

# 项目说明文档

## 数据结构课程设计

### ——关键字检索系统

作者姓名：\_\_\_\_\_杨鑫\_\_\_\_\_

学        号：\_\_\_\_\_1950787\_\_\_\_\_

指导教师：\_\_\_\_\_张颖\_\_\_\_\_

学院、专业：\_\_\_\_\_软件学院 软件工程\_\_\_\_\_

同济大学

Tongji University

## 目录

1	分析.....	- 1 -
1.1	背景分析.....	- 1 -
1.2	功能分析.....	- 1 -
2	设计.....	- 1 -
2.1	数据结构设计.....	- 1 -
2.2	类结构设计.....	- 1 -
2.3	成员与操作设计.....	- 2 -
	文件系统类: .....	- 2 -
2.4	系统设计.....	- 2 -
3	实现.....	- 3 -
3.1	文件输入功能的实现.....	- 3 -
3.1.1	文件输入功能的流程图.....	- 3 -
3.1.2	文件输入功能的核心代码.....	- 3 -
3.1.3	文件输入功能的截屏示例.....	- 3 -
3.2	关键字查找功能的实现.....	- 4 -
3.2.1	关键字查找功能的流程图.....	- 4 -
3.2.2	关键字查找功能的核心代码.....	- 4 -
3.2.3	关键字查找功能的截屏示例.....	- 5 -
4	测试.....	- 6 -
4.1	功能测试.....	- 6 -
4.1.1	全匹配测试.....	- 6 -
4.1.2	部分匹配测试.....	- 6 -
4.2	出错测试.....	- 7 -
4.2.1	后缀名错误测试.....	- 7 -

# 1 分析

## 1.1 背景分析

在计算机时代，文件是被广泛使用的，不可或缺的一个存在。在信息类越来越庞大的今天，如何处理文件中的信息也显得十分关键和重要。处理文件信息的一个方面就是关键字的检索。如何在大量的数据中快速找到某些关键字，是我们需要考虑的一个非常重要的问题。

## 1.2 功能分析

为了解决上述问题，本程序实现了一个文本文件的建立与检索系统，可以根据用户的输入建立文本文件，并根据该文件的内容检索某个关键字。

为了更加贴合实际，本程序还提供了用户两种选择：全匹配和部分匹配。这使得本程序的应用面更加广阔和更加便捷。在关键字匹配时采用了 KMP 算法，和传统算法相比大大减少了时间复杂度：传统的时间复杂度为  $O(m*n)$ ，而采用了 KMP 算法以后可以减少到  $O(m+n)$ ，大大提高了性能。

# 2 设计

## 2.1 数据结构设计

本程序最重要的数据结构就是文件。因为文件中的信息量通常非常多，所以在文件信息输入输出的时候，都在一个大小适中的缓冲区中进行多次处理，这样可以防止数据量过大带来一些意外的影响。

## 2.2 类结构设计

本程序设计了一个文件系统类 (FileSystem)。它保存着文件的名称，并整合了一系列操作：文件信息的输入，输出，关键字的查找等。这有利于程序的编写和用户的使用。

## 2.3 成员与操作设计

文件系统类：

私有成员：

1. `char* file;` // 文件名， 代表着一个文本文件
2. `int* next;` // `next` 数组， 是根据待查找的关键字产生的， 用于 KMP 算法进行关键字的查找
3. `void getNext(string pat, int* next);` // 产生 `next` 数组
4. `int Search(int start, string target, string pat, bool tag);` // 查找关键字

公有操作：

1. `FileSystem() : file(NULL), next(NULL) {}` // 构造函数
2. `~FileSystem();` // 析构函数
3. `bool CreateFile(char* fileName);` // 创建一个文本文件
4. `bool InputFile();` // 向文件输入文本
5. `int OutputFile(char* fileName, string pat, bool tag);` // 输出文件内容并用 KMP 算法统计与关键字匹配的数量

