程序设计基础训练课程实验二实验报告

作者：邱林

学号：17281214

目录

[项目一：统计单词个数 1](#_Toc509683156)

[项目二：文本统计 1](#_Toc509683157)

[对两个结构体类型 TEXTFILEINFO 和 FILESINFO 及其间关系的理解 1](#_Toc509683158)

[typedef 的作用 2](#_Toc509683159)

[英文单词 allocate 和 memory 的含义，函数 malloc 的具体功能 2](#_Toc509683160)

[CountWordsOfEuropeanTxtFile函数流程图 2](#_Toc509683161)

[函数 fopen 或 fclose 的具体功能 5](#_Toc509683162)

[四个子函数的填写 5](#_Toc509683163)

[第一次填写 5](#_Toc509683164)

[第二次填写 5](#_Toc509683165)

[第三次填写 7](#_Toc509683166)

[总结 8](#_Toc509683167)

[函数 fprintf 的功能 9](#_Toc509683168)

[项目三：利用链表对新词排序 9](#_Toc509683169)

[项目四：网址收藏夹 11](#_Toc509683170)

[函数sprintf的功能 11](#_Toc509683171)

[函数WinExec的功能 11](#_Toc509683172)

[为什么LabFunctions中要加#include "windows.h" 11](#_Toc509683173)

[参考资料 11](#_Toc509683174)

# 项目一：统计单词个数

代码及注释如图

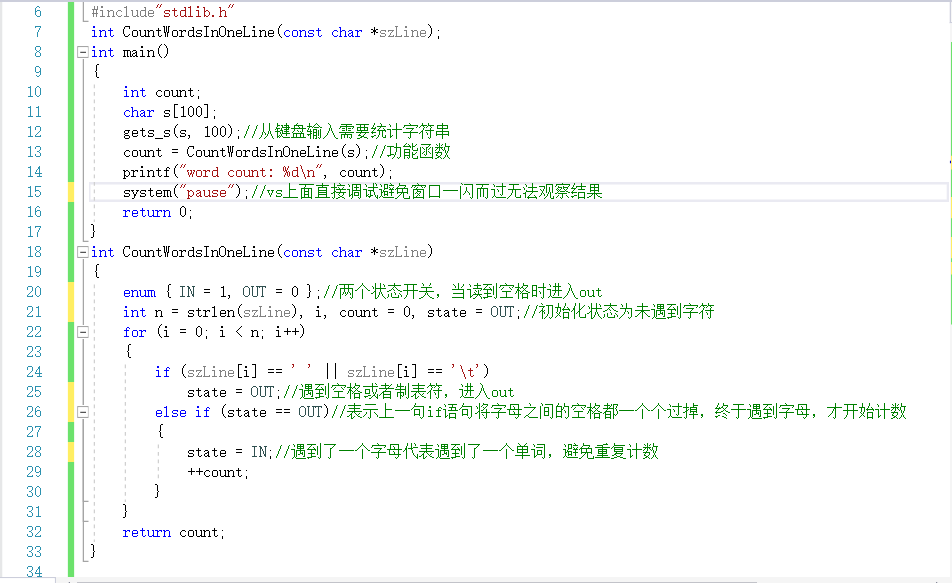


图1-1

# 项目二：文本统计

## 对两个结构体类型 TEXTFILEINFO 和 FILESINFO 及其间关系的理解

TEXTFILEINFO 是记录一个文本中各种字符的个数的结构体，包括用户结定的文件名，程序统计出单词数、文件大小、数字、空白、字母和其他字符的个数，定义一个TEXTFILEINFO 的数组就可以记录多个文本信息而不散乱。FILESINFO则是对所有文件信息的集合，里面包含了TEXTFILEINFO数组的首地址，也就是说，通过FILESINFO可以找到所有包含单个文本信息的TEXTFILEINFO结构体。同时TEXTFILEINFO中还记录了实验文件的数目，所有文件中新词的总数。

