

**模拟BC的汉化文本编辑器**

C语言课程设计

**终**

**期**

**报**

**告**

**专业班级：**

**小组成员：**

**指导老师：**

**上交时间：二〇二三年四月二十三日**

目录

[第一部分 前言 1](#_Toc55022825)

[一、 编写背景 1](#_Toc55022826)

[二、 软件功能 1](#_Toc55022827)

[三、 参考资料 1](#_Toc55022828)

[四、 参考软件 1](#_Toc55022829)

[第二部分 任务概述 2](#_Toc55022830)

[一、 目标功能 2](#_Toc55022831)

[二、 编写规范 2](#_Toc55022832)

[第三部分 运行环境和配置 4](#_Toc55022833)

[一、 硬件接口 4](#_Toc55022834)

[二、 软件接口 4](#_Toc55022835)

[三、 控制 4](#_Toc55022836)

[第四部分 需求分析与系统设计 5](#_Toc55022837)

[一、 需求分析 5](#_Toc55022838)

[二、 系统设计 6](#_Toc55022839)

[第五部分 算法设计 17](#_Toc55022840)

[一、 数据结构设计 17](#_Toc55022841)

[二、 算法说明 19](#_Toc55022842)

[第六部分 界面设计 23](#_Toc55022843)

[第七部分 函数说明及函数原型 27](#_Toc55022846)

[第八部分 源代码 33](#_Toc55022847)

[第九部分 个人总结 118](#_Toc55022848)

[一、 组员胡耀文总结 118](#_Toc55022849)

[二、 组员王思为总结 119](#_Toc55022850)

[第十部分 时间安排与代码分工 121](#_Toc55022851)

[一、 时间安排 121](#_Toc55022852)

[二、 代码分工 121](#_Toc55022853)

**第一部分 前言**

1. **编写背景**

文本编辑器（或称文字编辑器）是用作编写普通文字的应用软件，也是最常用的文档创建和编辑工具。随着计算机科学与技术的发展，用来处理文本的编辑器随处可见，并且形式多样。比如，Windows下的记事本，写字板等都是十分优秀的文本编辑器和处理工具，在编辑器种类越来越多，功能越来越丰富的今天，BorlandC++（BC）由于无汉化版本而存在一定使用上的困难，但其也同样具有文本编辑的功能，如果可以开发一个模拟BC的汉化文本编辑器即可在一定程度上改善这种问题。

文本编辑器一般拥有以下几种基本功能，即查找和替换（在一个文本文件中查找一个词或者一个字符串，把一个字符串替换成另一个字符串。可采取各种方式进行：全局查找替换、条件查找替换、无条件查找替换），剪切、复制、粘贴（文件内的或者文件间的复制或者移动），文字排版（如卷行、自动缩排、行号排版、注释排版等等），导入（读取或者合并另一个文本文件的内容至当前正在编辑的文件），过滤（把正在编辑的文件发送至另一个工具，并把过滤后的结果回读至这个文件）

我们制作了一个模拟BC的汉化编辑器，它不仅具有如输入、删除、搜索字符，替换，复制、剪贴、粘贴文字和打开、关闭、保存文本等基本功能，同时还可以模拟BC的部分功能，如括号匹配。此外，我们还添加了汉化功能，如VGA下的汉化菜单，以及gbk编码下汉语的输入删除。

本软件具有BC的基础功能，同时弥补了BC不能正常显示与编辑文本中的汉字，以及BC的菜单项没有汉化的问题，适合中国的编程开发者使用。

1. **软件功能**

利用C语言的链表数据结构及相关函数，本系统实现了一个与BC31相似的文本编辑器。在此文本编辑器中，用户可以通过快捷键或鼠标选择菜单项，完成基本的文本编辑器和文件处理工作，除此之外，我们改进了BC不能输入中文的问题，使得用户可以进行中文的键入和注释，并且可以进行基本的语法检查，还可以通过注册或登录账户保存个性化设置。

1. **参考资料**1. 王士元. C高级实用程序设计. 北京: 清华大学出版社. 1996年  
   2. 周纯杰，何顶新等. 程序设计与应用（用C/C++编程）. 北京: 机械工业出版社. 2008年  
   3. [美] Prata. C Primer Plus（第六版）北京：人民邮电出版社. 2016年
2. **参考软件**

1. Borland C++ 3.1

2. Dev C++ 5.11

# 第二部分 任务概述

## 目标功能

1. 字符的编辑，如输入、删除、剪切、复制、粘贴、括号匹配等
2. 文件管理操作，如新建文件、打开文件、保存文件、文件另存为
3. 提示功能，如关于界面和帮助界面
4. 汉化功能，汉化显示菜单，打开文件中的汉字，而且可以输入删除汉字
5. 查找替换功能，查找文件中的特定字符串，并且可以替换成另外的字符串
6. 侧边滑动条功能，通过鼠标点击进行快速跳转
7. 右键快捷选择功能，对部分编辑操作可以通过右键快捷选择

## 编写规范

1. 相对独立的程序块之间、变量说明之后加空行，较长的语句分成多行书写
2. 大括号独占一行并对齐，缩进采用四个空格
3. 边写代码边注释，注释的量应达到让程序清晰易懂
4. 使用/\* \*/注释时，添加在代码上方，注释内容前后空一格

/\* explain the usage of the function \*/

static void function(char\*\* s, int\* count, int\* len, int\* maxlen)

1. 使用//注释时，添加在一行代码之后，跟代码之间空一格

int a[4]; //explain the statement

1. 标识符的命名要清晰、明了，有明确含义
2. 变量和参数用小写字母开头的单词组合而成，如int graphdriver = VGA;
3. 不使用贴图，不使用goto 语句。

# 第三部分 运行环境和配置

## 硬件接口

处理器：Intel Pentium 166 MX 或以上。

硬盘：空间500MB 以上。

屏幕适配器：VGA 接口。

系统运行内存：要求32MB 以上。

## 软件接口

开发软件工具：Borland C++ 3.1

文字编辑工具：dev-c++5.11

操作系统：DOS / WINDOWS XP

## 控制

该系统通过鼠标与键盘直接进行控制。

用户将鼠标移至需要选择的菜单项进行点击，可点击鼠标左键将光标移至对应的文字处，选中文本以后也可以点击鼠标右键进行快捷编辑操作。

用户可以通过键盘完成文本的输入功能，同时通过一系列控制组合键来对文本进行操作。

# 第四部分 需求分析与系统设计

## 需求分析

使用本软件的目标用户，是对汉化的BC有需求的用户，或者是对能打开文本中内含汉字的文本编辑器有需求的用户。

对使用“模拟BC的汉化文本编辑器”的用户需求进行分析后，我们认为本软件首先应该具有文本编辑器的基本功能，如文本编辑和文件操作的基本功能。此外，还需要实现BC的编辑器无法识别、显示、编辑汉字的功能。

模拟BC方面，本软件实现了编写代码时括号自动匹配的功能。

汉化方面，本软件支持gbk编码下的文字显示与编辑，可以用中文查看菜单并选择菜单项，同时还提供了帮助界面，并且还能显示打开的文本中的汉字。

本软件旨在为用户提供一个多功能的文本编辑器，让用户既可以单纯地进行文字编辑工作，又可以在需要时方便快捷地编写代码。

## 系统设计

* + - 1. **程序执行主流程**

程序执行主流程为mainloop函数，它是在main()函数中调用实现的。它首先初始化一些变量及结构数组，接着调用drawmain()函数来显示主窗口，然后调用while(1)进入主循环，进行鼠标更新并等待用户操作，最后程序根据用户的按键值或点击位置，进行相应的处理，完成文本编辑的相关工作。下面对按键判断和相关处理作补充说明。

（1）若按键为常规字符，即其ASCII码大于32小于127，则调用insert()函数，并在该函数中判断插入位置对应的情况并执行插入操作。

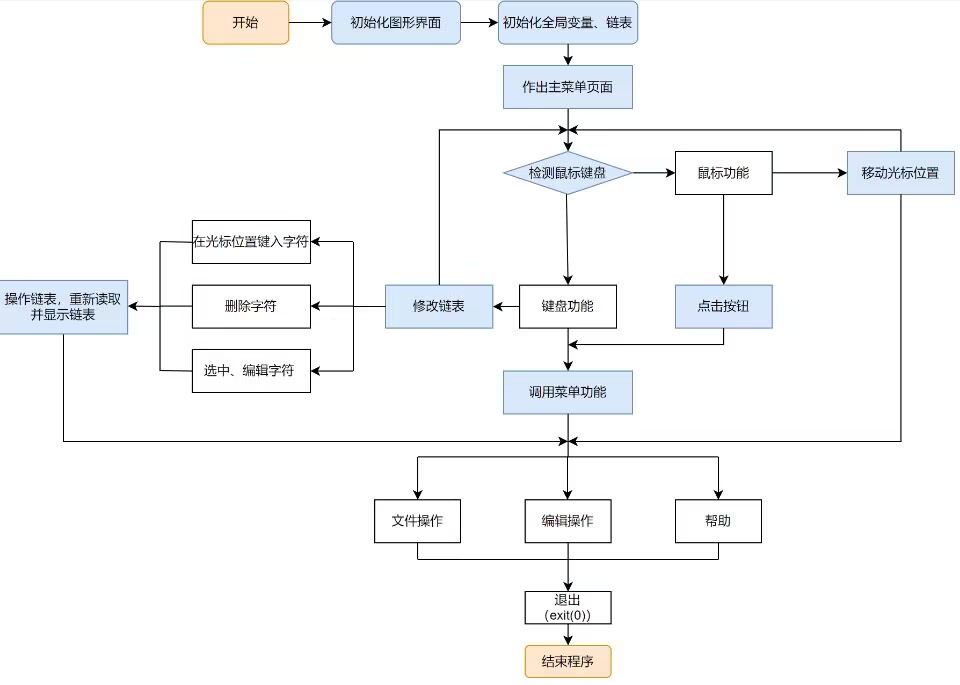
（2）若按键为Enter键，则调用enter函数进行相关的换行操作。

（3）若按键为光标移动键（左，右，上，下）且移动后的位置满足相关条件，则执行gotoxy()操作，将光标移动至目标位置。

（4）若按键为BackSpace键，则判断是否有选中的文字，如果有则调用循环执行连续的删除操作，否则调用deltext函数将光标的前一个字符从单链表中删除；若按键为Del键，也将调用deltext函数将光标的当前位置的字符从单链表中删除。

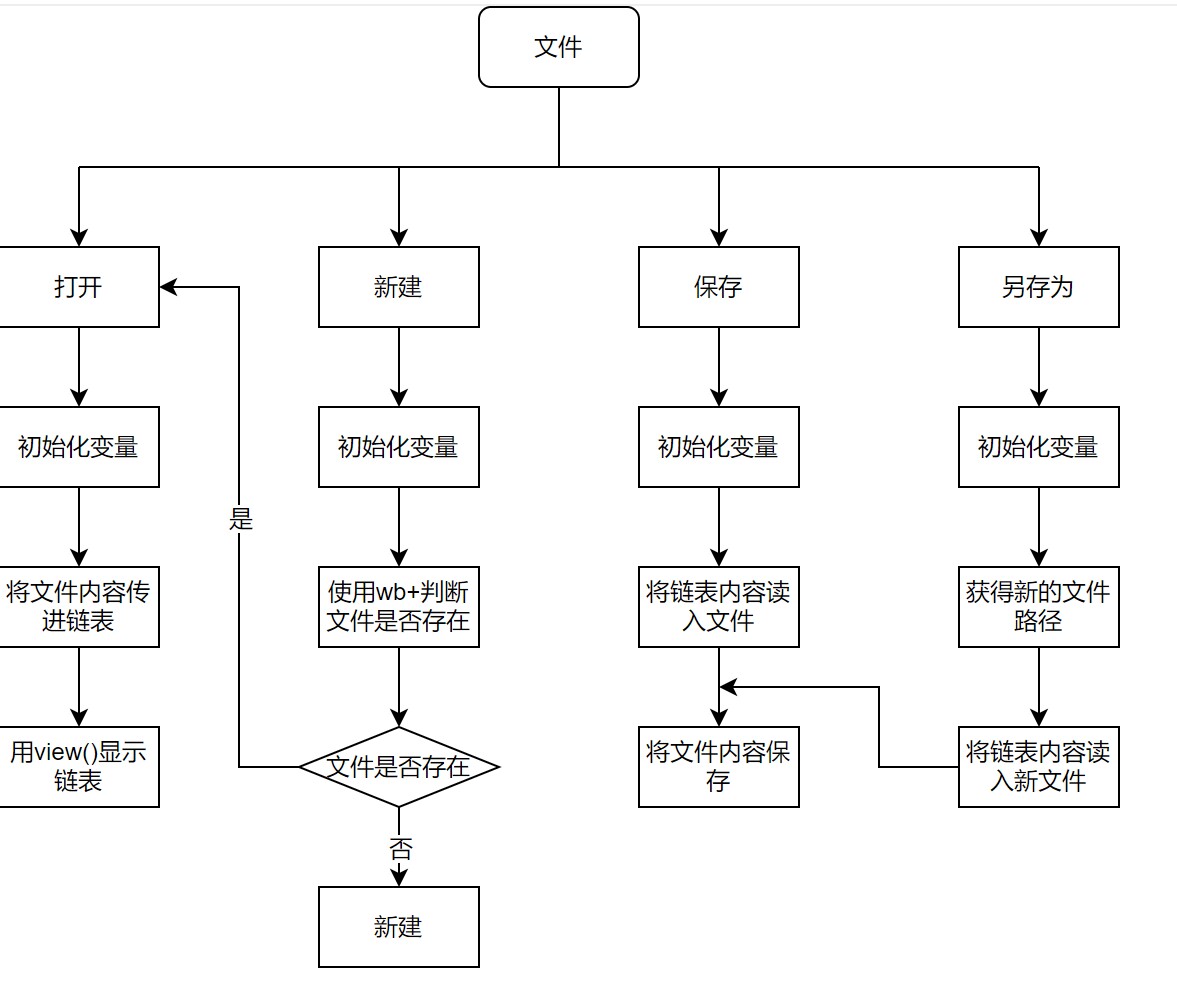
（5）若按键为Ctrl开头的按键，则执行与其相关的操作。具体来说，若为Ctrl+右移键（→），则将选定当前光标的位置开始向右的一个字符，若按住Ctrl键不放，连续按右移键，可以选定多个字符。若为Ctrl+左移键（←），则将执行与以上相同的操作。若为Ctrl+X键，则将选定相关内容保存起来，且从单链表中删除选定的字符后重新显示单链表的内容。若为Ctrl+C键，则将选定的相关内容保存起来，重新显示单链表中的内容（目的：为了去除字符的底色）。若为Ctrl+V键，则循环调用insert()函数将保存起来的字符执行插入操作。

（6）若按键为F1,F2,F3功能键，则调用对应的菜单选择函数，在此函数中完成案件的具体判断和执行相应功能操作。若为F1键，则调用File菜单；若为F2键，则调用Edit菜单；若为F3键，则调用Help菜单。



