

高级语言程序设计

课程设计报告

学 院 班 级

姓 名 学 号

成 绩 指导老师

年 月 日

**课程设计答辩记录表**

|  |  |
| --- | --- |
| **课程设计题目及要求（请填入内容）** | |
| **题目11：**简易文本编辑器  **要求：**不允许使用string对象。要使用字符数组，自行编写字符的基本操作处理方法。  文本数据要求行内采用字符数组，行间采用链式结构。 | |
| **完成情况：（答辩时由老师填写如下内容）** | |
| 基本数据结构设计 |  |
| 算法设计及流程图 |  |
| 输入输出设计 |  |
| 代码规范及注释 |  |
| 数据测试及调试 |  |
| 代码理解与说明 |  |
| 评语 |  |
| 综合评分 |  |

目 录

系统功能结构模块图 ………………………………………………… 1

数据结构设计及用法说明 …………………………………………… 2

程序结构及流程图 …………………………………………………… 3

各模块功能介绍 ……………………………………………………… 4

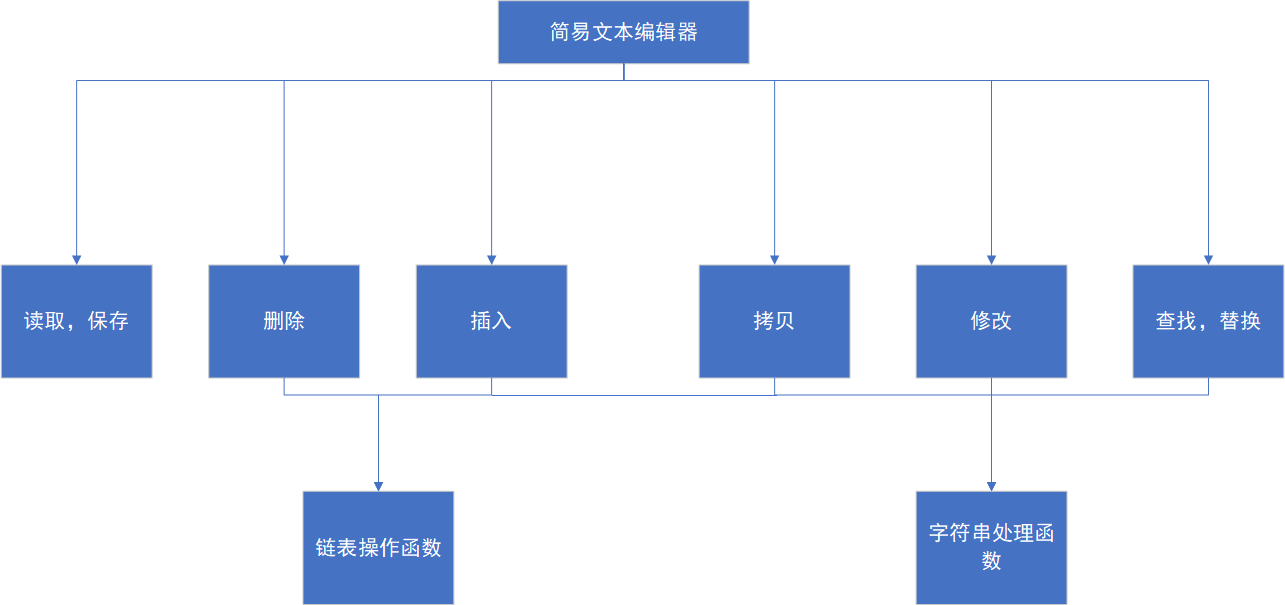
实验结果 ……………………………………………………………… 5

体会 …………………………………………………………………… 8

参考文献 ……………………………………………………………… 9

附录 1

系统功能结构模块图



本系统由六个主要功能组成，文本行间通过链表的形式链接，行内采用字符串处理函数，编辑完成的数据储存在文件中。

1

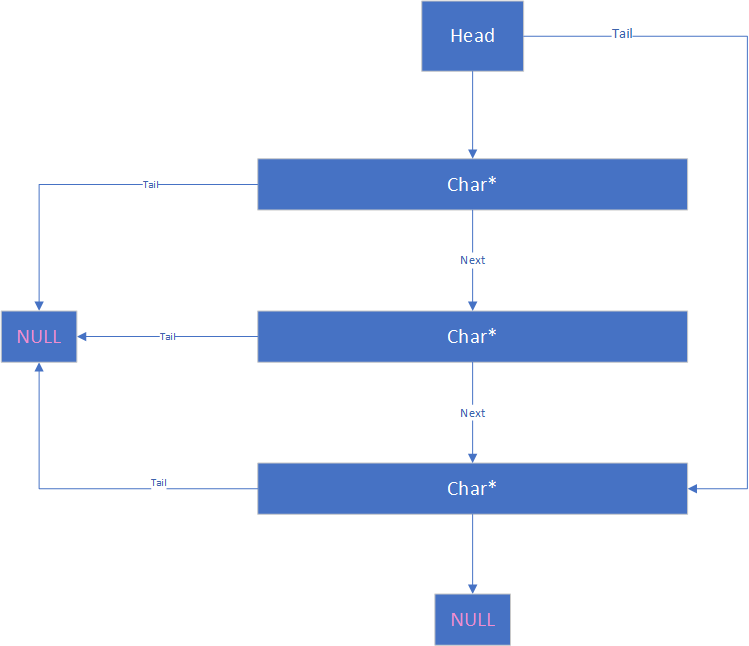
数据结构设计及用法说明

该文本编辑器采用单链表与数组的组合方式，行间用链表相连，行内用字符串数组储存字符。

1. 链表

其形式为带头结点及尾指针的单链表，尾指针只在头节点中定义，行间用next指针相连，使非连续的内存空间相连接。

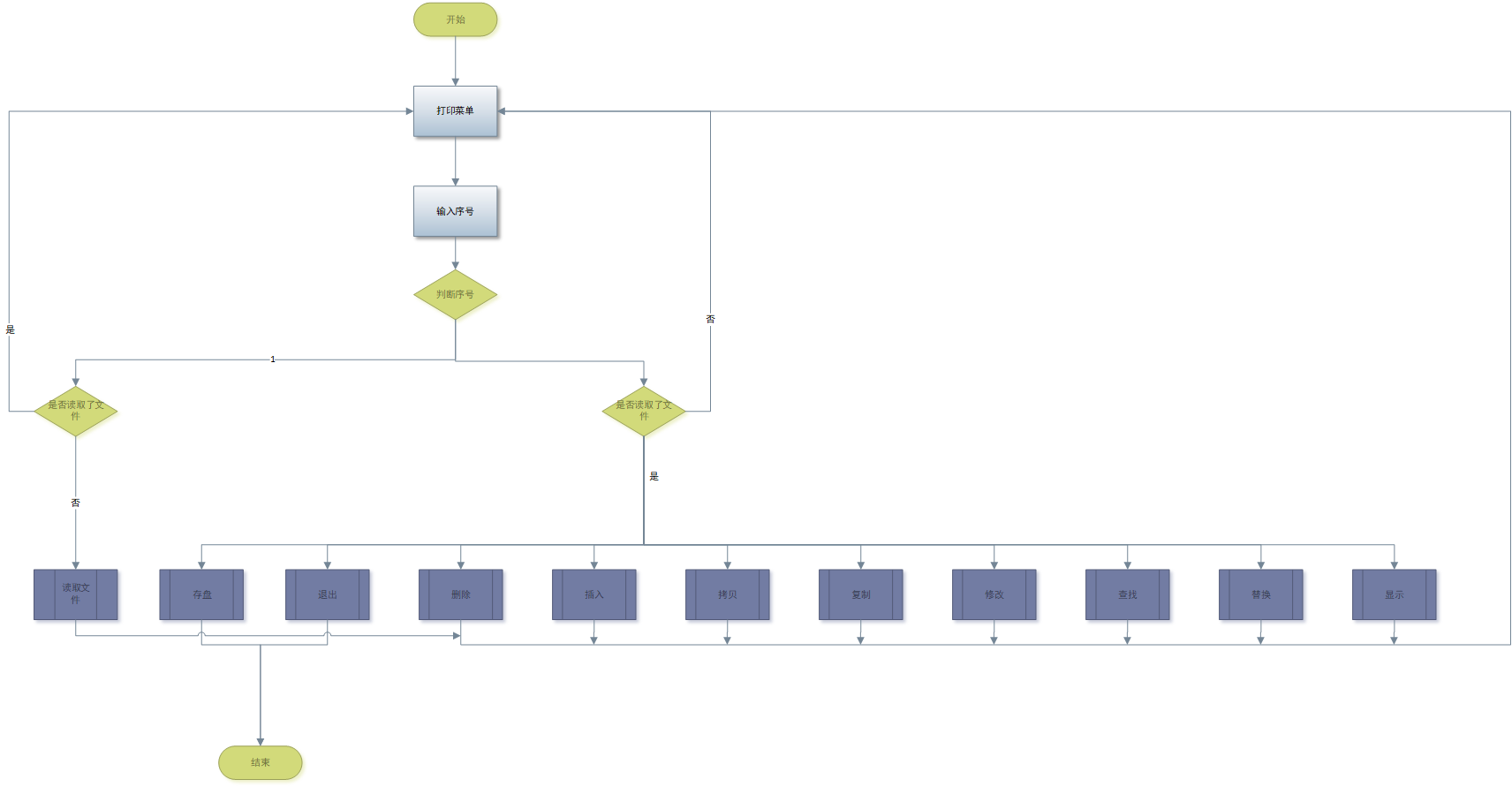
1. 字符串数组

其形式为内存空间固定的数组。

1. 结构体

包含了字符串数组以及next指针，为链表的单元。

2

程序结构及流程图

