {0}; //用于接收用户输入的字符串

char s[20][20] = {0}; //用于储存每一个不同的单词

int i = 0; //字符串下标

int m = 0; //行

int n = 0; //列

char temp[20] = {0}; //用于临时储存字符串中方每个单词

int len = 0; //记录单词的长度

int count[20] = {0}; //记录每个单词出现次数

int k; //单词每个字符下表

int flag; //判断能否找到

gets(str);

while(str[i] == ' ') //作用是把开头的空格给过滤掉

{

i++;

}

while(str[i] != 0) //外层循环判断str是否结束

{

k = 0; //temp长度清零

flag = -1; //是否找到相同单词打标志位重置

while(str[i] != ' '&& str[i] != 0) //每一个非''位存入temp

{

temp[k] = str[i];

k++; //每次加一，用于计算单词长度

i++; //一个单词都录完后，i下标的str处为空

}

if(str[i-1] != ' ') //一个单词录完后str[i]的值是空，find函数可以在s[][]中找到，会出现错乱

{

flag = find(s, temp, m); //找到单词为单词在s[][]中的行号，否则为-1

if(flag == -1)

{

strcpy(s[m], temp); //若是-1就在s[][]中加入新单词

for(int j = 0; j < k; j++) //要清空，不然前面长的单词会影响后面的短的单词

{

temp[j] = '\0';

}

count[m] = 1; //count[]要记录单词出现一次

m++; //s[][]行号加一

}

else

{

strcpy(s[flag], temp); //和上面没有找到类似

for(int j = 0; j < k; j++)

{

temp[j] = '\0';

}

count[flag] += 1;

}

}

else

{

break;

}

while(str[i] == ' ') //作用是把中间和最后的空格给过滤掉

{

i++;

}

}

cout << "文件读入有(两竖线中间内容)：\n|";

for(int ii = 0; ii < i; ii++)

{

cout << str[ii];

}

cout << "|\n此文本统计有：\n单词";

cout.width(21); //格式输出

cout << "个数\n";

for(i = 0; i < m; i++)

{

for(int j = 0; j < strlen(s[i]); j++)

{

cout << s[i][j];

}

cout.width(21 - strlen(s[i]));

cout << count[i] << endl;

}

cout << "统计完毕！";

return 0;

}