**2.文件操作模块**

在此模块中，主要实现文件的新建、打开、保存和另存为操作。在此系统中，文件的新建操作实现比较简单，文件另存为操作与保存操作类似，下面重点介绍在此文本编辑器程序中，文件的打开和保存操作的具体设计和实现。在介绍之前，我们先简单描述一下程序中用到的保存数据的数据结构。在此程序中，共有两种类型的单链表，我们称其为行单链表和列单链表，一个列单链表用来保存一行的字符，有多少行即有多少个这样的单链表。行单链表只有一个，它的每个节点的数据域用来保存不同列单链表的首节点的地址。例如，第4行第4列的字符保存在行单链表的第4个节点的数据域所指的列单链表的第4个节点的数据域中。



（1）打开文件

文件的打开过程中，它首先提示用户输入要打开文件的文件名，若该文件不存在或由于其他原因打开失败，则会结束文件打开操作。若文件成功打开并且文件指针没有到文件尾，则从文件中一次读取一个字符，并将该字符添加到一列单链表节点中，直至遇到换行符（ASCII码10）或连续读取字符个数大于76（在此文件编辑器中，每行最多为76个字符）。当列单链表形成后，它的首地址将被保存至行单链表的相应节点的数据域中，如此动作，直至文件指针指向文件尾部而结束。

注意：由于本程序中每行以回车符（ASCII码为13）结束，而当用Windows的记事本创建一个文本文件，打开此文件并用fgetc()函数读取时，程序写入列单链表节点中的值是ASCII码为13的回车符。打开的文件编码应当为ANSI格式，在存入西文和符号时，将其存入txt[0]，\0存入txt[1]，存入汉字时，因为每个汉字占用两个字节，故将其存入txt[0]、txt[1]中，\0存入txt[2] ，以此来实现读取、储存和显示中文的功能。

（2）保存文件

保存文件操作主要完成将链表中的数据写入文件中的任务，它的具体实现流程如下。

1>用户输入一个保存此单链表数据的文件名。

2>读取行单链表中的节点数据域的值，若值不为空，则执行步骤（3）；否则执行步骤（5）。

3>依次读取行单链表节点中保存的首地址的相应列单链表节点的数据域的值，将其值写入文件中，直至该列单链表中指针域为NULL的最后一个元素结束。

4>读取行单链表中的下一个节点，并返回步骤（2）。

5>关闭文件，退出。

（3）新建文件

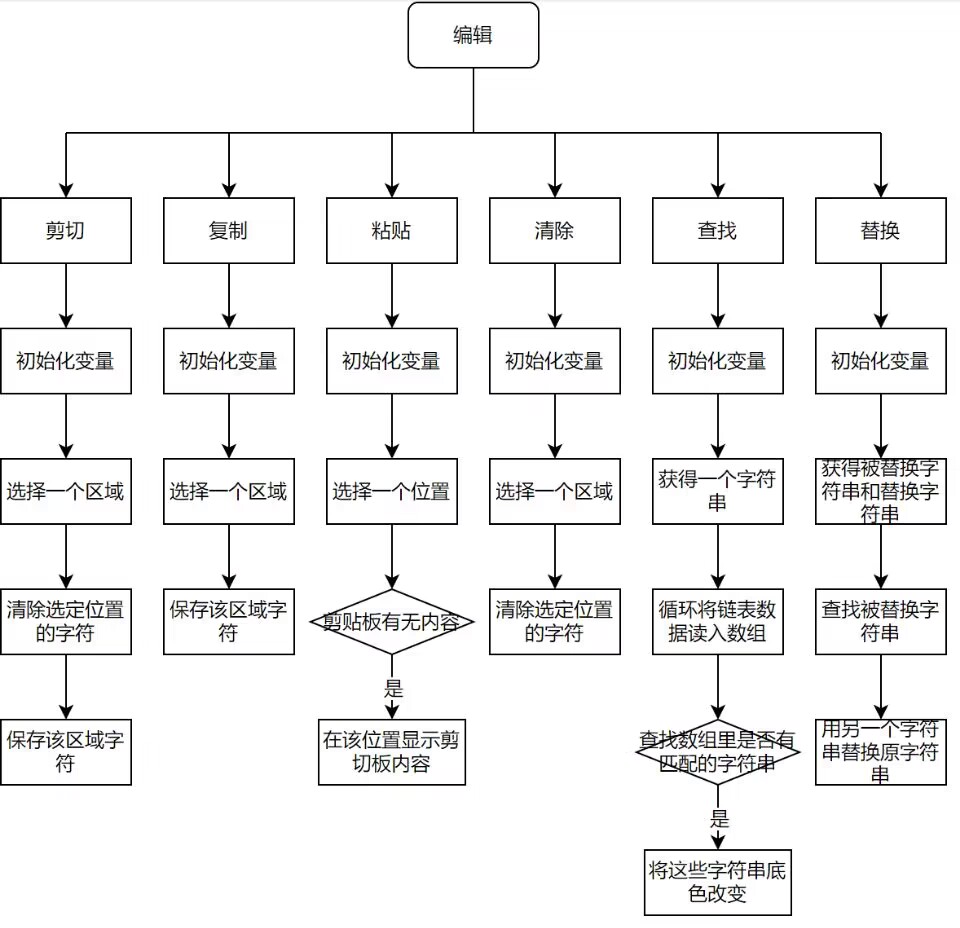
文件的新建过程中，它首先提示用户输入要新建文件的文件名，以 “wb+”打开文件。若文件不存在，即可完成新建操作，若文件存在，则文件新建失败，将在短暂等待（2s）后调用打开文件的函数，打开名字为此的文件。

（4）另存为文件

该功能实现方法与保存类似，使用wb+即可新建一个文件，只需要在新建文件后采用保存文件的思路，读取行单链表中的节点数据域的值，若值不为空，则依次读取行单链表节点中保存的首地址的相应列单链表节点的数据域的值，将其值写入文件中，直至该列单链表中指针域为NULL的最后一个元素结束即可。

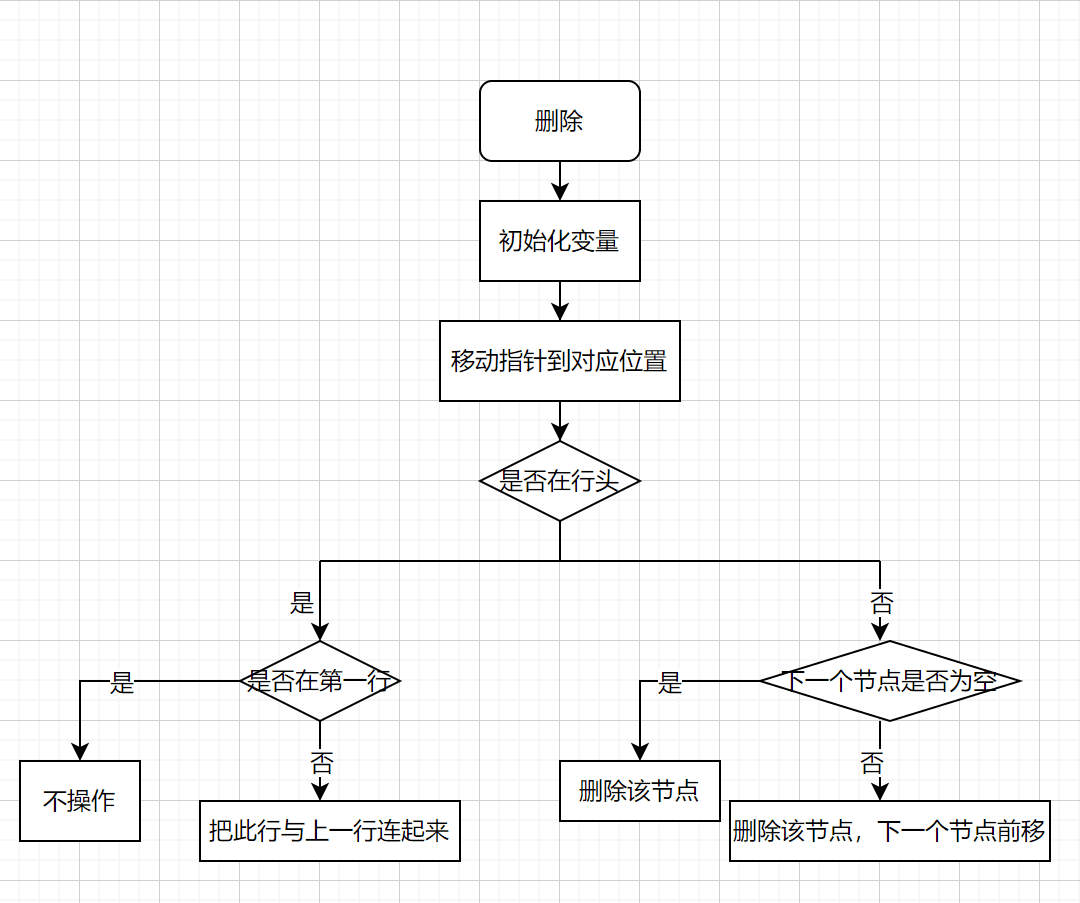
**3.文件编辑模块**

在文件编辑模块中，主要完成以添加或插入的方式输入字符、删除光标所在的当前位置或前一个位置的单个字符、朝上下左右4个方向的光标的移动操作。下面介绍这4个功能的具体设计与实现。



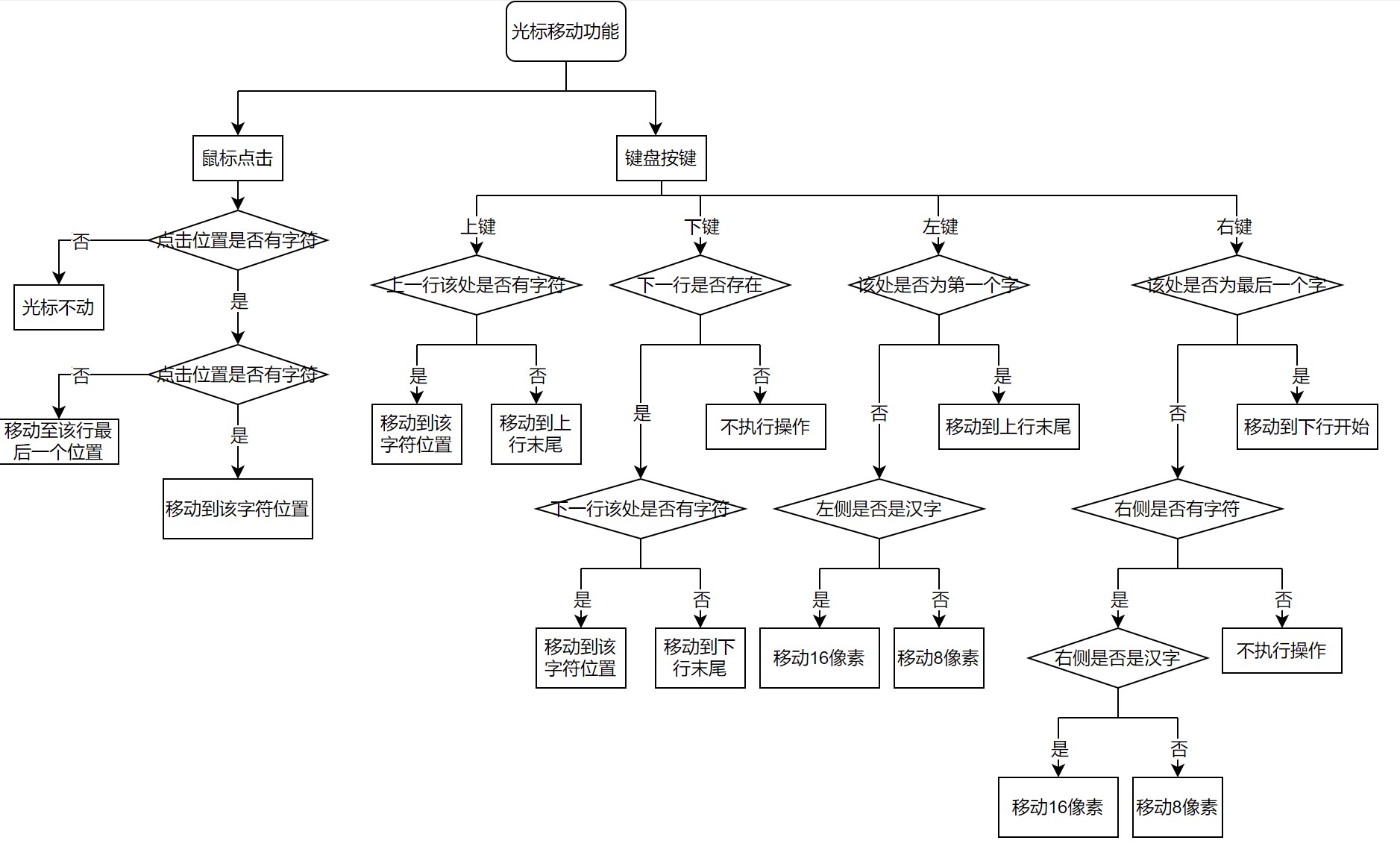
1.添加与插入字符  
用户按下常规键的时候，调用insert函数在光标位置插入按下的字符。用户按下F4键的时候，进入循环调用汉字输入法函数，直到按下ESC键退出。

2.删除字符



当用户按下Del键时，系统会调用delte()函数在单链表中删除当前光标所在处的字符；当用户按下BackSpace键时，系统会先移动光标再调用这个函数在单链表中删除之前光标所在处前一个位置的字符。

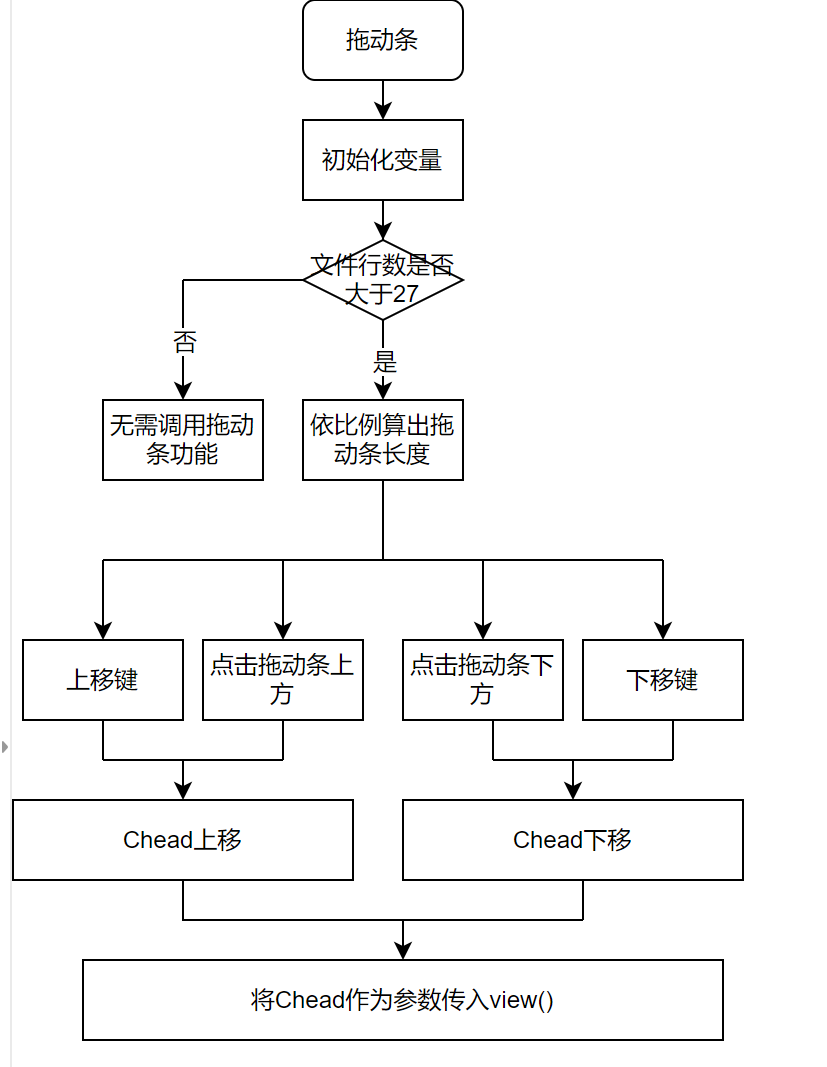
3.移动光标



移动光标可以通过鼠标点击或者键盘的上下键完成。

具体操作为：鼠标点击后判断当前位置有没有字符，如果有就移动光标到这个位置

如果该位置没有字符，但该行有，则移动光标到该行末尾



4.翻页与进度条

翻页的操作主要通过更改Chead指向的位置来实现。当删除到目前显示的第一行开头时，当按上下键到顶、底时，当点击左右边的滑动条和箭头时，分别会触发不同的翻页（切换显示的行）效果。点击箭头和上下键到限度都是向上或者向下移动一格Chead，而点击滑动条则会根据比例移动Chead。

