

Project 3 实验报告

21307035 邓栩瀛

1、 程序功能简要说明。

程序主要功能：

- (1) 创建文本文档，并输入一篇英语文章
- (2) 统计每个单词出现次数，并将词频从高往低进行排序
- (3) 查询指定文本文档中的指定单词,输出单词出现的位置(第几行)以及单词出现的次数

主函数（含输出界面）的实现：

```
2. int main()
3. {
4.     int j=1,l=0;
5.     int count=-1,c=-1;
6.     char filename[255];
7.     char afilename[255];
8.     char *p;
9.     while(true)
10.    {
11.        cout<<"文本分析工具"<<endl;
12.        cout<<"Enter 1: 创建文件"<<endl;
13.        cout<<"Enter 2: 单词查询及定位"<<endl;
14.        cout<<"Enter 3: 单词词频统计"<<endl;
15.        cout<<"Enter 4: Exit"<<endl;
16.        cin>>l;
17.        fflush(stdin);
18.        if(l!=1&&l!=2&&l!=3&&l!=4)
19.        {
20.            cout<<"Error,please enter 1-4"<<endl;
21.        }
22.        if(l==1)
23.        {
24.            cout<<"请输入所要创建的文件名（含拓展名）"<<endl;
25.            scanf("%[^\n]",filename);
26.            p=filename;
27.            createfile(p);
```

```
28.         cout<<"创建成功"<<endl;
29.     }
30.     if(l==2)
31.     {
32.         int next[200];
33.         while(count== -1)
34.         {
35.             cout<<"请输入所需查询的文件名称（含拓展名）"<<endl;
36.             scanf("%[^\n]",afilename);
37.             fflush(stdin);
38.             p=afilename;
39.             getnext(p,next);
40.             count=kmp(p,next);
41.             if(count==0)
42.             {
43.                 cout<<"None"<<endl;
44.             }
45.             if(count== -1)
46.             {
47.                 cout<<"Error"<<endl;
48.                 break;
49.             }
50.         }
51.     }
52.     if(l==3)
53.     {
54.         while(c== -1)
55.         {
56.             cout<<"请输入所需查询的文件名称（含拓展名）"<<endl;
57.             scanf("%[^\n]",afilename);
58.             fflush(stdin);
59.             p=afilename;
60.             c=vcount(p);
61.             if(c== -1)
62.             {
63.                 cout<<"Error,please enter a correct filename"<<en
64.                 dl;
65.                 break;
66.             }
67.         }
68.     }
69.     if(l==4)
70.     {
71.         cout<<"Exit"<<endl;
```

```
71.         break;
72.     }
73. }
74.     system("pause");
75. }
```

2、 程序运行截图，包括计算功能演示、部分实际运行结果展示、命令行或交互式界面效果等。

（1）创建文件

```
文本分析工具
Enter 1: 创建文件
Enter 2: 单词查询及定位
Enter 3: 单词词频统计
Enter 4: Exit
1
请输入所要创建的文件名(含拓展名)
1.txt
Please enter the passage(end with #):
In recent years, people have been interested in making a long journey to some unfamiliar cities which are far from their hometown.
One of their tourism purposes is mainly to experience the local culture and learn the traditions.
However, someone doubts that people can obtain culture-related information through books, films and internet instead of travel.
I disagree on this view to a certain degree, although books, films and internet are useful resources for providing valuable cultural information.#
创建成功
```

（2）统计词频 top-k 的单词

文本分析工具

Enter 1: 创建文件

Enter 2: 单词查询及定位

Enter 3: 单词词频统计

Enter 4: Exit

3

请输入所需查询的文件名称（含拓展名）

1.txt

There are 63 different words.

and 3

to 3

internet 2

films 2

books 2

information 2

the 2

of 2

their 2

are 2

a 2

people 2

cultural 1

valuable 1

providing 1

for 1

resources 1

useful 1

although 1

degree 1

certain 1

view 1

this 1

on 1

disagree 1

I 1

travel 1

instead 1

through 1

```
culture-related 1
obtain 1
can 1
that 1
doubts 1
someone 1
However 1
traditions 1
learn 1
culture 1
local 1
experience 1
mainly 1
is 1
purposes 1
tourism 1
One 1
hometown 1
from 1
far 1
which 1
cities 1
unfamiliar 1
some 1
journey 1
long 1
making 1
in 1
interested 1
been 1
have 1
years 1
recent 1
In 1
The number of words is 77
```

(3) 查询某个单词出现的次数及出现的位置（第几行）

```
文本分析工具
Enter 1: 创建文件
Enter 2: 单词查询及定位
Enter 3: 单词词频统计
Enter 4: Exit
2
请输入所需查询的文件名称（含拓展名）
1.txt
Please enter the vocabulary:
internet
Line 3
Line 4
The sum of this vocabulary is: 2
```