程序结构：顺序，循环，选择；

3

各模块功能介绍

系统功能介绍：

读取：该模块的功能是读取指定路径下的指定文件，如果没有该文件，则会新建一个新文件，如果已经读取过文件，则该功能不可用。

存盘：该模块功能是把链表中的内容写入指定文件中，如果未先读入文件，则该功能不可用。

退出：放弃编辑，对已执行的操作不做任何保存，直接退出系统。

删除：该功能为批量删除指定行，如果未先打开文件，文件里没有任何文本，及输入的行号有误，则该功能无法执行。

插入：该功能为批量插入行，如果未先打开文件，则无法执行。如果输入行号不存在则默认为插入第一行。如果单行太长，系统自动将超出部分换行。

查找：该功能为线性遍历链表，可按行号查找以及关键字查找两种功能，找到则返回目标前一个位置的指针，找不到则返回空指针。如果未载入文件或者文件为空则无法使用。

拷贝：该功能为批量复制黏贴，把指定行复制并插入到指定位置。如果为载入文件，文件为空或者输入行号有误则无法使用。

修改：该功能为指定修改某一行的内容。如果未载入文件，文件为空，输入行号有误则该功能无法使用。若修改后的字符串过长则自动换行。

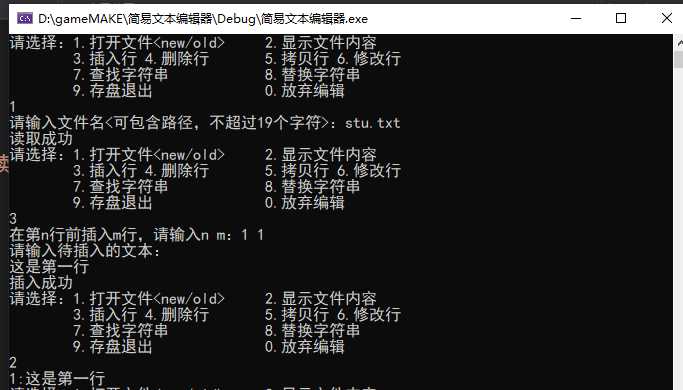
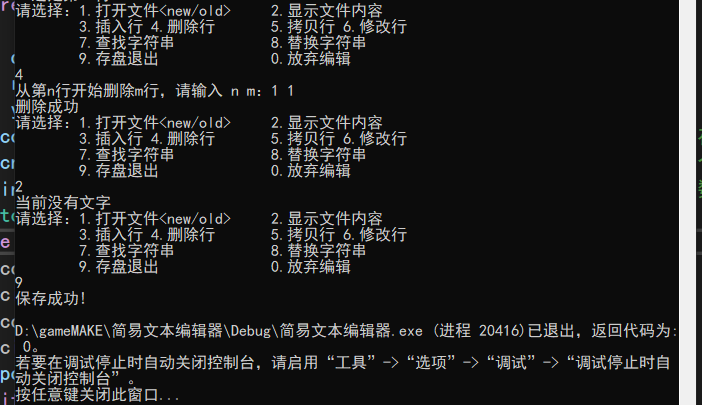
替换：能够将指定字符串替换成新字符串，如果文件未载入则无法使用。如果新字符串过长也会自动换行。

自定义函数介绍：

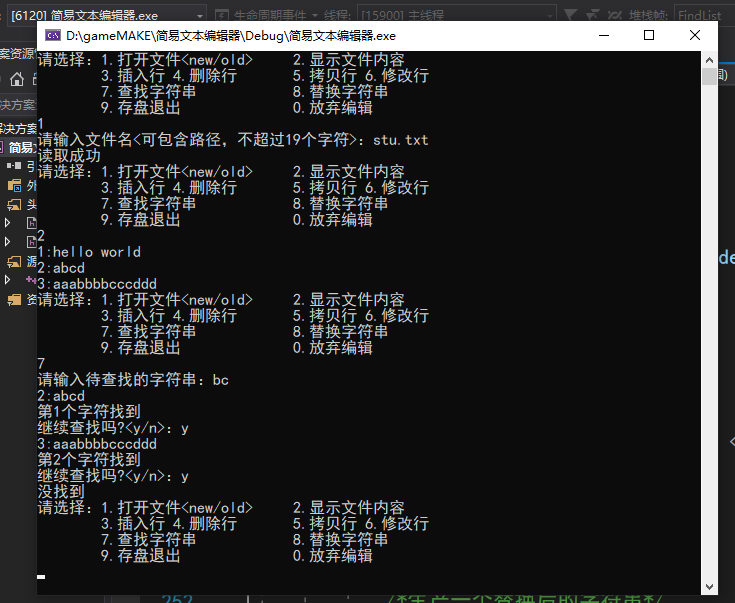
1. CreateList()：初始化链表的头节点。
2. InsertList()：插入链表的任意一个位置
3. AppendList()：在链表尾部添加一个单元
4. FindList()：查找链表，两个重载，一个按序号查找，一个按链表内容查找。
5. DeleteList()：批量删除指定的链表结点。
6. PrintList()：打印链表的所有内容。
7. ModifyList()：指定一个结点修改其内容。
8. Str\_cpy()：字符串复制。
9. Str\_len()：获取字符串长度。
10. Str\_strKMP()：用KMP算法查找子串。
11. Str\_cmp()：比较两个字符串。