**4.剪贴板操作模块**

在剪切板操作模块中，主要完成对已选定文本的剪切、复制和粘贴工作，因此剪贴板操作与文本选定工作密切相关。下面分别介绍文本的选定和对选定文本的剪切、复制和粘贴操作的具体实现。

（1）选定文本

用户可按Ctrl+<-或Ctrl+->来选定文本，就具体实现而言，两者基本相同。在介绍选定文本的实现过程之前，先简要介绍一个全局的字符数组rcd[]，它的元素的类型为char类型，每一个元素可保存一个字符的字符值。其文本选定的实现过程如下：

1>当用户按下Ctrl+<-或Ctrl+->键时，程序将当前光标位置向左或向右移动一个位置,。再调用Colorview()函数，用将当前的文本背景改成灰色，以达到突出选定文本的效果。

（2）剪切

用户可按Ctrl+X键或通过Edit菜单选项或通过右键单击的弹出框来剪切选定的文本，若之前没有选定的文本，此按键无效。它的实现过程如下：

1> 如果是左键选中，则先将光标移动到选中位置的右侧

2> 循环调用delete函数删除选中的字符

3>改变记录选定文本的变量rcdflg;

（3）复制

用户可按Ctrl+C键或通过Edit菜单选项或通过右键单击的弹出框来复制选定的文本，复制操作的实现比剪切操作简单，只需要改变rcdflg再重新view取消选中效果即可

（4）粘贴

用户可按Ctrl+V键或通过Edit菜单选项或通过右键单击的弹出框，完成粘贴操作。这一操作必须在剪切或复制操作之后出现。它的具体实现过程如下：

1. rcdflg的值非0（非0表示已经进行过剪切或复制操作）中，则执行下面的操作。
2. 如果rcdsign==2，表示是从右到左选中的，则将字符串导致使其正序
3. 循环调用insert将剪贴板内容插入当前光标位置。
4. **BC功能模拟模块**

BC功能模拟模块主要由代码高亮、自动匹配括号组成。

（1）代码高亮

代码高亮用于匹配保留字并用高亮显示，其功能实现如下

highlight函数定义了保留字，循环调用changecolor来查找并改变每个关键字的颜色。Changecolor执行查找到特定的字符串，如果关键之前后没有字符，则循环调用changflag来改变保留字节点的颜色值。

（2）自动匹配括号

每次进行输入之后都调用match函数检测输入的字符是否是需要匹配的括号和引号，如果是，则调用insert函数将另一半也插入。

（3）搜索

搜索功能可以在软件获取用户的搜索词后，在文本窗口中以选中的方式把文本中所有的搜索词标记。具体的实现步骤为首先将一行的链表数据连接成一个字符串。在字符串起开始使用strstr函数进行查找，在查找到搜索词后将循环调用colorview函数改变查找字的背景色。当一行检测结束后，便录入下一行节点，直到检查完所有的行为止。

（4）替换

替换的操作与查找较相似，先查找到特定字符串，再循环调用delete删除字符串，接着循环调用insert函数插入新的字符串，之后对字符数组也用字符串操作函数进行同样的操作，进行下一个检测

1. 帮助模块

帮助模块主要用于提示用户如何使用本软件，它的实现非常简单。同样，文本的快速预览模块是在原来主窗口显示模块的基础上，去除了菜单的显示。主窗口主要由菜单栏、文本编辑区和状态栏3大部分构成。菜单栏用来显示菜单项，文本编辑区主要用来完成文本字符的输入、删除等操作，状态栏主要用来显示当前光标在文本窗口中的坐标值。它主要利用文本窗口的gotoxy()函数和cprintf()函数来实现。

1. 汉化模块

本程序中汉化模块是融入到很多版块中的，由于程序使用VGA图形界面进行显示，故可以方便显示汉字。本程序只支持GBK编码汉字，打开文件时，会判断文件内容是否是GBK编码的汉字，并进行对应的储存到链表的操作，View函数会判断节点是字符还是汉字，并分别调用outtextxy和puthz来进行显示。删除时会判断是字符还是汉字并进行对应的操作。按下F4键可以进入汉字输入模式，调用hz\_put函数进行输入，按Esc键退出汉字输入模式。考虑到编辑代码对注释的编辑要求不是特别高，汉字只支持显示，输入和删除，无法进行剪贴复制等操作。

# 第五部分 算法设计

## 数据结构设计

本程序共定义了2个结构体，分别与列单链表、行单链表相关。下面分别介绍这个结构体。

（1）与列单链表相关的node结构体

typedef struct node

{

char txt[3]; //用于储存节点的内容

int flag; //用于记录该节点的颜色

unsigned short sign; //用于指示节点储存的是中文还是英文：英文为0，汉字为1。

struct node\* next; //指向下一个结构体的指针

}node;

node结构体定义了在一个单链表中保存行中的单个字符的结构，我们称由node类型的节点构成的单链表为列单链表。结构体中各字段表示的意义如下：

char txt[3]; //用于储存节点的内容

int flag; //用于记录该节点的颜色

unsigned short sign; //用于指示节点储存的是中文还是英文：英文为0，汉字为1。

（3）与行单链表相关的Hnode结构体

typedef struct Hnode /\*定义保存所有列单链表首节点的指针的结构\*/

{

node\* next; /\*指向列单链表的首节点的地址\*/

struct Hnode\* nextl; /\*指向下一个节点的指针\*/

}Hnode;/\* 由此类型节点构成的单链表，命名为：行单链表\*/

Hnode结构体定义了在一个单链表中保存列单链表首节点地址的结构，我们称由Hnode类型的节点构成的单链表为行单链表。结构体中各字段表示的意义如下：

node \*next：数据域，指向列单链表的首节点的地址。

struct Hnode\* next1：指针域，指向列单链表中的下一个节点。

在主循环中使用两个指针Hhead、Chead，Hhead指向整个文本的开头行，Chead指向显示的文本的开头行。

二、算法说明

**1.文本操作**

**综述：**

为了插入、删除字符方便，采用链表方式来存储数据。考虑到需要存放中文注释的需要，数据域采用char[3]的数据类型来进行存储，存储的字符编码为GBK格式，中文占两个字符位。英文占一个字符位。为了实现代码高亮，应用int型变量flag来存放该字符的高亮颜色。为了优化索引时长，引入行单链表来作为链表的索引，指向每行的第一个节点：

在引入行单链表后，一般情况下，各个列单链表指向其下一个节点，但是行末的节点不会与下一层的节点相连，这样对链表节点作分层，可以使结构显得简洁明了，同时使得算法编写需要的代码量大大减少了。

**新建、打开、保存、另存文件：**

打开文件的实质是通过链表读取文件里的字符，将其全部保存到链表里后通过view()函数将其显示到屏幕上，打开失败会输出提示。新建与之类似，只是多了一个判断，当输入的文件名不存在时则新建一个文件，若文件已存在则直接打开这个文件。

保存与另存为实现方法类似，其基本逻辑是键盘键入会修改链表，只需要将新的链表传入文件中即可完成保存操作，只要再加入一个修改文件名或文件路径的函数即可实现文件另存为的目的。

**插入字符、删除字符：**

程序实际上是将屏幕输入区划分成8\*16像素的格子，一个字符占一格，一个汉字占两格，但是二者都会保存在一个节点里，因此在输入删除的时候需要判断节点及其之前的位置是汉字还是字符从而正确移动节点。具体解释如下：

插入字符时，会先调用循环将节点指针循环移动到插入位置的前一个节点，考虑到汉字占两格的影响，会在指针移动之后检测是否是汉字并进行少移一位的操作。

删除时，删除字符和汉字两种情况，先将指针循环移动到光标位置的前两个节点的位置，判断要删除的位置是汉字还是字符，如果是汉字，则删除指针位置的汉字，如果是字符就后移一个节点再删除。

**搜索、代码高亮、替换：**

考虑到strstr函数可以用于字符串匹配，strrcmp函数可以用来作字符复制。将一行的字符保存在一个数组中，而后应用strstr函数，循环查找每一行与目标字符串匹配的字符串，如果找到，则移动指向这一行行头的字符指针到该字符串后的位置，查找这一行的下一个此字符串，如果返回空指针，则说明此行没有特定字符串了，循环到下一行找。

对于高亮，考虑到代码高亮需要将所有保留字染色，因此执行32次上述过程，将所有保留字进行染色。

替换与查找略有不同，在查找到字符串之后，进行链表操作以后，需要将该字符串在总字符串中替换，先用strcat将该字符串后面的内容连在新字符串后面，再用strcpy将新串复制到需要替换的字符串位置，就实现了字符串中的替换，再进入下一次循环进行下一个字符串的查找替换。

**复制、剪切、粘贴：**

在选中文本的时候字符就已经录入了rcd数组，为了与复制粘贴功能相契合，于是使用两个变量来记录剪贴板的状态。

利用字符数组rcd来记录选中的内容，rcdflg记录是否使用复制剪贴，rcdsign用于记录是左选还是右选。，粘贴时先检测rcdflg，判断有无使用复制剪切，再判断rcdsign，如果是左选需要进行字符串倒置过程。

**2.界面显示**

**显示文本：**

按行将列链表中的值输出到屏幕。利用数组来保存各个flag对应的颜色，每个字符输出之前将颜色调整好，在输出全部完成之后将光标颜色调回白色。当读到行末的时候跳转到下一行继续输出。

**选中：**

在按Ctrl+←、Ctrl+→可以进行选中操作。选中之后将选中的部分单独重新显示。

**翻页、显示行数：**

由于文本框的大小只能容纳27行文本，需要进行翻页的操作，也就是更改显示的第一个行单链表。行链表的头结点称为Hhead。在行链表头结点之外新建一个行链表指针，称作Chead。用Chead指向的位置来指定当前应当显示的第一行。因为屏幕大小的限制，只会显示Chead指向的行链表及其后26行的文本。在传递参数给view函数等函数时，将Chead代替Hhead传入。这种设计，不需要进行大量代码重构，只需要更改传入的参数和判断条件即可。

翻页可以通过键盘的上下按键和点击右侧的上下箭头和滑动条来完成。上下按键用来进行光标跳转，到顶或底时将Chead向上或下移位。上下箭头则不进行光标跳转，直接将Chead移位。

点击滑动条时，注意到滑动条可移动的范围有399\*16的像素块长，一个页面可以显示27行代码。为了让滑动条能够指示当前Chead的位置，我们运用了如下算法：

记总行数为r，参数count用于记录上下移动的次数（上移一次-1，下移一次+1），x用于表示滑动条的长度，于是我们不难得到如下关系：

x=(int)10773/r ； （10773=399\*27）

滑动条上方纵坐标y1\_skbar =(int)(41+count\*399/r)；

滑动条下方纵坐标y2\_skbar= (int)(41+count\*399/r+x)；

使用上移\下移按钮时逻辑较为简单，使用链表移动Chead到指定位置即可，再对count进行修改，利用view()显示链表，

若点击滑动条空白部分，则需要记录鼠标点击位置的纵坐标，即MouseY，记录其与当前滑动条上或下端纵坐标（y1\_skbar和y2\_skbar）的差值即可确定移动量，上下分别用change\_count1，change\_count2表示，于是我们有了一下算法得到其值，再使用循环操作显示链表即可。

y1\_skbar=(int)(41+x+count\*399/r-x);

y2\_skbar=(int)(41+x+count\*399/r);

change\_count1=(int)((y1\_skbar-MouseY)\*r/399);

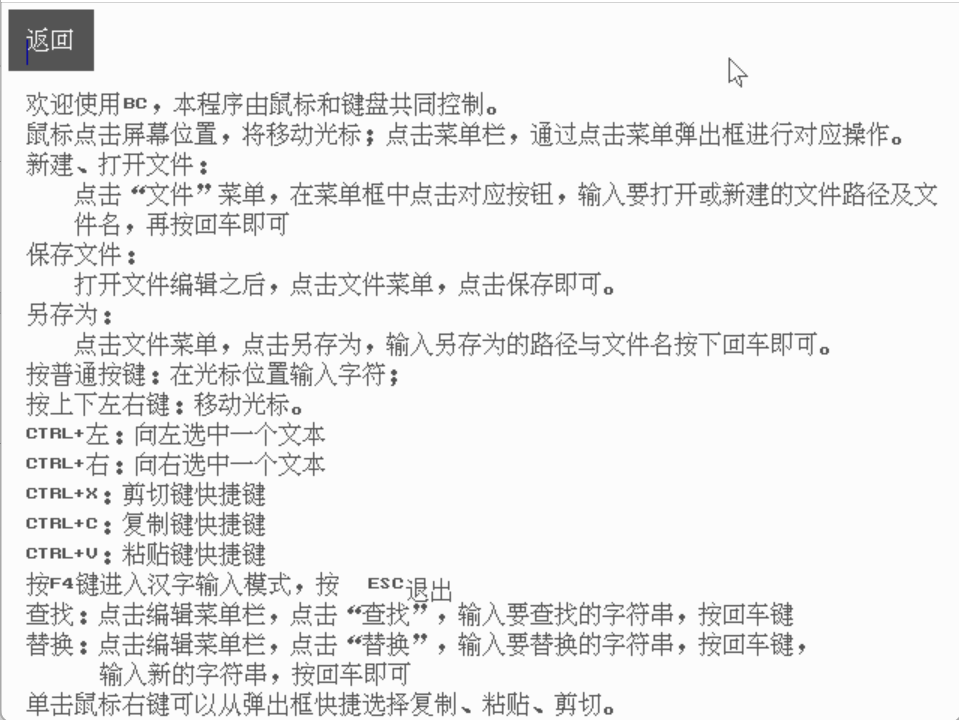
change\_count2=(int)((MouseY-y2\_skbar)\*r/399);