## 2.4 系统设计

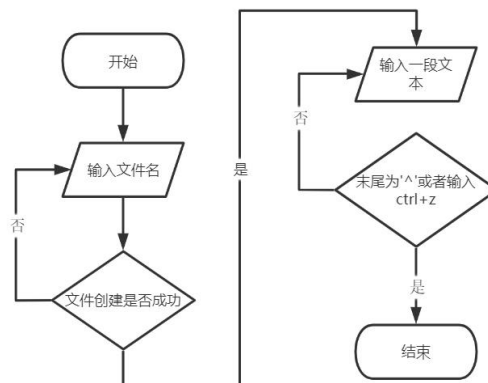
程序开始时先让用户输入文件名，并提示文件名后缀必须以.txt 结尾，若不符合将被提示重新输入；然后让用户输入一段文本作为文件输入，并以 ctrl+z 或者 ^ 作为结束标志。成功建立文件后，让用户输入待查找的关键字，已经查找方式（全匹配或者部分匹配）然后根据内部函数执行的结果返回查找结果。

程序兼容了 windows 和 LINUX 平台，在双平台下均可以正常运行。

## 3 实现

### 3.1 文件输入功能的实现

#### 3.1.1 文件输入功能的流程图



#### 3.1.2 文件输入功能的核心代码

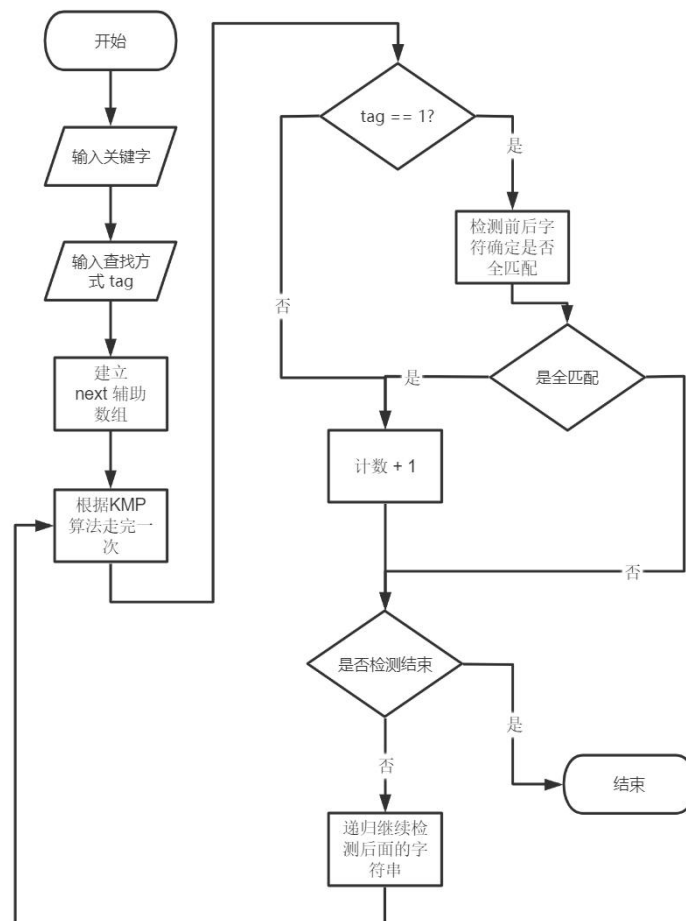
```
1.  char buffer[500]; // 缓存数组， 用于临时接受键盘输入
2.  cout << endl << "请输入一段英文：（在文本最后输入 ^ 或换行后输入 Ctrl + Z 以作为为结束标志）" << endl;
3.
4.  // 不断从键盘接受输入到缓存数组中， 直到遇到结束标志
5.  while (cin.getline(buffer, sizeof(buffer))) {
6.      out << buffer << endl; // 将缓存数组中的内容输入到文件
7.
8.      // 检测末尾是否有结束标志
9.      if (buffer[strlen(buffer) - 1] == '^') break;
10. }
11.
12. out.close(); // 关闭文件
```

#### 3.1.3 文件输入功能的截屏示例

```
请输入文件名: tem.txt
请输入一段英文：（在文本最后输入 ^ 或换行后输入 Ctrl + Z 以作为为结束标志）
Real beauty comes from learning, growing, and loving in the ways of life. That is
the Art of Life. You can learn slowly, and sometimes painfully, by just waiting fo
r life to happen to you. Or you can choose to accelerate your growth and intention
ally devour life and all it offers. You are the artist that paints your future wit
h the brush of today.
本段文本已保存到tem.txt中！
请输入要检索的文字（不含空格）：
```