在此程序中，构建了FILEINFO类型的数组，这个数组作为TEXTFILEINFO结构体的成员。

## typedef 的作用

typedef为C语言的关键字，作用是为一种数据类型定义一个新名字。这里的数据类型包括内部数据类型（int,char等）和自定义的数据类型（struct等）。 在编程中使用typedef目的一般有两个，一个是给变量一个易记且意义明确的新名字，另一个是简化一些比较复杂的类型声明。Typedef 的为数据类型定义一个新名字不只是简单的宏替换。可以用作同时声明指针型的多个对象。

## 英文单词 allocate 和 memory 的含义，函数 malloc 的具体功能

Allocate的英文含义是to use something for a particular purpose, give something to a particular person etc, especially after an official decision has been made。中文意思可以翻译为分配。在C语言中为动态内存分配。

Memory的A computer's memory is the part of the computer where information is stored, especially for a short time before it is transferred to disks or magnetic tapes.中文意思可以翻译为内存。在C语言中为也为内存。

函数malloc是存储分配函数。用于申请一块连续的指定大小的内存块区域以void\*类型返回分配的内存区域地址，当无法知道内存具体位置的时候，想要绑定真正的内存空间，就需要用到动态的分配内存。

## CountWordsOfEuropeanTxtFile函数流程图

**定义变量 nWords用以计数**

**定义指针fp用以指向文件**

**定义数组carrBuffer用以储存每行的字符**

carrBuffer

**跳过/出循环**

**进入统计循环**

**读到文件末尾**

**未读到文件末尾**

**判断是否进入统计循环**

**Return -1**

**打开失败**

**成功打开**

**打开文件**

**将文件中一行的字符输入carrBuffer中**

**将carrBuffer传入子函数CountWordsInOneLine统计carrBuffer（文件中某行）中字符个数**

**Return nWord**

**关闭文件**

**CountWordsInOneLine的返回值加入nWords中**

## 函数 fopen 或 fclose 的具体功能

Fopen：用于打开文件，并且返回一个指向一个包括了文件管理有关信息的数据结构的指针。如果成功的打开一个文件, fopen()函数返回文件指针, 否则返回空指针。

Fclose：来关闭一个由fopen()函数打开的文件。该函数返回一个整型数。当文件关闭成功时, 返回0,否则返回一个非零值。

## 四个子函数的填写

### 第一次填写

一经编译调试，很快就出现问题。首先发现的问题是PrepareBasicInfoOfFiles将文件目录里面的文件名字导入pFileInfos[i].szFileName出了问题。可以观察到所有的pFileInfos[i].szFileName都是乱码（屯屯）。

### 第二次填写

随后我发现问题所在了，在PrepareBasicInfoOfFiles函数中，我先是统计了文件个数，统计完，指针已经值到了文件的末尾，我却没有重置指针位置。于是我在统计完文件个数后重置了位置指针的位置。但是再次运行后，又出现文件程序突然中止的问题（第一次调试的时候也出现了，但我都归结与上一个问题）。错误如图2-1：

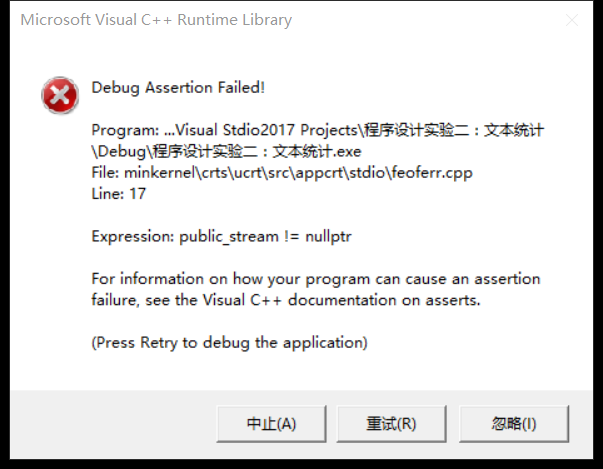


图2-1

我猜测这个是运行超时，超时在哪？我进行单步调试，发现在打开需要统计的文件时，这个框再次出现。再仔细一看，问题起源于我在调用fopen\_s的时候我没有给定正确的路径，稍加调整后给定了正确的路径可以顺利打开文件了。第二次修改如下：

