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：

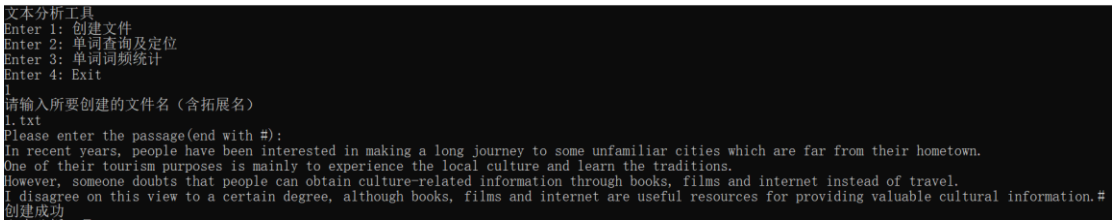
```
int main()
{
    int j=1,l=0;
    int count=-1,c=-1;
    char filename[255];
    char afilename[255];
    char *p;
    while(true)
    {
        cout<<"文本分析工具"<<endl;
        cout<<"Enter 1: 创建文件"<<endl;
        cout<<"Enter 2: 单词查询及定位"<<endl;
        cout<<"Enter 3: 单词词频统计"<<endl;
        cout<<"Enter 4: Exit"<<endl;
        cin>>l;
        fflush(stdin);
        if(l!=1&&l!=2&&l!=3&&l!=4)
        {
            cout<<"Error,please enter 1-4"<<endl;
        }
        if(l==1)
        {
            cout<<"请输入所要创建的文件名（含拓展名）"<<endl;
            scanf("%[^\n]",filename);
            p=filename;
            createfile(p);
```

```
            cout<<"创建成功"<<endl;
        }
        if(l==2)
        {
            int next[200];
            while(count==-1)
            {
                cout<<"请输入所需查询的文件名称（含拓展名）"<<endl;
                scanf("%[^\n]",afilename);
                fflush(stdin);
                p=afilename;
               getnext(p,next);
                count=kmp(p,next);
                if(count==0)
                {
                    cout<<"None"<<endl;
                }
                if(count==-1)
                {
                    cout<<"Error"<<endl;
                    break;
                }
            }
        }
        if(l==3)
        {
            while(c==-1)
            {
                cout<<"请输入所需查询的文件名称（含拓展名）"<<endl;
                scanf("%[^\n]",afilename);
                fflush(stdin);
                p=afilename;
                c=vcount(p);
                if(c==-1)
                {
                    cout<<"Error,please enter a correct filename"<<en
dl;
                    break;
                }
            }
        }
        if(l==4)
        {
            cout<<"Exit"<<endl;
```

```
            break;
        }
    }
    system("pause");
}
```

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



（2） 统计词频 top-k 的单词

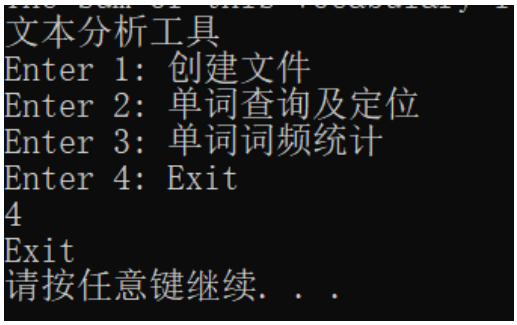
```
文本分析工具
Enter 1: 创建文件
Enter 2: 单词查询及定位
Enter 3: 单词词频统计
Enter 4: Exit
3
请输入所需查询的文件名称（含拓展名）
1.txt
There are 63 different words.
and 3
to 3
internet 2
films 2
books 2
information 2
the 2
of 2
their 2
are 2
a 2
people 2
cultural 1
valuable 1
providing 1
for 1
resources 1
useful 1
although 1
degree 1
certain 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



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)                  {
```

```
                    cout<<"Line "<<row<<" ";
                    counts++;
                    flag_row=row;
                    j = 0;
                }
            }
            if (i == strlen(T))
            {
                return 0;
            }
        }
        if(S[j]=='\n')
        {
            row=row+1;
        }
        if(S[j]=='#')
        {
            cout<<"The sum of this vocabulary is: "<<counts<<endl;
        }

    }
    if (j == strlen(S))
    {
        return counts;
    }
    else
    {
        return -1;
    }
}
```

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.    {
```

```
        cout<<"Error"<<endl;
    }
    char ch1=fgetc(fp);
    while(ch1!='#')
    {
         T[t]=ch1;
         t++;
         ch1=fgetc(fp);
    }
    T[t]='#';
    cout<<"Please enter the vocabulary:"<<endl;
    cin>>S;
    while (true)
    {
        if (j == -1 || T[i] == S[j])
        {
            i++;
            j++;
        }
        else
        {
            j = next[j];
        }

        if (!(i < strlen(T) && j <strlen(S)))
        {
            if (j == strlen(S))
            {
                if(flag_row!=row)
                {
                    cout<<"Line "<<row<<" ";
                    counts++;
                    flag_row=row;
                    j = 0;
                }
            }
            if (i == strlen(T))
            {
                return 0;
            }
        }
        if(S[j]=='\n')
        {
            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. }
```