## 3.2 关键字查找功能的实现

### 3.2.1 关键字查找功能的流程图



### 3.2.2 关键字查找功能的核心代码

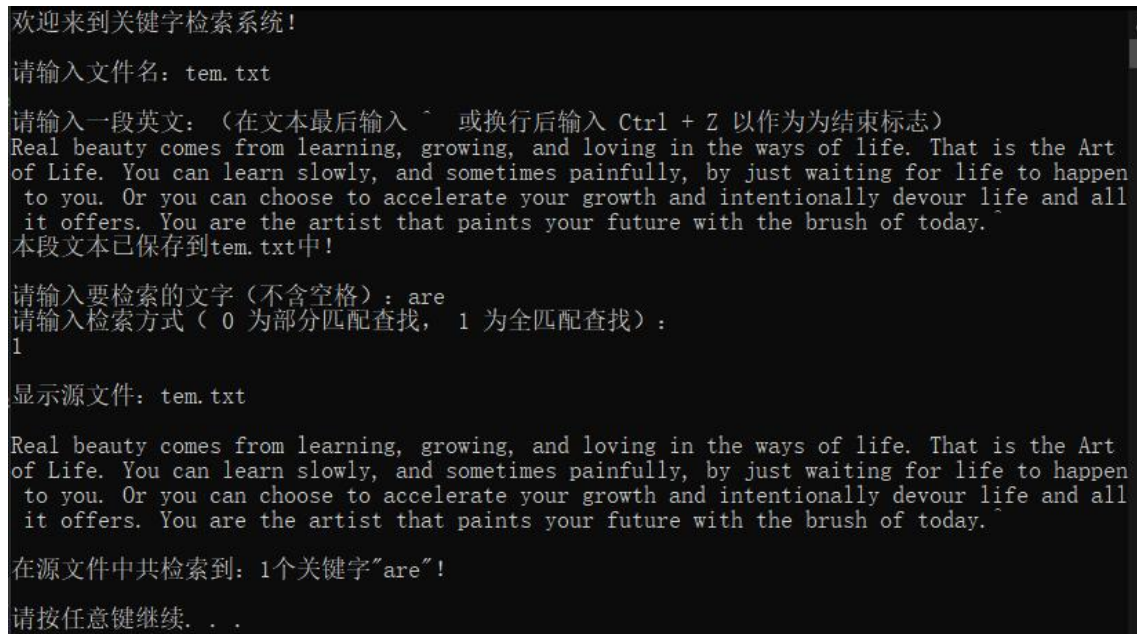
```
1.  int posP = 0, posT = start; // 初始化 pat 和 target 的起始位置
2.  int lP = pat.length(), lT = target.length(); // 获取长度
3.  if (start >= lT) return 0; // 起始位置超过最后的位置， 直接返回 0
4.
5.  // 根据 KMP 算法， 得到 next 数组后， 对两个字符串进行搜索， 时间复杂度
   // 为 O(m + n)
6.  while (posP < lP && posT < lT) {
7.      if (posP == -1 || pat[posP] == target[posT]) posP++, posT++;
8.      else posP = next[posP];
9.  }
```

```

10.
11. // 检测到末尾， 无匹配的字段， 则关键字为 0 个
12. if (posP < lP) return 0;
13. else if (!tag) { // 若匹配模式为部分匹配， 无需检测前后是什么字符
14.     return 1 + Search(posT, target, pat, tag); // 找到 1 个关键字
        后， 继续从后面开始进行下一轮查找
15. }
16. else { // 若匹配模式为全匹配， 则需检测前面和后面的字符
17.     if ((posT - lP == 0 || !isalnum(target[posT - lP - 1])) && (posT >= lT || !isalnum(target[posT]))) { // 若前面还有字符，则不能为字母或者数字； 若后面还有字符，也不能为字母或者数字。当满足该条件时说明成功
        找到一个全匹配的目标， 计数 + 1
18.         return 1 + Search(posT, target, pat, tag); // 找到 1 个关键字后， 继续从后面开始进行下一轮查找
19.     }
20.     else { // 不满足时， 计数不增加， 继续从后面开始检测
21.         return Search(posT, target, pat, tag);
22.     }
23. }