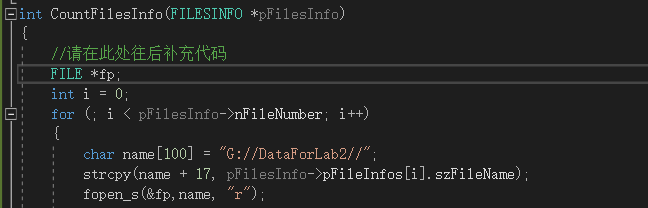


图2-2

运行结果如图：

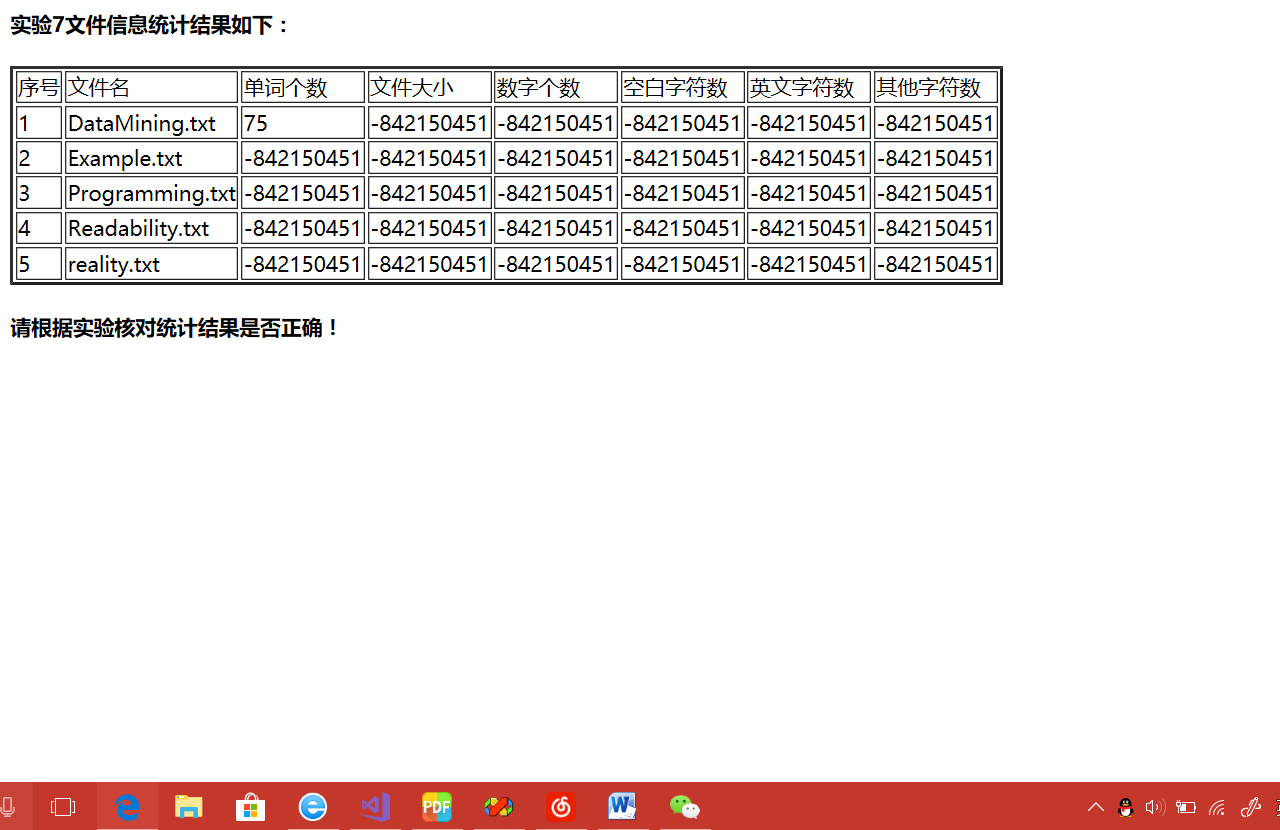


图2-3

可见，程序还是存在问题。

### 第三次填写

再次调试函数，图2-3出现的问题的原因第一个是函数CountFilesInfo中的return 0；位置不对，如图2-4。导致在循环第一次的过程中就返回了返回值，函数提前结束，后面的文件没有进行统计。第二个是CountDifferentCharactersInFile函数中的fopen\_s也没有给予正确的地址，如图2-5。