(4) 退出程序

```
文本分析工具
Enter 1: 创建文件
Enter 2: 单词查询及定位
Enter 3: 单词词频统计
Enter 4: Exit
4
Exit
请按任意键继续. . .
```

3、 部分关键代码及其说明。

① 获得 next[] 数组

```
(1) void getnext(char* p, int next[])
(2) {
(3)     char temp[200];
(4)     FILE *fp;
(5)     if((fp=fopen(p,"r"))==NULL)
(6)     {
(7)         cout<<"Error"<<endl;
(8)         return;
(9)     }
(10)    int i, j;
(11)    i = 0; //指向字符串每个字符的下标
(12)    j = -1;
(13)    next[i] = j; //next[0]放上-1
(14)    int length=0;
(15)    while(fscanf(fp,"%s",temp)!=EOF)
(16)    {
(17)        length+=strlen(temp);
(18)        while (i < length)
(19)        {
(20)            //没有到达结尾的话
(21)            if (j == -1 || temp[i] == temp[j])
(22)                { //如果是第一个字符或遇到相同的字符
(23)                    next[++i] = ++j;
(24)                }
(25)            else
(26)            {
(27)                j = next[j];
(28)            }
(29)        }
(30)    }
```

```
(31)
(32) }
```

② Kmp 算法

```
(33) int kmp(char* p, int next[])
(34) {
(35)     int counts=0;
(36)     int i=0,j=0,t=0;
(37)     char T[1000],S[50];
(38)     FILE *fp;
(39)     int flag_row=0;
(40)     int row=1;
(41)     if((fp=fopen(p,"r"))==NULL)
(42)     {
(43)         cout<<"Error"<<endl;
(44)     }
(45)     char ch1=fgetc(fp);
(46)     while(ch1!='#')
(47)     {
(48)         T[t]=ch1;
(49)         t++;
(50)         ch1=fgetc(fp);
(51)     }
(52)     T[t]='#';
(53)     cout<<"Please enter the vocabulary:"<<endl;
(54)     cin>>S;
(55)     while (true)
(56)     {
(57)         if (j == -1 || T[i] == S[j])
(58)         {
(59)             i++;
(60)             j++;
(61)         }
(62)         else
(63)         {
(64)             j = next[j];
(65)         }
(66)
(67)         if (!(i < strlen(T) && j < strlen(S)))
(68)         {
(69)             if (j == strlen(S))
(70)             {
(71)                 if(flag_row!=row)
(72)                 {
```

```

(73)             cout<<"Line "<<row<<" ";
(74)             counts++;
(75)             flag_row=row;
(76)             j = 0;
(77)             }
(78)         }
(79)         if (i == strlen(T))
(80)         {
(81)             return 0;
(82)         }
(83)     }
(84)     if(S[j]=='\n')
(85)     {
(86)         row=row+1;
(87)     }
(88)     if(S[j]=='#')
(89)     {
(90)         cout<<"The sum of this vocabulary is: "<<counts<<endl;
(91)     }
(92)
(93)     }
(94)     if (j == strlen(S))
(95)     {
(96)         return counts;
(97)     }
(98)     else
(99)     {
(100)         return -1;
(101)     }
(102) }

```

③ 词频统计

```

1.     int kmp(char* p, int next[])
2.     {
3.         int counts=0;
4.         int i=0,j=0,t=0;
5.         char T[1000],S[50];
6.         FILE *fp;
7.         int flag_row=0;
8.         int row=1;
9.         if((fp=fopen(p,"r"))==NULL)
10.        {

```



```

11.         cout<<"Error"<<endl;
12.     }
13.     char ch1=fgetc(fp);
14.     while(ch1!='#')
15.     {
16.         T[t]=ch1;
17.         t++;
18.         ch1=fgetc(fp);
19.     }
20.     T[t]='#';
21.     cout<<"Please enter the vocabulary:"<<endl;
22.     cin>>S;
23.     while (true)
24.     {
25.         if (j == -1 || T[i] == S[j])
26.         {
27.             i++;
28.             j++;
29.         }
30.         else
31.         {
32.             j = next[j];
33.         }
34.
35.         if (!(i < strlen(T) && j < strlen(S)))
36.         {
37.             if (j == strlen(S))
38.             {
39.                 if(flag_row!=row)
40.                 {
41.                     cout<<"Line " <<row<<" ";
42.                     counts++;
43.                     flag_row=row;
44.                     j = 0;
45.                 }
46.             }
47.             if (i == strlen(T))
48.             {
49.                 return 0;
50.             }
51.         }
52.         if(S[j]=='\n')
53.         {
54.             row=row+1;

```

```
55.     }
56.     if(S[j]=='#')
57.     {
58.         cout<<"The sum of this vocabulary is: "<<counts<<endl;
59.     }
60.
61.     }
62.     if (j == strlen(S))
63.     {
64.         return counts;
65.     }
66.     else
67.     {
68.         return -1;
69.     }
70. }
```