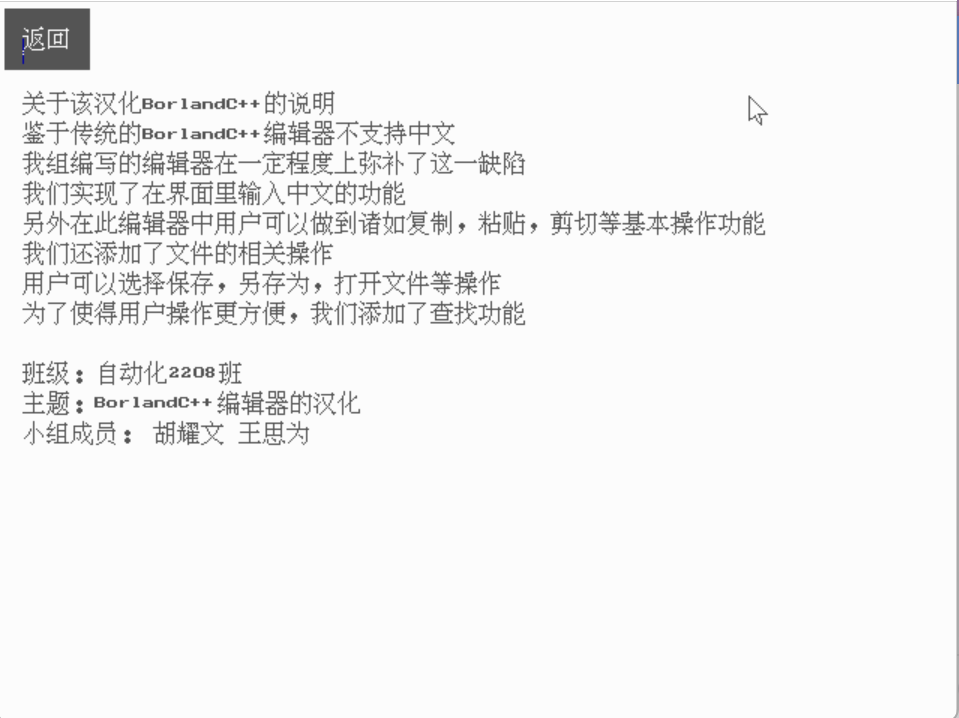
# 第六部分 界面设计

## 界面设计

1. 主界面

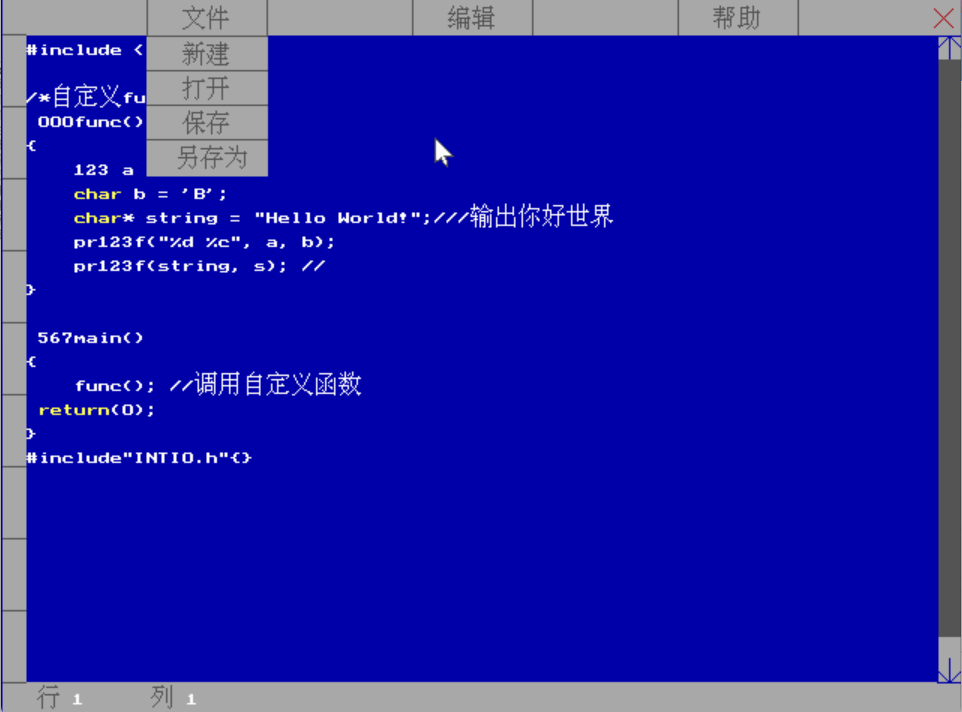
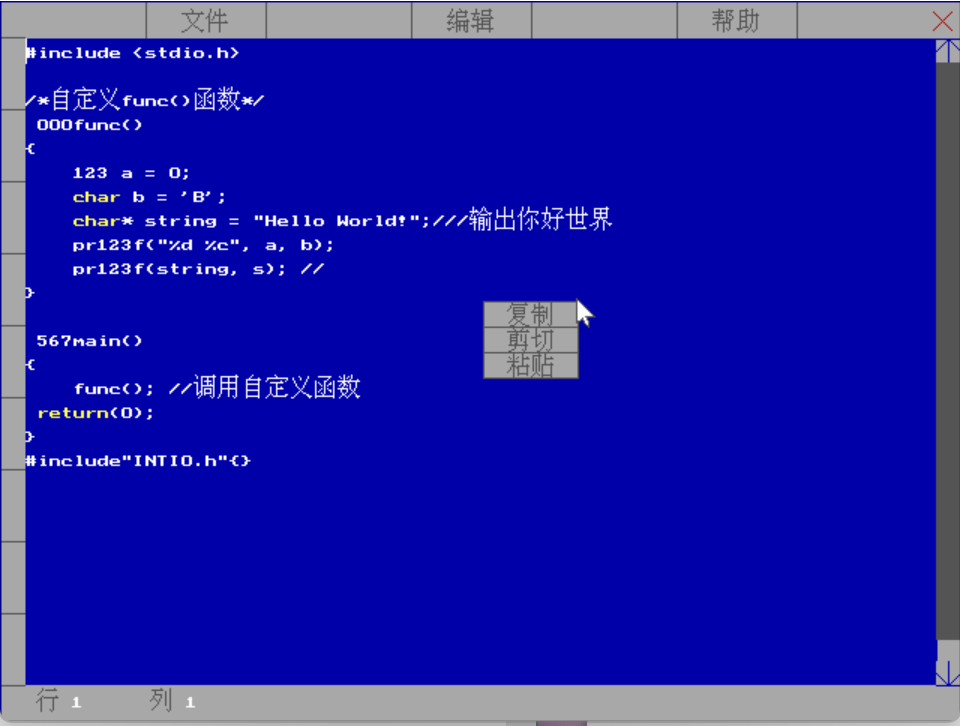
考虑到BC的文本窗口不能有效显示汉字，主界面使用的是BorlandC的图形窗口，模式为VGA模式。文本的显示全部调用BC的图形函数绘制的。该界面的可以执行程序编辑文本的主要任务。

1. 帮助界面



1. 菜单与快捷右键

在菜单界面中，用户可以通过鼠标来点选对应菜单项



# 第七部分 函数说明及函数原型 函数说明及函数原型

1. void drawmenu();

传入参数：无

功能：画主菜单

1. void drawfile() ;

传入参数：无

功能：画文件菜单

1. void drawedit();

传入参数：无

功能：画编辑菜单

1. void drawhelp();

传入参数：无

功能：画帮助菜单

1. void drawrc(int r, int c);

传入参数：行r，列c

功能：画出当前行列位置

1. void drawwin(int x,int y);

传入参数：

功能：绘制小窗弹出框

1. void full\_skbar();

传入参数：无

功能：画一个占满位置的滑动条

1. void drawcursor();

传入参数：无

功能：画光标

1. void delcursor();

传入参数：无

功能：删除光标

1. void movecursor(int x,int y);

传入参数：位置x，y

功能：移动光标

1. void getcsim();

传入参数：无

功能：获取光标处图像

1. void newcursor();

传入参数：无

功能：更新光标

1. int save(Hnode\* Hp) ;

传入参数：行链表指针Hp

功能：保存文件

1. int saveas(Hnode\* Hp);

传入参数：行链表指针Hp

功能：另存文件

1. int FreeAll(Hnode\* p);

传入参数：行链表指针p

功能：删除p及后面的行节点和节点

1. int Freenode(node\* p) ;

传入参数：链表指针p

功能：删除p及后面的节点

1. Hnode\* opens(Hnode\* Hp,Hnode \*\*Chead);

传入参数：行链表指针Hp，二级指针Chead

功能：打开文件

1. int newbuild(Hnode\* Hp,Hnode \*\*Chead);

传入参数：行链表指针Hp，二级指针Chead

功能：新建文件

1. void find(Hnode\* Chead);

传入参数：行链表指针Chead

功能：查找

1. void replace(Hnode\* Chead) ;

传入参数：行链表指针Chead

功能：替换

1. void ctrlr(int rcdlen,Hnode\* Chead,char \*rcd,int m,int n);

传入参数：整型变量redlen，字符rcd，位置m，n，链表Chead

功能：向右选中一个字符

1. void ctrll(int rcdlen,Hnode\* Chead,char \*rcd,int m,int n);

传入参数：整型变量redlen，字符rcd，位置m，n，链表Chead

功能：向左选中一个字符

1. int turnp(char \*rcd,int rcdlen);

传入参数：剪切板数组，剪切板当前长度

功能：向左选字符后倒序

1. int colorview(int m,int n,Hnode\* Chead);

传入参数：位置m，n，行链表指针Chead

功能：改变字符底色

1. int highlight(Hnode \*Chead);

传入参数：行链表指针Chead

功能：代码高亮

1. int changecolor(char\* str,int slen,Hnode \*Chead);

传入参数：特定字符串，长度，行链表指针

功能：查找并改变关键字颜色

1. int changeflag(int m,int n,Hnode \*Chead,int color);

传入参数：位置m，n，链表指针Chead，颜色color

功能：改变m行n列颜色

1. void help();

传入参数：无

功能：画帮助页面

1. void about();

传入参数：无

功能：画关于页面

1. int insert(char\* bioget,int m,int n,Hnode\* Chead);

传入参数：字符bioget，位置m，n，指针Chead

功能：插入

1. void enter(int m,int n,Hnode\* Chead);

传入参数：位置m，n，指针Chead

功能：换行

1. int deltext(Hnode\* Chead,int r,int c);

传入参数：行r，列c，指针Chead

功能：删除文本

1. int del(int flg,node\* p);

传入参数：链表p，整形flg

功能：删除

1. int movepic(int x1,int y1,int l,int x2);

传入参数：位置x1，y1，l，x2

功能：将x1,y1处长为l的列图像保存，再放到x2,y2处

1. int getlen(node\* p);

传入参数：链表p

功能：检查传入的指针后面有几个节点

1. void match(char\* bioget,Hnode\* Chead);

传入参数：链表Chead

功能：用于检测括号引号并进行相应的匹配

1. void mainloop();

传入参数：无

功能：主循环

1. int chosedit(Hnode\* Chead,char\* bioget,char \*rcd);

传入参数：链表Chead，字符bioget，rcd

功能：响应点击编辑菜单栏后的操作

1. int chosfile(Hnode \*Hhead,Hnode \*\*Chead);

传入参数：链表Hhead，二级指针Chead

功能：响应点击文件菜单栏后的操作

1. int choshelp();

传入参数：无

功能：响应点击帮助菜单栏后的操作

1. void drawskbar(Hnode\* Hhead);

传入参数：列链表指针Hhead

功能：画滑动条

1. int up\_page(Hnode\* Hp,Hnode \*\*Chead);

传入参数：列链表指针Hp，二级指针Chead

功能：显示上一行

1. int down\_page(Hnode\* Hp,Hnode \*\*Chead);

传入参数：列链表指针Hp，二级指针Chead

功能：显示下一行

1. int newskbar(Hnode\* Hhead,int r);

传入参数：列链表指针Hhead，行数r

功能：画移动后的滑动条

1. int click\_skbar(Hnode\* Hhead,Hnode \*\*Chead);

传入参数：列链表指针Hhead，二级指针Chead

功能：点击滑动条使得页面上下移动

1. void view(Hnode\* Chead);

传入参数：列链表指针Chead

功能：显示链表中存有的字符

1. int checktxt(int x,int y,Hnode\* Chead);

传入参数：坐标，列链表指针Chead

功能：检查x,y位置有没有字符

1. int gettxtname(char\* s,int x,int y);

传入参数：字符串s，位置x，y

功能：输入一个字符串

1. int getlinelen(Hnode \*q);

传入参数：列链表指针q

功能：得到一行的长度

1. int getHlen(Hnode \*Hhead);

传入参数：链表Hhead

功能：获得链表行数

**第八部分 源代码**

1．Main.c

#include<common.h>

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*【main函数】*

*编写人：胡耀文*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**void** main**()**

**{**

**int** gd**=**VGA**;**

**int** gm**=**VGAHI**;**

initgraph**(&**gd**,&**gm**,"C:\\BORLANDC\\BGI");**

cleardevice**();**

mouseinit**();**

mainloop**();**

**return** 0**;**

**}**

2.mianloop.c

#include<common.h>

#include "mainloop.h"

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*【主循环函数】*

*编写人：胡耀文*

*mainloop为主循环函数*

*chosedit响应点击编辑菜单栏后的操作*

*chosfile响应点击文件菜单栏后的操作*

*choshelp响应点击帮助菜单栏后的操作*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**char** rcd**[**80**];**

**int** rcdlen**=**0**;***//用于储存剪贴板的长度*

**int** rcdflg**=**0**;** *//用于指示剪贴板有没有存东西*

**int** rcdsign**=**0**;***//用于指示剪贴板存的是左选（2）还是右选（1）*

**extern** **int** cursorX**;**

**extern** **int** cursorY**;**

**extern** **void** **\***bitmap**;**

**extern** **int** imgflag**;**

*//主循环函数*

**void** mainloop**()**

**{**

*//初始化变量 ，链表*

**int** ch**,**rowy**=**1**,**clmx**=**1**,**o**,**i**=**0**,**mx**,**my**,**j**,**sig**;**

**unsigned** s**=**imagesize**(**1**,**1**,**1**,**8**);***//用于储存光标处的图像*

**char** a**;**

**char** bioget**[**2**];***//用于存放当前的键值*

**void** **\***bitmap2**;**

Hnode**\*** Hhead**,\***q**,\***Chead**,\***temp**;**

node**\*** p1**,**p2**;**

Hhead**=(**Hnode**\*)**malloc**(sizeof(**Hnode**));***//为行单链表首节点分配空间*

q**=**Hhead**;** Hhead**->**nextl**=**NULL**;** q**->**next**=**NULL**;**

p1**=**q**->**next**=(**node**\*)**malloc**(sizeof(**node**));***//为列单链表中的首节点分配空间*

p1**->**txt**[**0**]=**13**;**p1**->**txt**[**2**]=**'\0'**;**p1**->**sign**=**0**;**p1**->**next**=**NULL**;***//初始化第一个节点*

