




1109 综合实验：文件操作与字符处理

原创

红烧春鸽  于 2013-12-14 17:01:52 发布  6935  收藏 31

分类专栏: [SCAU_OJ](#) 文章标签: [c](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/samlee946/article/details/17320735>

版权



[SCAU_OJ](#) 专栏收录该内容

4 篇文章 2 订阅

订阅专栏

1109 综合实验：文件操作与字符处理

时间限制:4000MS 内存限制:65536K

提交次数:6265 通过次数:1646

题型: 填空题 语言: 无限制

Description

在当前目录中存在文件名为"case1.in" (其中case后为数字1, 不是字母1, 写错提交后会判错) 的文本文件, 其内容为一篇英文文章 (以EOF作为结束标志)。现要求读取该文本文件内容, 统计文章中每个单词出现的次数, 并输出出现次数最多的前5个单词及其出现次数 (按出现次数由多到少的顺序输出, 次数相同时按字典顺序输出, 不足5个单词时, 按序输出全部单词)。程序中注意如下细节:

- (1) 空格、标点符号与回车符起到分隔单词的作用。
- (2) 文章一行的末尾可能有连字符, 出现连字符时, 该行最末的字符串与下行最先出现的字符串构成一个单词;
- (3) 名词缩写算一个单词;
- (4) 数字不算单词;
- (5) 单词不区分大小写;
- (6) 输出时单词全使用小写;

```
#include "stdio.h"
#include "math.h"
#include "string.h"
#include "stdlib.h"
```

```
main()
{
}
}
```

输入格式

文件case1.in中一篇英文文章, 包含多段文字, 单词数不超过10000, 每个单词不超过20个字符

输出格式

按题意输出答案

输入样例

(如case1.in内容如下)

```
I am a student. My school is SCAU. It is a beautiful university. I like it.
```

输出样例

```
a 2
i 2
is 2
it 2
am 1
```

作者

admin

这题虽然不难,但是还是挺麻烦的,一开始看漏题目,用fscanf来读取字符串,看到连字符的条件就傻了,后来把原来的代码都删了用fgetc来读取,然后好不容易过了第一个样例,在第二个样例时WA了,-t-t-t-t这个情况没考虑到,然后考虑进去以后一提交居然超时了...orz实验题都超时了...后来乱改一通,代码的可读性下降了许多,最终找到了原因:题目说单词字符数不超过20,我定义了100,然后应该是初始化时耗费了太多的时间,后来我修改为30就过了(表示提交了两次,一次是水过去的,只初始化了10000个,用时3.1s,后来初始化100000个时用时就变3.4s了...),你要改成20也行,改成20会更快

我的算法与别人的一对比,慢了许多,可能是循环用得太多的原因吧..

估计有偷懒的同学百度到了这里想看题解,我劝同学经过思考后再看,程序员重要的是独立思考..

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
struct s
{
    int time;
    char words[30];
}word[10010];
int main()
{
    FILE *fp;
    fp=fopen("case1.in","r");
    int count=0.p=0.i:      /*count is number of words.p is a point*/
```

```

for(p=0;p<10010;p++)
{
    word[p].time=1;
    memset(word[p].words,0,sizeof(word[p].words));
}
p=0;
char ch,ch1;
char temp[30];
while((ch=fgetc(fp))!=EOF)
{
    if(ch>='a'&&ch<='z')
        temp[p++]=ch;
    else if(ch>='A'&&ch<='Z')
        temp[p++]=ch+32;
    else if(ch=='-')
        {
            ch1=fgetc(fp);
            if(ch1=='\n')
                continue;
            else
                {
                    strcpy(word[count++].words,temp);
                    memset(temp,0,sizeof(temp));
                    p=0;
                    if(ch1>='a'&&ch1<='z')
                        temp[p++]=ch1;
                    else if(ch1>='A'&&ch1<='Z')
                        temp[p++]=ch1+32;
                }
        }
    else
        {
            strcpy(word[count++].words,temp);
            memset(temp,0,sizeof(temp));
            p=0;
        }
}
for(p=0;p<count;p++)
{
    for(i=0;i<count;i++)
        if(i!=p&&strcmp(word[p].words,word[i].words)==0)
        {
            if(p>i)
            {
                word[i].time++;
                memset(word[p].words,0,sizeof(word[p].words));
            }
            else
            {
                word[p].time++;
                memset(word[i].words,0,sizeof(word[i].words));
            }
        }
}
for(i=0;i<count;i++)
{
    if(strcmp(word[i].words,"a")<0)

```