4

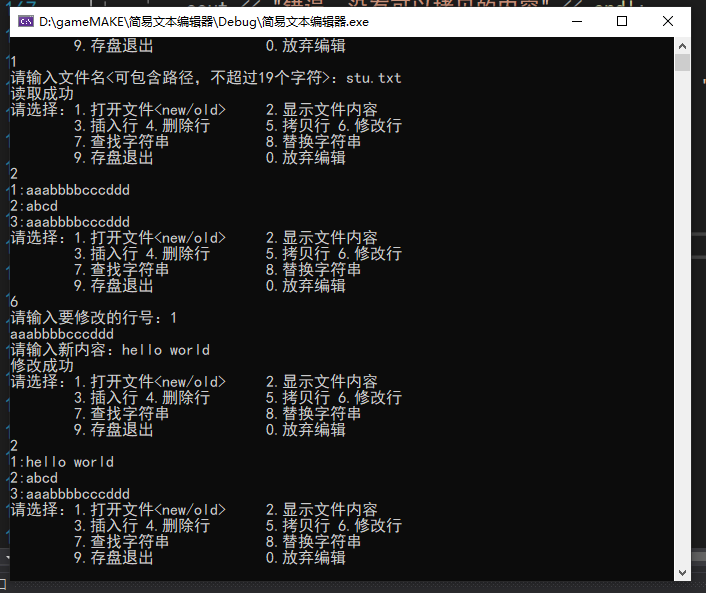
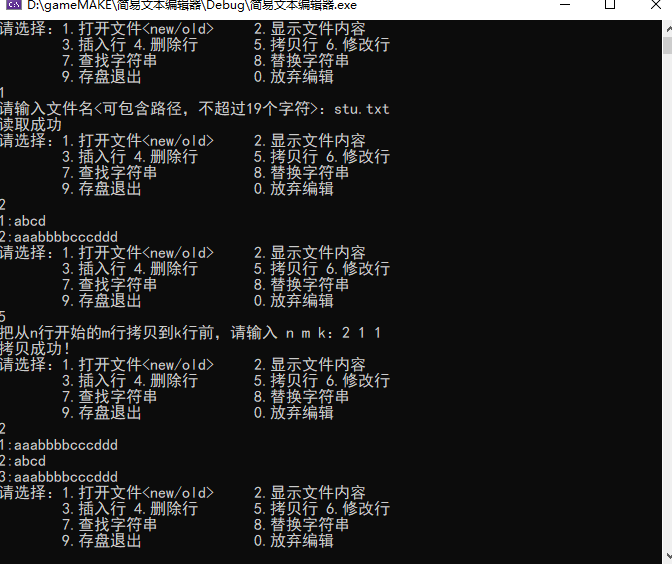
实验结果

1. 基本功能演示

5



6

2.容错演示（现场演示）

7

心得体会

开发这个系统，让我收获良多。在实验过程中，我遇到了不少困难，在修复bug，跨越这些困难的同时，也让我领会到了开发的乐趣和不易，让我对编程有了更加深刻的理解。

经过总结与反思，我认为这个系统的最大难点在于第八个功能——替换字符串，在实践过程中，不仅需要设计实现的算法，也要考虑到许多细节问题，如替换的字符串长度小于原来的字符串，则应把原来多出来的去掉。算法的设计我采用的是另外开辟一个字符串数组作为替换完成的字符串，然后直接覆盖原结点的内容。在实现这个功能的过程中，我想到，如果文本量非常大的时候，搜索就会变得很慢，因为遍历链表的时间复杂度是固定的，于是我就想到要优化搜索子串的算法，经过在网上查找资料和学习，我掌握了KMP算法，虽然不是最快的算法，但比起暴力搜索已经优化了不少了。

KMP算法的思路大致是利用标记数组记录最长相同前后缀的长度，然后搜索时不仅要移动查找串，也要移动模式串，通过减少重复搜索来优化效率。

其他功能的实现也同样有值得思考的地方，如批量删除，和批量插入也需要认真思考实现方法，批量插入我采用的是逐个插入，节约了内存空间，而拷贝我则是利用建立新链表再连接的方法。

我将自定义函数封装进了两个头文件，提高了代码的可读性。在开发过程中我发现，既然是文本编辑器，那一定需要如word一样具有自动换行的功能，于是我在修改及插入功能内添加了自动换行的机制，丰富了程序的交互性。

经过了这一次锻炼，我发现开发一个项目不仅能巩固已经学过的知识，也能收获新的体会和新的知识，也发现了自身的缺点和不足，帮助了我改正错误，为继续深入学习打下基础，做足准备。

8

参考文献

《C语言程序设计（第三版）》 作者：颜晖

<https://www.cnblogs.com/zhangtianq/p/5839909.html> 作者：July

附录1

源码GitHub：<https://github.com/838239178/TextEdit>

下载 网盘：<https://pan.baidu.com/s/1aq2aBPUgOg5-CsKjKDU_SA>