Chead**=**Hhead**;**

setbkcolor**(**BLUE**);**

drawmenu**();**

full\_skbar**();**

*//主循环*

**while(**1**)**

**{**

newmouse**(&**MouseX**,&**MouseY**,&**press**);**

memset**(**bioget**,**'\0'**,**2**);**

rowy**=(**cursorY**-**24**)/**16**+**1**;**

clmx**=(**cursorX**-**16**)/**8**+**1**;**

drawrc**(**rowy**,**clmx**);**

**if(**imgflag**==**0**)***//如果还没有储存光标的图像，就储存，防止光标掩盖屏幕内容*

**{**

clrmous**(**MouseX**,**MouseY**);**

getcsim**();**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

imgflag**=**1**;**

**}**

drawcursor**();***//绘制光标*

*//判断鼠标点击*

**if(**mouse\_press**(**9**,**25**,**631**,**455**)==**3**)***//右键点击*

**{**

bitmap2**=**malloc**(**imagesize**(**1**,**1**,**64**,**52**));**

**if(**MouseX**>**17**&&**MouseX**<=**304**&&**MouseY**>**25**&&**MouseY**<=**216**)***//左上角*

**{**

mx**=**MouseX**;**

my**=**MouseY**;**

clrmous**(**MouseX**,**MouseY**);**

getimage**(**mx**,**my**,**mx**+**63**,**my**+**51**,**bitmap2**);**

drawwin**(**mx**,**my**);**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

**}**

**else** **if(**MouseX**>**304**&&**MouseX**<**624**&&**MouseY**>**25**&&**MouseY**<=**216**)***//右上角*

**{**

mx**=**MouseX**-**63**;**

my**=**MouseY**;**

clrmous**(**MouseX**,**MouseY**);**

getimage**(**mx**,**my**,**mx**+**63**,**my**+**51**,**bitmap2**);**

drawwin**(**mx**,**my**);**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

**}**

**else** **if(**MouseX**>**17**&&**MouseX**<=**304**&&**MouseY**>**216**&&**MouseY**<**456**)***//左下角*

**{**

mx**=**MouseX**;**

my**=**MouseY**-**51**;**

clrmous**(**MouseX**,**MouseY**);**

getimage**(**mx**,**my**,**mx**+**63**,**my**+**51**,**bitmap2**);**

drawwin**(**mx**,**my**);**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

**}**

**else** **if(**MouseX**>**304**&&**MouseX**<**624**&&**MouseY**>**216**&&**MouseY**<**456**)***//右下角*

**{**

mx**=**MouseX**-**63**;**

my**=**MouseY**-**51**;**

clrmous**(**MouseX**,**MouseY**);**

getimage**(**mx**,**my**,**mx**+**63**,**my**+**51**,**bitmap2**);**

drawwin**(**mx**,**my**);**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

**}**

**while(**1**)**

**{**

newmouse**(&**MouseX**,&**MouseY**,&**press**);**

**if(**mouse\_press**(**mx**,**my**,**mx**+**63**,**my**+**17**)==**1**)***//点击复制*

**{**

putimage**(**mx**,**my**,**bitmap2**,**0**);**

free**(**bitmap2**);**

delcursor**();**

view**(**Chead**);**

rcdflg**=**1**;**

imgflag**=**0**;**

**break;**

**}**

**else** **if(**mouse\_press**(**mx**,**my**+**18**,**mx**+**63**,**my**+**34**)==**1**)***//点击剪切*

**{**

putimage**(**mx**,**my**,**bitmap2**,**0**);**

free**(**bitmap2**);**

**if(**rcdsign**==**2**)**

movecursor**(((**cursorX**-**16**)/**8**+**1**)+**rcdlen**,(**cursorY**-**24**)/**16**+**1**);***//如果是从右到左选择的 ,删除前先移动光标*

**for(**i**=**0**;**i**<**rcdlen**;**i**++)**

**{**

deltext**(**Chead**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**));**

**}**

view**(**Chead**);**

rcdflg**=**1**;**

**break;**

**}**

**else** **if(**mouse\_press**(**mx**,**my**+**35**,**mx**+**63**,**my**+**63**)==**1**)***//点击粘贴*

**{**

putimage**(**mx**,**my**,**bitmap2**,**0**);**

free**(**bitmap2**);**

**if(**rcdflg**==**1**)** *//如果剪贴板有内容*

**{**

**if(**rcdsign**==**2**)**

turnp**(**rcd**,**rcdlen**);**

**for(**i**=**0**;**i**<**rcdlen**;**i**++)**

**{**

bioget**[**0**]=**rcd**[**i**];**

insert**(**bioget**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**),**Chead**);**

**}**

memset**(**rcd**,**'\0'**,**10**);**

**}**

view**(**Chead**);**

rcdlen**=**0**;**

**break;**

**}**

**else** **if((**mouse\_press**(**17**,**25**,**mx**-**1**,**my**+**51**)==**1**)||(**mouse\_press**(**mx**-**1**,**25**,**624**,**my**)==**1**)**

**||(**mouse\_press**(**17**,**my**+**52**,**mx**+**64**,**456**)==**1**)||(**mouse\_press**(**mx**+**64**,**my**,**624**,**456**)==**1**))***//点击其他区域*

**{**

clrmous**(**MouseX**,**MouseY**);**

putimage**(**mx**,**my**,**bitmap2**,**0**);**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

free**(**bitmap2**);**

**break;**

**}**

**}**

**}**

**else** **if(**mouse\_press**(**9**,**25**,**615**,**455**)==**1**)***//如果鼠标点击屏幕位置，移动光标到鼠标的位置*

**{**

**if((**rcdlen**!=**0**)&&(**rcdflg**==**0**))**

**{**

rcdlen**=**0**;**

view**(**Chead**);**

**}**

i**=** checktxt**(((**MouseX**-**16**)/**8**+**1**),((**MouseY**-**24**)/**16**+**1**),**Chead**);**

**if(**i**==**1**)***//如果返回值是1，就移动光标 到字符位置*

**{**

clrmous**(**MouseX**,**MouseY**);**

delcursor**();**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

cursorX**=(**MouseX**-**16**)/**8**\***8**+**17**;**

cursorY**=(**MouseY**-**24**)/**16**\***16**+**25**;**

imgflag**=**0**;**

**}**

*// else if(i==2)//如果返回值是2，移动光标到这一行最后一个位置*

*// {*

*// q=Chead;*

*// for(j=1;j<((MouseY-24)/16+1);j++)*

*// q=q->nextl;*

*// clrmous(MouseX,MouseY);*

*// delcursor();*

*// movecursor(getlinelen(q)+1,((MouseY-24)/16+1));*

*//*

*// imgflag=0;*

*// save\_bk\_mou(MouseX,MouseY);*

*// drawmous(MouseX,MouseY);*

*// continue;*

*// }*

**}**

**else** **if(**mouse\_press**(**616**,**25**,**640**,**41**)==**1**)**

**{**

up\_page**(**Hhead**,&**Chead**);**

view**(**Chead**);**

**}**

**else** **if(**mouse\_press**(**616**,**440**,**640**,**456**)==**1**)**

**{**

down\_page**(**Hhead**,&**Chead**);**

view**(**Chead**);**

**}**

**else** **if(**mouse\_press**(**616**,**40**,**639**,**440**)==**1**)**

**{**

click\_skbar**(**Hhead**,&**Chead**);**

view**(**Chead**);**

**}**

**else** **if(**mouse\_press**(**616**,**0**,**640**,**24**)==**1**)***//点击关闭按钮结束程序*

**{**

closegraph**();**

exit**(**0**);**

**}**

**else** **if(**mouse\_press**(**97**,**1**,**177**,**24**)==**1**)***//点击文件按钮*

**{**

chosfile**(**Hhead**,&**Chead**);**

Hhead**=**Chead**;**

highlight**(**Chead**);**

view**(**Chead**);**

clrmous**(**MouseX**,**MouseY**);**

drawskbar**(**Hhead**);**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

**}**

**else** **if(**mouse\_press**(**274**,**1**,**354**,**24**)==**1**)***//点击编辑按钮*

**{**

chosedit**(**Chead**,**bioget**,**rcd**);**

**}**

**else** **if(**mouse\_press**(**451**,**1**,**531**,**24**)==**1**)***//点击帮助按钮*

**{**

delcursor**();**

choshelp**();**

clrmous**(**MouseX**,**MouseY**);**

drawmenu**();**

drawskbar**(**Hhead**);**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

view**(**Chead**);**

imgflag**=**0**;**

**}**

*//处理键盘输入事件*

**if** **(**bioskey**(**1**))***//检测键盘输入*

**{**

a**=**ch**=**bioskey**(**0**);***//获取键入的键值*

**if(**ch**==**0x4B00**&&**cursorX**>=**17**)***//光标左移*

**{**

**if(**cursorX**!=**17**)**

**{**

delcursor**();**

cursorX**-=**8**;**

imgflag**=**0**;**

**}**

**else**

**{**

q**=**Chead**;**

**for(**j**=**1**;**j**<((**cursorY**-**24**)/**16**);**j**++)**

q**=**q**->**nextl**;**

delcursor**();**

movecursor**(**getlinelen**(**q**)+**1**,((**cursorY**-**24**)/**16**));**

imgflag**=**0**;**

**}**

**}**

**else** **if(**ch**==**0x4800**&&**cursorY**>**25**)***//光标上移*

**{**

i**=** checktxt**(((**cursorX**-**16**)/**8**+**1**),((**cursorY**-**24**)/**16**),**Chead**);**

**if(**i**==**1**)***//如果返回值是1，就移动光标 到上一行该位置*

**{**

delcursor**();**

cursorY**-=**16**;**

imgflag**=**0**;**

newcursor**();**

**}**

**else** **if(**i**==**2**)***//如果返回值是2，移动光标到上一行最后一个位置*

**{**

q**=**Chead**;**

**for(**j**=**1**;**j**<((**cursorY**-**24**)/**16**);**j**++)**

q**=**q**->**nextl**;**

delcursor**();**

movecursor**(**getlinelen**(**q**)+**1**,((**cursorY**-**24**)/**16**));**

imgflag**=**0**;**

newcursor**();**

**}**

**}**

**else** **if(**ch**==**0x4d00**&&**cursorX**<**616**)***//光标右移*

**{**

i**=** checktxt**(((**cursorX**-**16**)/**8**+**2**),((**cursorY**-**24**)/**16**+**1**),**Chead**);***//检测下一个位置是否有值*

**if(**i**==**1**)***//如果返回值1，右移光标*

**{**

delcursor**();**

cursorX**+=**8**;**

imgflag**=**0**;**

**}**

**else** **if(**i**==**2**)***//如果返回值2，且有下一行移动到下一行第一个位置*

**{**

q**=**Chead**;**

**for(**j**=**1**;**j**<((**cursorY**-**24**)/**16**+**2**);**j**++)**

q**=**q**->**nextl**;**

**if(**q**!=**NULL**)**

movecursor**(**1**,((**cursorY**-**24**)/**16**+**2**));**

**}**

**}**

**else** **if(**ch**==**0x5000**&&**cursorY**<**440**)***//光标下移*

**{**

i**=** checktxt**(((**cursorX**-**16**)/**8**+**1**),((**cursorY**-**24**)/**16**+**2**),**Chead**);**

**if(**i**==**1**)***//如果返回值是1，就移动光标 到下一行该位置*

**{**

delcursor**();**

cursorY**+=**16**;**

imgflag**=**0**;**

newcursor**();**

**}**

**else** **if(**i**==**2**)***//如果返回值是2，移动光标到下一行最后一个位置*

**{**

q**=**Chead**;**

**for(**j**=**1**;**j**<((**cursorY**-**24**)/**16**+**2**);**j**++)**

q**=**q**->**nextl**;**

delcursor**();**

movecursor**(**getlinelen**(**q**)+**1**,((**cursorY**-**24**)/**16**+**2**));**

imgflag**=**0**;**

newcursor**();**

**}**

**}**

**else** **if(**ch**==**F4**)***//按F4键进入汉语输入模式*

**{**

delcursor**();**

**while(**1**)**

**{**

i**=**hz\_put**(**cursorX**,**cursorY**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**),**Chead**);**

**if(**i**==**1**)break;**

cursorX**+=**16**;**

**}**

settextstyle**(**0**,**0**,**1**);**

imgflag**=**0**;**

newcursor**();**

**}**

**else** **if(**ch**==**F1**)***//按F1键打开文件菜单*

**{**

chosfile**(**Hhead**,&**Chead**);**

**}**

**else** **if(**ch**==**F2**)***//按F2键打开编辑菜单*

**{**

chosedit**(**Chead**,**bioget**,**rcd**);**

**}**

**else** **if(**ch**==**F3**)***//按F3键打开帮助菜单*

**{**

choshelp**();**

**}**

**else** **if(**ch**==**F10**)***//按F10键刷新链表*

**{**

highlight**(**Chead**);**

view**(**Chead**);**

**}**

**else** **if(**ch**==**0x1C0D**)***//回车键*

**{**

enter**(((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**),**Chead**);**

**}**

**else** **if(**ch**==**CR**)***//用ctrl+加右移键选中文本*

**{**

colorview**(**rowy**,**clmx**,**Chead**);**

ctrlr**(**rcdlen**,**Chead**,**rcd**,**rowy**,**clmx**);**

rcdlen**++;**

delcursor**();**

cursorX**+=**8**;**

imgflag**=**0**;**

rcdsign**=**1**;**

**}**

**else** **if(**ch**==**CL**)***//用ctrl+加左移键选中文本*

**{**

colorview**(**rowy**,**clmx**-**1**,**Chead**);**

ctrll**(**rcdlen**,**Chead**,**rcd**,**rowy**,**clmx**-**1**);**

rcdlen**++;**

delcursor**();**

cursorX**-=**8**;**

imgflag**=**0**;**

rcdsign**=**2**;**

**}**

**else** **if(**ch**==**Cc**)***//复制键*

**{**

delcursor**();**

view**(**Chead**);**

rcdflg**=**1**;**

imgflag**=**0**;**

**}**

**else** **if(**ch**==**Cx**)***//剪切键*

**{**

**if(**rcdsign**==**2**)**movecursor**(**clmx**+**rcdlen**,**rowy**);***//如果是从右到左选择的 ,删除前先移动光标*

**for(**i**=**0**;**i**<**rcdlen**;**i**++)**

**{**

deltext**(**Chead**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**));**

**}**

view**(**Chead**);**

rcdflg**=**1**;**

sig**=**1**;**

**}**

**else** **if(**ch**==**Cv**)***//粘贴键*

**{**

**if(**rcdflg**==**1**)** *//如果剪贴板有内容*

**{**

**if(**rcdsign**==**2**)**

turnp**(**rcd**,**rcdlen**);**

**for(**i**=**0**;**i**<**rcdlen**;**i**++)**

**{**

bioget**[**0**]=**rcd**[**i**];**

insert**(**bioget**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**),**Chead**);**