1. **int** CountFilesInfo(FILESINFO \*pFilesInfo)
2. {
3. //请在此处往后补充代码
4. **FILE** \*fp;
5. **int** i = 0;
6. **for** (; i < pFilesInfo->nFileNumber; i++)
7. {
8. fopen\_s(&fp,"pFilesInfo->pFileInfos[i].szFileName", "r");
9. **int** c, count = 0, state = 0;
10. **while** ((c = fgetc(fp)) != 0)//统计英文单词个数
11. **if** (c == ' ' || c == '\t' || c == '\n')
12. {
13. state = 0;
14. }
15. **else** **if** ((c >= 'a'&&c <= 'z') || (c >= 'A'&&c <= 'Z') || state == 0)
16. {
17. state = 1;
18. ++count;
19. }
20. pFilesInfo->pFileInfos[i].nWords = count;
21. fseek(fp, 0, SEEK\_SET);
22. CountDifferentCharactersInFile(pFilesInfo->pFileInfos + i);

25. **return** 0;
26. }
27. }

图2-4

1. **if** (fopen\_s(&fp,pTextFileInfo->szFileName, "r")!=0)

图2-5

### 总结

最后，将错误修改以后再稍加调整。得到正确的函数。代码详情见源文件。

运行结果如图2-6.