```

### 3.2.3 关键字查找功能的截屏示例



```

欢迎来到关键字检索系统！
请输入文件名： tem.txt

请输入一段英文：（在文本最后输入 ^ 或换行后输入 Ctrl + Z 以作为为结束标志）
Real beauty comes from learning, growing, and loving in the ways of life. That is the Art
of Life. You can learn slowly, and sometimes painfully, by just waiting for life to happen
to you. Or you can choose to accelerate your growth and intentionally devour life and all
it offers. You are the artist that paints your future with the brush of today.^
本段文本已保存到tem.txt中！

请输入要检索的文字（不含空格）： are
请输入检索方式（ 0 为部分匹配查找， 1 为全匹配查找）：
1

显示源文件： tem.txt

Real beauty comes from learning, growing, and loving in the ways of life. That is the Art
of Life. You can learn slowly, and sometimes painfully, by just waiting for life to happen
to you. Or you can choose to accelerate your growth and intentionally devour life and all
it offers. You are the artist that paints your future with the brush of today.^

在源文件中共检索到： 1个关键字“are”！
请按任意键继续. . .

```

## 4 测试

### 4.1 功能测试

#### 4.1.1 全匹配测试

测试用例：输入一段文本并全匹配检测

预期结果：程序正常运行和输出

实验结果：

```
欢迎来到关键字检索系统！
请输入文件名： tem.txt
请输入一段英文：（在文本最后输入 ^ 或换行后输入 Ctrl + Z 以作为结束标志）
Real beauty comes from learning, growing, and loving in the ways of life. That is the Art
of Life. You can learn slowly, and sometimes painfully, by just waiting for life to happen
to you. Or you can choose to accelerate your growth and intentionally devour life and all
it offers. You are the artist that paints your future with the brush of today.
本段文本已保存到tem.txt中！

请输入要检索的文字（不含空格）： are
请输入检索方式（ 0 为部分匹配查找， 1 为全匹配查找）：
1

显示源文件： tem.txt

Real beauty comes from learning, growing, and loving in the ways of life. That is the Art
of Life. You can learn slowly, and sometimes painfully, by just waiting for life to happen
to you. Or you can choose to accelerate your growth and intentionally devour life and all
it offers. You are the artist that paints your future with the brush of today.

在源文件中共检索到：1个关键字“are”！

请按任意键继续. . .
```

#### 4.1.2 部分匹配测试

测试用例：输入一段文本并全匹配检测

预期结果：程序正常运行和输出

实验结果：

```
欢迎来到关键字检索系统！
请输入文件名： tem.txt
请输入一段英文：（在文本最后输入 ^ 或换行后输入 Ctrl + Z 以作为结束标志）
Real beauty comes from learning, growing, and loving in the ways of life. That is the
Art of Life. You can learn slowly, and sometimes painfully, by just waiting for life t
o happen to you. Or you can choose to accelerate your growth and intentionally devour
life and all it offers. You are the artist that paints your future with the brush of t
oday.
本段文本已保存到tem.txt中！

请输入要检索的文字（不含空格）： be
请输入检索方式（ 0 为部分匹配查找， 1 为全匹配查找）：
0

显示源文件： tem.txt

Real beauty comes from learning, growing, and loving in the ways of life. That is the
Art of Life. You can learn slowly, and sometimes painfully, by just waiting for life t
o happen to you. Or you can choose to accelerate your growth and intentionally devour
life and all it offers. You are the artist that paints your future with the brush of t
oday.

在源文件中共检索到：1个关键字“be”！

请按任意键继续. . .
```



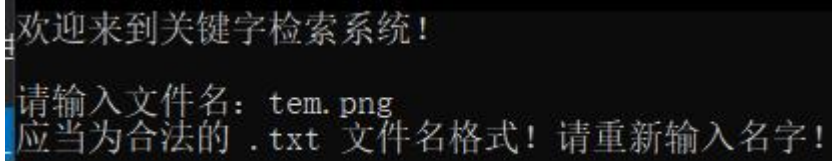
## 4.2 出错测试

### 4.2.1 后缀名错误测试

测试用例：输入一个带有错误后缀的文件名

预期结果：程序会提示用户重新输入

实验结果：



```
欢迎来到关键字检索系统！  
请输入文件名：tem.png  
应当为合法的 .txt 文件名格式！请重新输入名字！
```