**}**

memset**(**rcd**,**'\0'**,**10**);**

**}**

view**(**Chead**);**

rcdlen**=**0**;**

sig**=**1**;**

**}**

**else** **if(**ch**==**TAB**)** *//TAB键入4空格*

**{**

sig**=**1**;**

bioget**[**0**]=**' '**;**

**for(**i**=**0**;**i**<**4**;**i**++)**

**{**

insert**(**bioget**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**),**Chead**);**

**}**

**}**

**else** **if(**a**>=**32**&&**a**<**127**)** *//输入数字，符号*

**{**

sig**=**1**;**

bioget**[**0**]=**a**;**

insert**(**bioget**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**),**Chead**);**

match**(**bioget**,**Chead**);***//检测是否需要匹配*

**}**

**else** **if(**ch**==**BACK**)***//退格键*

**{**

sig**=**1**;**

**if(**rcdlen**!=**0**)**

**{**

**if(**rcdsign**==**2**)**movecursor**(**clmx**+**rcdlen**,**rowy**);**

**for(**i**=**0**;**i**<**rcdlen**;**i**++)**

deltext**(**Chead**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**));**

rcdlen**=**0**;**

**}**

**else**

deltext**(**Chead**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**));**

**}**

**else** **if(**ch**==**DEL**)***//DEL键*

**{**

sig**=**1**;**

delcursor**();**

cursorX**+=**8**;**

imgflag**=**0**;**

newcursor**();**

deltext**(**Chead**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**));**

**}**

**if((**highlight**(**Chead**)==**0**)&&(**sig**==**1**))**

**{**

view**(**Chead**);**

sig**=**0**;**

**}**

**}**

**}**

**}**

*//用于执行选择编辑按钮的操作*

**int** chosedit**(**Hnode**\*** Chead**,char\*** bioget**,char** **\***rcd**)**

**{**

**int** i**;**

**void** **\***geti**;**

geti**=**malloc**(**imagesize**(**274**,**25**,**354**,**166**));**

clrmous**(**MouseX**,**MouseY**);**

getimage**(**274**,**25**,**354**,**166**,**geti**);**

drawedit**();**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

**while(**1**)***//检测点击循环*

**{**

newmouse**(&**MouseX**,&**MouseY**,&**press**);**

**if((**mouse\_press**(**17**,**17**,**274**,**456**)==**1**)||(**mouse\_press**(**274**,**166**,**354**,**456**)==**1**)||(**mouse\_press**(**354**,**25**,**624**,**456**)==**1**))**

**{**

putimage**(**274**,**25**,**geti**,**0**);**

free**(**geti**);**

**break;**

**}**

**else** **if(**mouse\_press**(**274**,**25**,**354**,**48**)==**1**)***//点击剪切*

**{**

putimage**(**274**,**25**,**geti**,**0**);**

free**(**geti**);**

**if(**rcdsign**==**2**)**

movecursor**(((**cursorX**-**16**)/**8**+**1**)+**rcdlen**,(**cursorY**-**24**)/**16**+**1**);***//如果是从右到左选择的 ,删除前先移动光标*

**for(**i**=**0**;**i**<**rcdlen**;**i**++)**

**{**

deltext**(**Chead**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**));**

**}**

view**(**Chead**);**

rcdflg**=**1**;**

**break;**

**}**

**else** **if(**mouse\_press**(**274**,**49**,**354**,**71**)==**1**)***//点击复制*

**{**

putimage**(**274**,**25**,**geti**,**0**);**

free**(**geti**);**

delcursor**();**

view**(**Chead**);**

rcdflg**=**1**;**

imgflag**=**0**;**

**break;**

**}**

**else** **if(**mouse\_press**(**274**,**72**,**354**,**94**)==**1**)***//点击粘贴*

**{**

putimage**(**274**,**25**,**geti**,**0**);**

free**(**geti**);**

**if(**rcdflg**==**1**)** *//如果剪贴板有内容*

**{**

**if(**rcdsign**==**2**)**

turnp**(**rcd**,**rcdlen**);**

**for(**i**=**0**;**i**<**rcdlen**;**i**++)**

**{**

bioget**[**0**]=**rcd**[**i**];**

insert**(**bioget**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**),**Chead**);**

**}**

memset**(**rcd**,**'\0'**,**10**);**

**}**

view**(**Chead**);**

rcdlen**=**0**;**

**break;**

**}**

**else** **if(**mouse\_press**(**274**,**95**,**354**,**117**)==**1**)***//点击替换*

**{**

replace**(**Chead**)** **;**

**break;**

**}**

**else** **if(**mouse\_press**(**274**,**118**,**354**,**140**)==**1**)***//点击查找*

**{**

find**(**Chead**);**

**break;**

**}**

**else** **if(**mouse\_press**(**616**,**0**,**640**,**24**)==**1**)***//点击关闭按钮结束程序*

**{**

closegraph**();**

exit**(**0**);**

**}**

**}**

**return** 0**;**

**}**

*//用于执行选择文件按钮的操作*

**int** chosfile**(**Hnode**\*** Hhead**,**Hnode**\*\***Chead**)**

**{**

**void** **\***geti**;**

geti**=**malloc**(**imagesize**(**97**,**25**,**177**,**118**));**

clrmous**(**MouseX**,**MouseY**);**

getimage**(**97**,**25**,**177**,**118**,**geti**);**

drawfile**();**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

**while(**1**)***//检测点击循环*

**{**

newmouse**(&**MouseX**,&**MouseY**,&**press**);**

**if((**mouse\_press**(**17**,**17**,**97**,**456**)==**1**)||**

**(**mouse\_press**(**97**,**118**,**177**,**456**)==**1**)||**

**(**mouse\_press**(**177**,**25**,**624**,**456**)==**1**))**

**{***//点击一般位置收起菜单栏*

putimage**(**97**,**25**,**geti**,**0**);**

free**(**geti**);**

**return** 0**;**

**}**

**else** **if(**mouse\_press**(**97**,**25**,**177**,**48**)==**1**)***//点击新建*

**{**

putimage**(**97**,**25**,**geti**,**0**);**

free**(**geti**);**

newbuild**(**Hhead**,&**Chead**);**

**return** 0**;**

**}**

**else** **if(**mouse\_press**(**97**,**49**,**177**,**71**)==**1**)***//点击打开*

**{**

putimage**(**97**,**25**,**geti**,**0**);**

free**(**geti**);**

opens**(**Hhead**,**Chead**)** **;**

**return** 0**;**

**}**

**else** **if(**mouse\_press**(**97**,**72**,**177**,**94**)==**1**)***//点击保存*

**{**

putimage**(**97**,**25**,**geti**,**0**);**

free**(**geti**);**

save**(**Hhead**);**

**return** 0**;**

**}**

**else** **if(**mouse\_press**(**97**,**95**,**177**,**117**)==**1**)***//点击另存为*

**{**

putimage**(**97**,**25**,**geti**,**0**);**

free**(**geti**);**

saveas**(**Hhead**);**

**return** 0**;**

**}**

**}**

**}**

*// 用于执行选择帮助按钮的操作*

**int** choshelp**()**

**{**

**void** **\***geti**;**

geti**=**malloc**(**imagesize**(**451**,**25**,**531**,**71**));**

clrmous**(**MouseX**,**MouseY**);**

getimage**(**451**,**25**,**531**,**71**,**geti**);**

drawhelp**();**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

**while(**1**)***//检测点击循环*

**{**

newmouse**(&**MouseX**,&**MouseY**,&**press**);**

**if((**mouse\_press**(**17**,**17**,**451**,**456**)==**1**)||(**mouse\_press**(**451**,**71**,**531**,**456**)==**1**)||(**mouse\_press**(**531**,**25**,**624**,**456**)==**1**))**

**{**

putimage**(**451**,**25**,**geti**,**0**);**

free**(**geti**);**

**break;**

**}**

**else** **if(**mouse\_press**(**451**,**25**,**531**,**48**)==**1**)***//点击关于*

**{**

putimage**(**451**,**25**,**geti**,**0**);**

free**(**geti**);**

clrmous**(**MouseX**,**MouseY**);**

about**();**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

**while(**1**)**

**{**

newmouse**(&**MouseX**,&**MouseY**,&**press**);**

**if(**mouse\_press**(**5**,**5**,**61**,**45**)==**1**)**

**break;**

**}**

**break;**

**}**

**else** **if(**mouse\_press**(**451**,**49**,**531**,**71**)==**1**)***//点击帮助*

**{**

putimage**(**451**,**25**,**geti**,**0**);**

free**(**geti**);**

clrmous**(**MouseX**,**MouseY**);**

help**();**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

**while(**1**)**

**{**

newmouse**(&**MouseX**,&**MouseY**,&**press**);**

**if(**mouse\_press**(**5**,**5**,**61**,**45**)==**1**)**

**break;**

**}**

**break;**

**}**

**}**

**return** 0**;**

**}**

3.func.c

#include<common.h>

#include"func.h"

**extern** **int** imgflag**;**

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*【编辑功能模块】*

*编写人：胡耀文*

*find用于查找并标记显示*

*replace用于替换字符串*

*ctrlr，ctrll，turnp用于处理选定字符的工作*

*colorview用于标记字符*

*highlight，changecolor，changeflag一起处理关键字代码高亮*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*//作用：查找并标记特定的字符串*