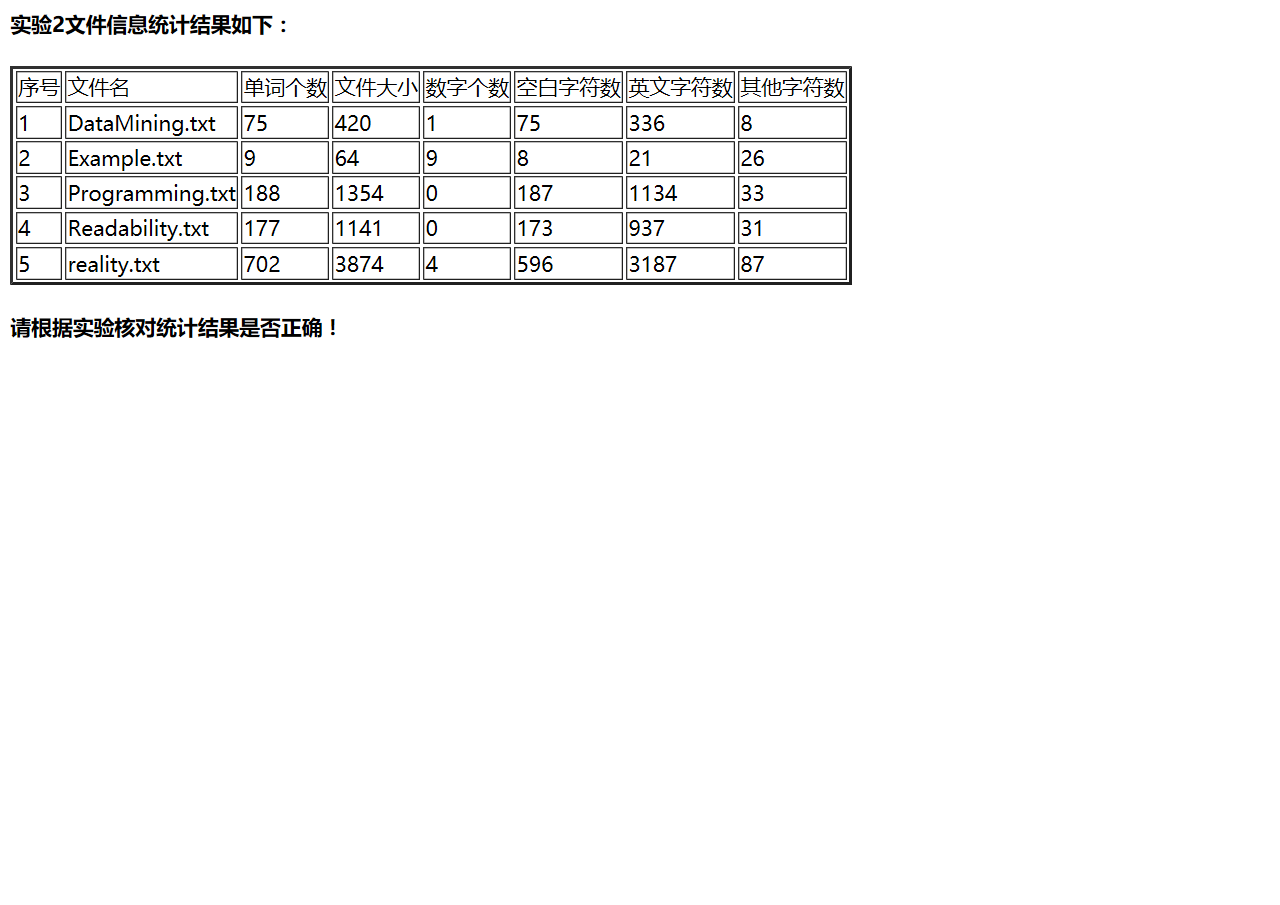


图2-6

另外，我发现使用：while ((c = fgetc(fp)) != EOF) All++; pTextFileInfo->nSize = All;和pTextFileInfo->nSize = ftell(fp);统计文本大小结果不同，并且相差值恰好等于文本中换行符的个数。查阅资料得到原因：在windows下，以文本方式写入文件的\n会被转换为\r\n（也就是0x0D0A)，输出的时候，\r\n会被转换回\n。fgetc在读入时会将\r\n转换成一个\n；也就是说，在文本中换行符表示为\r\n，占两个字节。而用第一种方法统计大小时，\r\n被转换\n，被看作是一个字符。

另外，我学会了如何在word中插入美观的代码图。抑制住了临时学一些HTML相关知识的冲动有点惭愧。

## 函数 fprintf 的功能

将字符串按一定格式输入到指定文件。

# 项目三：利用链表对新词排序

代码见源文件。

这部分虽然用到了链表方面的知识，但是看了资料中词频统计的程序后，编写程序已经有了思路。而且语法上已经没有遇到新问题了，所遇到的问题都在项目二中遇到过，也都在项目二中得到解决方法。但是增强了用一个函数将所有子函数组建起来的能力。使用vs2017也愈发顺手。

代码如下：

1 1 

# 项目四：网址收藏夹

## 函数sprintf的功能

1. 常用方式：sprintf函数的功能与printf函数的功能基本一样，只是它把结果输出到指定的字符串中了
2. 格式化数字字符串。可以将数字作为字符数组的元素输入到数组中，并且可以控制输入格式，如左右对齐，进制转换，
3. 连接字符串。如sprintf(s, "%.\*s%.\*s", sizeof(a1), a1, sizeof(a2), a2);或者sprintf(szIECommandLine,"C:\\ProgramFiles\\InternetExplorer\\IEXPLORE.EXE %s",pszWebAddress);
4. 打印地址信息。sprintf(s,"%p",&i);

## 函数WinExec的功能

运行指定的程序。函数原型为UINT Win Exec(LPCSTR lpCmdLine, UINTuCmdShow);参数有：lpCmdLine：指向一个空结束的字符串，串中包含将要执行的应用程序的命令行（文件名加上可选参数），或者可以理解为需要打开的程序的路径和名字。uCmdShow：定义Windows应用程序的窗口如何显示。

## 为什么LabFunctions中要加#include "windows.h"

WINDOWS.H是一个最重要的头文件，它包含了其他Windows头文件，这些头文件的某些也包含了其他头文件。这些头文件定义了Windows的所有资料型态、函数调用、资料结构和常数识别字，它们是Windows文件中的一个重要部分。WinExec的函数调用中需要用到这些。

# 参考资料

百度百科、CSDN博客、CSDN论坛、博客园。