```

        word[i].time=0;
    }
    //用冒泡进行排序
    for(i=1;i<count;i++)
        for(p=0;p<count-i;p++)
        {
            if(word[p].time<word[p+1].time)
            {
                int t=word[p].time;
                word[p].time=word[p+1].time;
                word[p+1].time=t;

                memset(temp,0,sizeof(temp));
                strcpy(temp,word[p].words);
                memset(word[p].words,0,sizeof(word[p].words));
                strcpy(word[p].words,word[p+1].words);
                memset(word[p+1].words,0,sizeof(word[p+1].words));
                strcpy(word[p+1].words,temp);
            }
            else if(word[p].time==word[p+1].time)
            {
                if(strcmp(word[p].words,word[p+1].words)>0)
                {
                    memset(temp,0,sizeof(temp));
                    strcpy(temp,word[p].words);
                    memset(word[p].words,0,sizeof(word[p].words));
                    strcpy(word[p].words,word[p+1].words);
                    memset(word[p+1].words,0,sizeof(word[p+1].words));
                    strcpy(word[p+1].words,temp);
                }
            }
        }
    for(i=0;i<(count>5?5:count);i++)
    {
        if(word[i].time!=0) //test
            printf("%s %d\n",word[i].words,word[i].time);
    }

    fclose(fp);
    return 0;
}

```

原来的代码交上OJ貌似老超时..

优化了一下 用时在2.3s~3.1s,比起标程的530ms 我仍需努力啊...

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

```

```

#include <string.h>
struct s
{
    int time;
    char words[21];
}word[10010];
int main()
{
    FILE *fp;
    fp=fopen("case1.in", "r");
    int count=0,p=0,q=0,i;
    for(p=0;p<10010;p++)
    {
        word[p].time=1;
        memset(word[p].words,0,sizeof(word[p].words));
    }
    p=0;
    char ch,temp[21];
    while((ch=fgetc(fp))!=EOF)
    {
        if(ch>='a'&&ch<='z')
            word[count].words[q++]=ch;
        else if(ch>='A'&&ch<='Z')
            word[count].words[q++]=ch+32;
        else if(ch=='-')
            {
                ch=fgetc(fp);
                if(ch=='\n')
                    continue;
                else
                    {
                        count++;
                        q=0;
                        if(ch>='a'&&ch<='z')
                            word[count].words[q++]=ch;
                        else if(ch>='A'&&ch<='Z')
                            word[count].words[q++]=ch+32;
                    }
            }
        else
            {
                count++;
                q=0;
            }
    }
    for(p=0;p<count;p++)
    {
        for(i=0;i<count;i++)
            if(i!=p&&strcmp(word[p].words,word[i].words)==0)
            {
                if(p>i)
                {
                    word[i].time++;
                    memset(word[p].words,0,sizeof(word[p].words));
                }
                else
                {
                    word[p].time++;
                    memset(word[i].words,0,sizeof(word[i].words));
                }
            }
    }
}

```

```

    }
}
for(i=0;i<count;i++)
{
    if(word[i].words[0]<'a' || word[i].words[0]>'z')
        {
            word[i].time=0;
        }
}
for(i=1;i<count;i++)
    for(p=0;p<count-i;p++)
    {
        if(word[p].time<word[p+1].time)
        {
            int t=word[p].time;
            word[p].time=word[p+1].time;
            word[p+1].time=t;
            strcpy(temp,word[p].words);
            strcpy(word[p].words,word[p+1].words);
            strcpy(word[p+1].words,temp);
        }
        else if(word[p].time==word[p+1].time)
        {
            if(strcmp(word[p].words,word[p+1].words)>0)
            {
                strcpy(temp,word[p].words);
                strcpy(word[p].words,word[p+1].words);
                strcpy(word[p+1].words,temp);
            }
        }
    }
}
for(i=0;i<(count>5?5:count);i++)
{
    if(word[i].time!=0)
        printf("%s %d\n",word[i].words,word[i].time);
}

fclose(fp);
return 0;
}

```