**void** find**(**Hnode**\*** Chead**)**

**{**

**int** i**,**j**,**k**,**slen**,**c**=**1**,**r**=**1**;**

**char** row**[**80**];**

**char** **\***str**,\***s0**,\***s1**;**

Hnode **\***q**;**

node **\***p1**;**

q**=**Chead**;**

str**=(char\*)**malloc**(**10**);**

setfillstyle**(**1**,** BLUE**);**

bar**(**17**,**25**,**632**,**458**);***//字符区域呈蓝色*

puthz**(**150**,**200**,"请输入要查找的字符串：",** 16**,** 16**,**WHITE**);**

slen**=**gettxtname**(**str**,**326**,**206**)-**1**;***//记录字符串长度*

view**(**Chead**);**

**for(**i**=**1**;(**q**!=**NULL**)&&(**i**<=**27**);**i**++)***//循环检测每行*

**{**

c**=**1**;**

p1**=**q**->**next**;**

**for(**j**=**0**;**p1**!=**NULL**;**j**++)** *//循环将链表数据读入数组*

**{**

**if(**p1**->**sign**==**1**)***//如果遇到汉字存两位使列相配*

**{**

row**[**j**]=**p1**->**txt**[**0**];**

j**++;**

**}**

row**[**j**]=**p1**->**txt**[**0**];**

p1**=**p1**->**next**;**

**}**

row**[**j**]=**'\0'**;**

*//查找数组里是否有匹配的字符串*

s0**=**row**;**

*//查找输入的字符串，s0不断改变位置指向特定字符后一位，s1指向特定字符串，利用colorview标记显示查找的字符串；*

**do**

**{**

s1**=**strstr**(**s0**,**str**);**

c**=**c**+(**s1**-**s0**);**

**if(**s1**!=**NULL**)**

**for(**k**=**0**;**k**<**slen**;**k**++)**

**{**

colorview**(**r**,**c**,**Chead**);**

c**++;**

**}**

s0**=**s1**+**slen**;**

**}while(**s1**!=**NULL**);**

r**++;**

q**=**q**->**nextl**;**

**}**

free**(**str**);**

**}**

*//作用：查找并替换特定字符串*

**void** replace**(**Hnode**\*** Chead**)**

**{**

**int** i**,**j**,**k**,**alen**,**blen**,**c**=**1**,**r**=**1**;**

**char** row**[**80**];**

**char** **\***astr**,\***s0**,\***s1**,\***s2**,\***bstr**,**sstr**[**2**];***//astr为原来的字符，bstr为现在的字符*

Hnode **\***q**;**

node **\***p1**;**

q**=**Chead**;**

astr**=(char\*)**malloc**(**10**);**

bstr**=(char\*)**malloc**(**76**);**

setfillstyle**(**1**,** BLUE**);**

bar**(**17**,**25**,**632**,**458**);***//字符区域呈蓝色*

puthz**(**150**,**200**,"输入要替换的字符串：",** 16**,** 16**,**WHITE**);**

alen**=**gettxtname**(**astr**,**326**,**206**)-**1**;***//记录字符串长度*

bar**(**17**,**25**,**632**,**458**);**

puthz**(**150**,**200**,"输入新的字符串：",** 16**,** 16**,**WHITE**);**

blen**=**gettxtname**(**bstr**,**278**,**206**)-**1**;***//记录字符串长度*

view**(**Chead**);**

**for(**i**=**1**;(**q**!=**NULL**)&&(**i**<=**27**);**i**++)***//循环检测每行*

**{**

c**=**1**;**

p1**=**q**->**next**;**

**for(**j**=**0**;**p1**!=**NULL**;**j**++)** *//循环将链表数据读入数组*

**{**

**if(**p1**->**sign**==**1**)***//如果遇到汉字存两位使列相配*

**{**

row**[**j**]=**p1**->**txt**[**0**];**

j**++;**

**}**

row**[**j**]=**p1**->**txt**[**0**];**

p1**=**p1**->**next**;**

**}**

row**[**j**]=**'\0'**;**

*//查找数组里是否有匹配的字符串*

s0**=**row**;**

**do**

**{**

s1**=**strstr**(**s0**,**astr**);**

**if(**s1**!=**NULL**)**

**{**

c**=**s1**-**s0**+**alen**+**1**;**

movecursor**(**c**+**1**,**r**);**

**for(**k**=**0**;**k**<**alen**;**k**++)***//这个for用于执行删除原来的字符*

**{**

deltext**(**Chead**,**r**,**c**);**

c**--;**

**}**

s2**=**s1**+**alen**;**

strcpy**(**s1**,**s2**);***//执行字符串上的修改*

**for(**k**=**0**;**k**<**blen**;**k**++)***//这个用于执行插入行的字符串*

**{**

sstr**[**0**]=**bstr**[**k**];**

sstr**[**1**]=**'\0'**;**

insert**(**sstr**,**r**,**c**,**Chead**);**

c**++;**

**}**

strcat**(**bstr**,**s1**);**

strcpy**(**s1**,**bstr**);**

bstr**[**blen**]=**'\0'**;**

**}**

**}while(**s1**!=**NULL**);**

r**++;**

q**=**q**->**nextl**;**

**}**

free**(**astr**);**

free**(**bstr**);**

view**(**Chead**);**

**}**

*//作用：光标向右选中一个字符*

**void** ctrlr**(int** rcdlen**,**Hnode**\*** Chead**,char** **\***rcd**,int** m**,int** n**)**

**{**

**int** i**;**

Hnode**\*** q1**;***//此指针为指向一行数据的指针*

node**\***p1**;***//此指针为指向一个数据节点的指针*

q1**=**Chead**;***//q1指向当前显示区域的第一行*

**for(**i**=**1**;**i**<**m**;**i**++)**

q1**=**q1**->**nextl**;***//循环使得q1移动到第m行*

p1**=**q1**->**next**;***//p1指向第m行的第一列*

**for(**i**=**1**;**i**<**n**;**i**++)***//循环到第n列*

**{**

p1**=**p1**->**next**;**

**if(**p1**->**sign**==**1**)** i**++;***//如果遇到汉字，检测的列后移一位 (因为汉字占两列*

**}**

rcd**[**rcdlen**]=**p1**->**txt**[**0**];**

rcd**[**rcdlen**+**1**]=**'\0'**;**

**}**

*//作用：光标向左选中一个字符*

**void** ctrll**(int** rcdlen**,**Hnode**\*** Chead**,char** **\***rcd**,int** m**,int** n**)**

**{**

**int** i**;**

Hnode**\*** q1**;***//此指针为指向一行数据的指针*

node**\***p1**;***//此指针为指向一个数据节点的指针*

q1**=**Chead**;***//q1指向当前显示区域的第一行*

**for(**i**=**1**;**i**<**m**;**i**++)**

q1**=**q1**->**nextl**;***//循环使得q1移动到第m行*

p1**=**q1**->**next**;***//p1指向第m行的第一列*

**for(**i**=**1**;**i**<**n**;**i**++)***//循环到第n列*

**{**

p1**=**p1**->**next**;**

**if(**p1**->**sign**==**1**)** i**++;***//如果遇到汉字，检测的列后移一位 (因为汉字占两列*

**}**

rcd**[**rcdlen**]=**p1**->**txt**[**0**];**

rcd**[**rcdlen**+**1**]=**'\0'**;**

**}**

*//作用：因左选后字符串是倒序的，故调用此函数倒置使其正序*

**int** turnp**(char** **\***rcd**,int** rcdlen**)**

**{**

**char** temp**;**

**int** i**,**j**,**k**;**

i**=**0**;**

j**=** rcdlen**-**1**;**

**if(**rcdlen**==**1**)***//如果是两个字符*

**{**

temp**=**rcd**[**i**];**

rcd**[**i**]=**rcd**[**j**];**

rcd**[**j**]=**temp**;**

**}**

**else**

**{**

**for(**k**=**0**;**k**<(**rcdlen**/**2**);**k**++)**

**{**

temp**=**rcd**[**i**];**

rcd**[**i**]=**rcd**[**j**];**

rcd**[**j**]=**temp**;**

i**++;**

j**--;**

**}**

**}**

**}**

*//用于执行循环检测高亮*

**int** highlight**(**Hnode **\***Chead**)**

**{** *//定义保留字*

**static** **char\*** c**[**32**]={"long","short","float","double","char",**

**"unsigned","signed","const","void","volatile","enum",**

**"struct","union","if","else","goto","switch","case",**

**"do","while","for","continue","break","return","default",**

**"typedef","auto","register","extern","static","sizeof","int"};**

**static** **int** clen**[**32**]={**4**,**5**,**5**,**6**,**4**,**8**,**6**,**5**,**4**,**8**,**4**,**6**,**5**,**2**,**4**,**4**,**6**,**4**,**2**,**5**,**3**,**8**,**5**,**6**,**7**,**7**,**4**,**8**,**6**,**6**,**6**,**3**};**

**int** i**,**sig**;**

**for(**i**=**0**;**i**<**32**;**i**++)**

sig**=**changecolor**(**c**[**i**],**clen**[**i**],**Chead**);**

**return** sig**;**

**}**

*//用于查找并改变字符串的颜色*

**int** changecolor**(char\*** str**,int** slen**,**Hnode **\***Chead**)**

**{**

**int** i**,**j**,**k**,**c**=**1**,**r**=**1**,**sig**=**1**;**

**char** row**[**80**];**

**char** **\***s0**,\***s1**,**ch1**,**ch2**;***//ch1,ch2用于记录保留字前面和后面的字符判断保留字是否单独出现*

Hnode **\***q**;**

node **\***p1**;**

q**=**Chead**;**

**for(**i**=**1**;(**q**!=**NULL**)&&(**i**<=**27**);**i**++)***//循环检测每行*

**{**

c**=**1**;**

p1**=**q**->**next**;**

**for(**j**=**0**;**p1**!=**NULL**;**j**++)** *//循环将链表数据读入数组*

**{**

**if(**p1**->**sign**==**1**)***//如果遇到汉字存两位使列相配*

**{**

row**[**j**]=**p1**->**txt**[**0**];**

j**++;**

**}**

row**[**j**]=**p1**->**txt**[**0**];**

p1**=**p1**->**next**;**

**}**

row**[**j**]=**'\0'**;**

*//查找数组里是否有匹配的字符串*

s0**=**row**;**

*//查找固定的字符串，s0不断改变位置指向特定字符后一位，s1指向特定字符串*

**do**

**{**

s1**=**strstr**(**s0**,**str**);**

c**=**c**+(**s1**-**s0**);**

**if(**s1**!=**NULL**)** *//如果查找到*

**{**

ch1**=\*(**s1**-**1**);**

ch2**=\*(**s1**+**slen**);**

**if(!((**ch1**>=**'0'**&&**ch1**<=**'9'**)||(**ch1**>=**'A'**&&**ch2**<=**'Z'**)||(**ch1**>=**'a'**&&**ch1**<=**'z'**)||(**ch1**==**'\_'**)||**

**(**ch2**>=**'0'**&&**ch2**<=**'9'**)||(**ch2**>=**'A'**&&**ch2**<=**'Z'**)||(**ch2**>=**'a'**&&**ch2**<=**'z'**)||(**ch2**==**'\_'**)))**

**{**

**for(**k**=**0**;**k**<**slen**;**k**++)**

**{**

changeflag**(**r**,**c**,**Chead**,**YELLOW**);**

c**++;**

**}**

sig**=**0**;**

**}**

**else**

**{**

**for(**k**=**0**;**k**<**slen**;**k**++)**

**{**

changeflag**(**r**,**c**,**Chead**,**WHITE**);**

c**++;**

**}**

sig**=**0**;**

**}**

**}**

s0**=**s1**+**slen**;**

**}while(**s1**!=**NULL**);**

r**++;**

q**=**q**->**nextl**;**

**}**

**return** sig**;**

**}**

*//作用：改变m行n列的颜色*

**int** changeflag**(int** m**,int** n**,**Hnode **\***Chead**,int** color**)**

**{**

**int** i**,**j**;**

Hnode**\*** q1**;***//此指针为指向一行数据的指针*

node**\***p1**;***//此指针为指向一个数据节点的指针*

q1**=**Chead**;***//q1指向当前显示区域的第一行*

**for(**i**=**1**;**i**<**m**;**i**++)**

q1**=**q1**->**nextl**;***//循环使得q1移动到第m行*

p1**=**q1**->**next**;***//p1指向第m行的第一列*

**for(**i**=**1**;**i**<**n**;**i**++)***//循环到第n列*

**{**

p1**=**p1**->**next**;**

**if(**p1**->**sign**==**1**)** i**++;** *//如果遇到汉字，检测的列后移一位 (因为汉字占两列*

**}**

p1**->**flag**=**color**;**

**}**

4.insert.c

#include<common.h>

#include"INSERT.h"

**extern** **int** cursorX**;**

**extern** **int** cursorY**;**

**extern** **void** **\***bitmap**;**

**extern** **int** imgflag**;**

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*【文本基础编辑模块】*

*编写人：胡耀文*

*insert用于执行字符插入操作；*

*deltext用于执行删除操作 ；*

*enter用于执行换行操作；*

*movepic，getlen为辅助函数*

*match用于执行括号引号匹配操作。*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*//insert函数执行在一个位置插入节点，并把bioget存的字符保存进节点*

**int** insert**(char\*** bioget**,int** m**,int** n**,**Hnode**\*** Chead**)**

**{**

**int** i**;**

Hnode**\*** q1**,\***q2**;***//此指针为指向一行数据的指针*

node**\***p1**,\***p2**,\***p3**,\***p4**;***//此指针为指向一个数据节点的指针*

q1**=**Chead**;***//q1指向当前显示区域的第一行*

**for(**i**=**1**;**i**<**m**;**i**++)**

q1**=**q1**->**nextl**;***//循环使得q1移动到第m行*

p1**=**q1**->**next**;***//p1指向第m行的第一列*

**for(**i**=**1**;**i**<**n**-**1**;**i**++)***//循环到第n-1列*

**{**

p1**=**p1**->**next**;**

**if(**p1**->**sign**==**1**)** i**++;** *//如果遇到汉字，检测的列后移一位 (因为汉字占两列*

**}**

*//以上为初始化过程*

**if(**p1**->**next**!=**NULL**)** *//处理行间插入*

**{**

**if(**cursorX**==**17**)***//处理在第一列插入*

**{**

**if(**getlinelen**(**q1**)==**76**)***//处理本行已经满的情况*

**{**

p2**=(**node**\*)**malloc**(sizeof(**node**));**

p2**->**txt**[**0**]=**bioget**[**0**];**p2**->**txt**[**1**]=**'\0'**;**p2**->**sign**=**0**;**p2**->**flag**=**WHITE**;**

p2**->**next**=**q1**->**next**;**

q1**->**next**=**p2**;**

**for(**p3**=**p2**;**p3**->**next**->**next**->**txt**[**0**]!=**13**;**p3**=**p3**->**next**){;}**

p4**=**p3**->**next**;**

p3**->**next**=**p4**->**next**;**

**if(**q1**->**nextl**==**NULL**)***//如果下一个行节点是空的，新建一行*

**{**

q2**=(**Hnode**\*)**malloc**(sizeof(**Hnode**));**

q2**->**next**=**p4**;**

q2**->**nextl**=**NULL**;**

q1**->**nextl**=**q2**;**

p4**->**next**=**NULL**;**

**}**

**else***//下一行节点非空，写入下一行*

**{**

q2**=**q1**->**nextl**;**

p4**->**next**=**q2**->**next**;**

q2**->**next**=**p4**;**

**}**

view**(**Chead**);**

cursorX**+=**8**;**

imgflag**=**0**;**

newcursor**();**

**}**

**else***//处理本行未满的情况*

**{**

p2**=(**node**\*)**malloc**(sizeof(**node**));***//新建一个节点*

p2**->**txt**[**0**]=**bioget**[**0**];**p2**->**txt**[**1**]=**'\0'**;**p2**->**sign**=**0**;** p2**->**flag**=**WHITE**;**

q1**->**next**=**p2**;***//将该节点作为该行的头结点*

p2**->**next**=**p1**;***//将p1接到p2后面*

delcursor**();**

clrmous**(**MouseX**,**MouseY**);**

movepic**(**cursorX**,**cursorY**,**getlen**(**p1**),**cursorX**+**8**);**

setfillstyle**(**1**,** BLUE**);**

bar**(**cursorX**,**cursorY**,**cursorX**+**7**,**cursorY**+**15**);**

settextstyle**(**0**,**0**,**1**);**

setcolor**(**WHITE**);**

outtextxy**(**cursorX**,**cursorY**+**6**,**bioget**);**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

cursorX**+=**8**;**

imgflag**=**0**;**

newcursor**();**

**}**

**}**

**else***//处理不在第一行的插入*

**{**

**if(**getlinelen**(**q1**)==**76**)***//处理本行已经满的情况*

**{**

p2**=(**node**\*)**malloc**(sizeof(**node**));**

p2**->**txt**[**0**]=**bioget**[**0**];**p2**->**txt**[**1**]=**'\0'**;**p2**->**sign**=**0**;**p2**->**flag**=**WHITE**;**

p2**->**next**=**p1**->**next**;**

p1**->**next**=**p2**;**

**for(**p3**=**p2**;**p3**->**next**->**next**->**txt**[**0**]!=**13**;**p3**=**p3**->**next**){;}**

p4**=**p3**->**next**;**

p3**->**next**=**p4**->**next**;**

**if(**q1**->**nextl**==**NULL**)***//如果下一个行节点是空的，新建一行*

**{**

q2**=(**Hnode**\*)**malloc**(sizeof(**Hnode**));**

q2**->**next**=**p4**;**

q2**->**nextl**=**NULL**;**

q1**->**nextl**=**q2**;**

p4**->**next**=**NULL**;**

**}**

**else**

**{**

q2**=**q1**->**nextl**;**

p4**->**next**=**q2**->**next**;**

q2**->**next**=**p4**;**

**}**

view**(**Chead**);**

cursorX**+=**8**;**

imgflag**=**0**;**

newcursor**();**

**}**

**else***//处理本行未满的情况*

**{**

*//链表操作：*

p3**=**p1**->**next**;**

p2**=(**node**\*)**malloc**(sizeof(**node**));**

p2**->**txt**[**0**]=**bioget**[**0**];**p2**->**txt**[**1**]=**'\0'**;**p2**->**sign**=**0**;**p2**->**flag**=**WHITE**;**

p1**->**next**=**p2**;**

p2**->**next**=**p3**;**

*//图形操作*

delcursor**();**

clrmous**(**MouseX**,**MouseY**);**

movepic**(**cursorX**,**cursorY**,**getlen**(**p3**),**cursorX**+**8**);**

setfillstyle**(**1**,** BLUE**);**

bar**(**cursorX**,**cursorY**,**cursorX**+**7**,**cursorY**+**15**);**

setcolor**(**WHITE**);**

settextstyle**(**0**,**0**,**1**);**

outtextxy**(**cursorX**,**cursorY**+**6**,**bioget**);**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

cursorX**+=**8**;**

imgflag**=**0**;**

newcursor**();**

**}**

**}**

**}**

**else***//处理行尾输入*

**{**

**if(**cursorX**==**17**)***//处理在第一个位置输入*

**{**

**if(**p1**!=**NULL**)***// 处理在第一个位置有节点的情况，此时不需要新建*

**{**

p1**->**txt**[**0**]=**bioget**[**0**];**

p1**->**txt**[**1**]=**'\0'**;**

p1**->**sign**=**0**;**

p1**->**next**=**NULL**;**

p1**->**flag**=**WHITE**;**

delcursor**();**

clrmous**(**MouseX**,**MouseY**);**

setcolor**(**WHITE**);**

settextstyle**(**0**,**0**,**1**);**

outtextxy**(**cursorX**,**cursorY**+**6**,**bioget**);**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

cursorX**+=**8**;**

imgflag**=**0**;**

newcursor**();**

**}**

**else***//处理在第一个位置没有节点的情况*

**{**

p1**=(**node**\*)**malloc**(sizeof(**node**));**

p1**->**txt**[**0**]=**bioget**[**0**];**

p1**->**txt**[**1**]=**'\0'**;**

p1**->**sign**=**0**;**

p1**->**next**=**NULL**;**

p1**->**flag**=**WHITE**;**

delcursor**();**

clrmous**(**MouseX**,**MouseY**);**

settextstyle**(**0**,**0**,**1**);**

setcolor**(**WHITE**);**

outtextxy**(**cursorX**,**cursorY**+**6**,**bioget**);**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

cursorX**+=**8**;**

imgflag**=**0**;**

newcursor**();**

**}**

**}**

**else** **if(**n**==**76**)***//处理在行末输入*

**{**

p2**=(**node**\*)**malloc**(sizeof(**node**));***//新建一个节点 存数据*

p2**->**txt**[**0**]=**bioget**[**0**];**

p2**->**txt**[**1**]=**'\0'**;**

p2**->**sign**=**0**;**

p2**->**flag**=**WHITE**;**

p3**=(**node**\*)**malloc**(sizeof(**node**));***//新建一个节点存回车键*

p3**->**txt**[**0**]=**13**;**

p3**->**txt**[**1**]=**'\0'**;**

p3**->**sign**=**0**;**

p1**->**next**=**p2**;**

p2**->**next**=**p3**;**

p3**->**next**=**NULL**;**

**if(**q1**->**nextl**==**NULL**)***//如果下一个行节点是空的，新建一行*

**{**

q2**=(**Hnode**\*)**malloc**(sizeof(**Hnode**));**

q2**->**next**=**NULL**;**

q2**->**nextl**=**NULL**;**

q1**->**nextl**=**q2**;**

**}**

delcursor**();**

clrmous**(**MouseX**,**MouseY**);**

settextstyle**(**0**,**0**,**1**);**

setcolor**(**WHITE**);**

outtextxy**(**cursorX**,**cursorY**+**6**,**bioget**);**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

movecursor**(**1**,(**cursorY**-**24**)/**16**+**2**);**

imgflag**=**0**;**

newcursor**();**

**}**

**else** *//处理在中间列的行尾*

**{**

p2**=(**node**\*)**malloc**(sizeof(**node**));**

p2**->**txt**[**0**]=**bioget**[**0**];**p2**->**flag**=**WHITE**;**

p2**->**txt**[**1**]=**'\0'**;**

p2**->**sign**=**0**;**

p2**->**next**=**NULL**;**

p1**->**next**=**p2**;**

delcursor**();**

clrmous**(**MouseX**,**MouseY**);**

settextstyle**(**0**,**0**,**1**);**

setcolor**(**WHITE**);**

outtextxy**(**cursorX**,**cursorY**+**6**,**bioget**);**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

cursorX**+=**8**;**

imgflag**=**0**;**

newcursor**();**

**}**

**}**

**return** 0**;**

**}**

*//用于执行删除r行c列字符或者汉字的操作*

**int** deltext**(**Hnode**\*** Chead**,int** r**,int** c**)**

**{**

**int** i**;**

Hnode**\*** q1**,\***q2**;**

node**\*** p1**,\***p2**,\***p3**,\***p4**;**

q1**=**Chead**;**q2**=**Chead**;**

**for(**i**=**1**;**i**<**r**;**i**++)**

q1**=**q1**->**nextl**;**

p1**=**q1**->**next**;**

**for(**i**=**1**;**i**<**c**-**2**;**i**++)***//循环到该列前2列*

**{**

p1**=**p1**->**next**;**

**if(**p1**->**next**->**sign**==**1**)** i**++;***//如果遇到汉字，检测列后移一位*

**}**

p2**=**p1**->**next**;**

**if(**c**==**1**)***//如果在行头*

**{**

**if(**r**==**1**)return** 0**;***// 如果在第一行不操作*

**else***//不在第一行，把此行和上一行连起来*

**{**

**for(**i**=**1**;**i**<**r**-**1**;**i**++)**

q2**=**q2**->**nextl**;**

p3**=**q2**->**next**;**

**for(**i**=**1**;**i**<**getlinelen**(**q2**);**i**++)**

**{**

p3**=**p3**->**next**;**

**}**

p4**=**p3**->**next**;**

free**(**p4**);**

p3**->**next**=**p1**;**

q2**->**nextl**=**q1**->**nextl**;**

free**(**q1**);**

view**(**Chead**);**

imgflag**=**0**;**

newcursor**();**

movecursor**(**i**+**1**,**r**-**1**);***//移动光标到上一行的位置*

imgflag**=**0**;**

newcursor**();**

**}**

**}**

**else***//不在行头*

**{**

**if((**p2**->**next**==**NULL**)||(**p2**==**NULL**))***//如果该处下一个节点是空的,删除此节点*

**{**

**if(**c**==**2**)** *//如果光标在第二个位置，删除第一个字符*

**{**

free**(**p1**);**

q1**->**next**=**NULL**;**

delcursor**();**

clrmous**(**MouseX**,**MouseY**);**

setfillstyle**(**1**,** BLUE**);**

bar**(**cursorX**-**8**,** cursorY**,** cursorX**-**1**,**cursorY**+**15**);**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

cursorX**-=**8**;**

imgflag**=**0**;**

newcursor**();**

**}**

**else***//不在第二个位置*

**{**

**if(**p2**->**sign**==**0**)***//如果是字符*

**{**

p1**->**next**=**NULL**;**

free**(**p2**);**

delcursor**();**

clrmous**(**MouseX**,**MouseY**);**

setfillstyle**(**1**,** BLUE**);**

bar**(**cursorX**-**8**,** cursorY**,** cursorX**-**1**,**cursorY**+**15**);**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

cursorX**-=**8**;**

imgflag**=**0**;**

newcursor**();**

**}**

**else** **if(**p2**->**sign**==**1**)***//如果是汉字*

**{**

p1**->**next**=**NULL**;**

free**(**p2**);**

delcursor**();**

clrmous**(**MouseX**,**MouseY**);**

setfillstyle**(**1**,** BLUE**);**

bar**(**cursorX**-**16**,** cursorY**,** cursorX**-**1**,**cursorY**+**15**);**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

cursorX**-=**16**;**

imgflag**=**0**;**

newcursor**();**

**}**

**}**

**}**

**else***//如果下节点非空，删掉此处节点，将后面的图像前移*

**{**

**if(**c**==**2**)** *//如果光标在第二个位置，删除第一个字符*

**{**

free**(**p1**);**

q1**->**next**=**p2**;**

view**(**Chead**);**

cursorX**-=**8**;**

imgflag**=**0**;**

newcursor**();**

**}**

**else***//不在第二个位置*

**{**

**if(**p2**->**sign**==**0**)***//如果是字符*

**{**

p2**=**p1**->**next**;**

p1**->**next**=**p2**->**next**;**

free**(**p2**);**

view**(**Chead**);**

cursorX**-=**8**;**

imgflag**=**0**;**

newcursor**();**

**}**

**else** **if(**p2**->**sign**==**1**)***//如果是汉字*

**{**

p1**->**next**=**p2**->**next**;**

free**(**p2**);**

delcursor**();**

view**(**Chead**);**

cursorX**-=**16**;**

imgflag**=**0**;**

**}**

**}**

**}**

**}**

**return** 0**;**

**}**

*//enter函数用于执行按下enter键的换行操作*

**void** enter**(int** m**,int** n**,**Hnode**\*** Chead**)**

**{**

**int** i**;**

Hnode**\*** q1**,\***q2**;***//此指针为指向一行数据的指针*

node**\***p1**,\***p2**,\***p3**;***//此指针为指向一个数据节点的指针*

q1**=**Chead**;***//q1指向当前显示区域的第一行*

**for(**i**=**1**;**i**<**m**;**i**++)**

q1**=**q1**->**nextl**;***//循环使得q1移动到第m行*

p1**=**q1**->**next**;***//p1指向第m行的第一列*

**for(**i**=**1**;**i**<**n**-**1**;**i**++)***//循环到第n-1列*

**{**

p1**=**p1**->**next**;**

**if(**p1**->**sign**==**1**)** i**++;***//如果遇到汉字，检测的列后移一位 (因为汉字占两列*

**}**

p2**=(**node**\*)**malloc**(sizeof(**node**));**

p2**->**txt**[**0**]=**13**;**

p2**->**txt**[**1**]=**'\0'**;**

p2**->**sign**=**0**;**

q2**=(**Hnode**\*)**malloc**(sizeof(**Hnode**));**

**if(**n**==**1**)***//处理在一行开头的情况*

**{**

**if(**q1**->**nextl**!=**NULL**)**

**{**

q2**->**nextl**=**q1**->**nextl**;**

q1**->**nextl**=**q2**;**

q2**->**next**=**p1**;**

q1**->**next**=**p2**;**

p2**->**next**=**NULL**;**

**}**

**else**

**{**

q2**->**next**=**p1**;**

q1**->**next**=**p2**;**

q1**->**nextl**=**q2**;**

q2**->**next**=**NULL**;**

**}**

**}**

**else***//处理不在一行开头的情况*

**{**

**if(**q1**->**nextl**!=**NULL**)***//处理有下一行的情况*

**{**

q2**->**nextl**=**q1**->**nextl**;**

q1**->**nextl**=**q2**;**

**if(**p1**->**next**!=**NULL**)**

**{**

q2**->**next**=**p1**->**next**;**

p1**->**next**=**p2**;**

p2**->**next**=**NULL**;**

**}**

**else**

**{**

p1**->**next**=**p2**;**

q2**->**next**=**NULL**;**

**}**

**}**

**else***//处理没有下一行的情况*

**{**

**if(**p1**->**next**!=**NULL**)**

**{**

q2**->**next**=**p1**->**next**;**

q2**->**nextl**=**NULL**;**

q1**->**nextl**=**q2**;**

p1**->**next**=**p2**;**

p2**->**next**=**NULL**;**

**}**

**else{**

q1**->**nextl**=**q2**;**

q2**->**nextl**=**NULL**;**

q2**->**next**=**NULL**;**

**}**

**}**

**}**

view**(**Chead**);**

cursorX**=**17**;**

cursorY**+=**16**;**

imgflag**=**0**;**

**}**

*//作用：将x1,y1处长为l的列图像保存，再放到x2,y2处*

**int** movepic**(int** x1**,int** y1**,int** l**,int** x2**)**

**{**

**void** **\***pic**;**

pic**=**malloc**(**imagesize**(**x1**,**y1**,**x1**+**7**+**l**\***8**,**y1**+**15**));**

getimage**(**x1**,**y1**,**x1**+**7**+(**l**-**1**)\***8**,**y1**+**15**,**pic**);**

setcolor**(**WHITE**);**

putimage**(**x2**,**y1**,**pic**,**0**);**

free**(**pic**);**

**return** 0**;**

**}**

*//检查传入的指针后面有几个节点并返回这个值*

**int** getlen**(**node**\*** p**)**

**{**

**int** i**=**1**;**

**while(**p**->**next**!=**NULL**)**

**{**

i**++;**

p**=**p**->**next**;**

**}**

**return** i**;**

**}**

*//用于检测括号引号并进行相应的匹配*

**void** match**(char\*** bioget**,**Hnode**\*** Chead**)**

**{**

**char** match**[**2**]={**'\0'**,**'\0'**};**

**if(**bioget**[**0**]-**'('**==**0**)**

**{**

match**[**0**]=**41**;**

insert**(**match**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**),**Chead**);**

**}**

**else** **if(**bioget**[**0**]-**'{'**==**0**)**

**{**

match**[**0**]=**125**;**

insert**(**match**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**),**Chead**);**

**}**

**else** **if(**bioget**[**0**]==**34**)**

**{**

insert**(**bioget**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**),**Chead**);**

**}**

**else** **if(**bioget**[**0**]-**'['**==**0**)**

**{**

match**[**0**]=**93**;**

insert**(**match**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**),**Chead**);**

**}**

**else** **if(**bioget**[**0**]-**'<'**==**0**)**

**{**

match**[**0**]=**62**;**

insert**(**match**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**),**Chead**);**

**}**

**else** **if(**bioget**[**0**]==**39**)**

**{**

insert**(**bioget**,((**cursorY**-**24**)/**16**+**1**),((**cursorX**-**16**)/**8**+**1**),**Chead**);**

**}**

**}**

5.cursor.c

#include<common.h>

#include<cursor.h>

**int** cursorX**=**17**;**

**int** cursorY**=**25**;**

**char** bitmap**[**30**];***//用于储存光标的图像*

**int** imgflag**=**0**;**

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*【光标模块】*

*编写人：胡耀文*

*drawcursor用于绘制光标*

*getcsim用于保存光标下的图像防止留痕*

*delcursor用于删除当前光标*

*movecursor用于将光标移动到第y行x列的位置*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**void** drawcursor**()** *//光标绘制函数*

**{**

**static** **int** i**=**0**,**j**=**50**;***//循环检测,画出白线和蓝线，白线显示光标，蓝线删除光标 。*

setlinestyle**(**0**,**0**,**1**);**

**if(**i**<**j**&&**i**>=**0**)**

**{**

setcolor**(**WHITE**);**

line**(**cursorX**,**cursorY**,**cursorX**,**cursorY**+**16**);**

**}**

**else** **if(**j**<=**i**&&**i**<**2**\***j**)**

**{**

**if(**i**==**50**)**putimage**(**cursorX**,**cursorY**,**bitmap**,**0**);**

**}**

**else**

**{**

i**=**0**;**

**}**

i**++;**

**}**

*//删除当前光标*

**void** delcursor**()**

**{**

putimage**(**cursorX**,**cursorY**,**bitmap**,**0**);**

**}**

*//用于获取光标处的图像*

**void** getcsim**()**

**{**

getimage**(**cursorX**,**cursorY**,**cursorX**,**cursorY**+**16**,**bitmap**);**

**}**

*//更新光标*

**void** newcursor**()**

**{**

**if(**imgflag**==**0**)***//如果还没有储存光标的图像，就储存，防止光标掩盖屏幕内容*

**{**

clrmous**(**MouseX**,**MouseY**);**

getcsim**();**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

imgflag**=**1**;**

**}**

drawcursor**();***//绘制光标*

**}**

*//移动光标到x列,y行*

**void** movecursor**(int** x**,int** y**)**

**{**

delcursor**();**

cursorX**=**17**+**8**\*(**x**-**1**);**

cursorY**=**25**+**16**\*(**y**-**1**);**

imgflag**=**0**;**

**}**

6.draw.c

#include<common.h>

#include<draw.h>

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*【画图模块】*

*编写人：王思为*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**void** drawmenu**()**

**{**

cleardevice**();**

setfillstyle**(**1**,** LIGHTGRAY**);**

bar**(**1**,** 1**,** 640**,**24**);**

bar**(**1**,** 1**,** 16**,** 480**);**

bar**(**1**,** 456**,** 640**,** 480**);**

bar**(**97**,**1**,**177**,**24**);**

bar**(**274**,**1**,**354**,**24**);**

bar**(**451**,**1**,**533**,**24**);**

bar**(**624**,** 25**,** 640**,** 455**);**

setcolor**(**DARKGRAY**);**

line**(**97**,**1**,**97**,**24**);**

line**(**177**,**1**,**177**,**24**);**

line**(**274**,**1**,**274**,**24**);**

line**(**354**,**1**,**354**,**24**);**

line**(**451**,**1**,**451**,**24**);**

line**(**531**,**1**,**531**,**24**);**

line**(**1**,**24**,**16**,**24**);**

line**(**1**,**72**,**16**,**72**);**

line**(**1**,**120**,**16**,**120**);**

line**(**1**,**168**,**16**,**168**);**

line**(**1**,**216**,**16**,**216**);**

line**(**1**,**264**,**16**,**264**);**

line**(**1**,**312**,**16**,**312**);**

line**(**1**,**360**,**16**,**360**);**

line**(**1**,**408**,**16**,**408**);**

line**(**1**,**456**,**16**,**456**);**

setcolor**(**RED**);**

line**(**622**,**7**,**634**,**19**);**

line**(**622**,**19**,**634**,**7**);**

puthz**(**121**,**5**,"文件",** 16**,** 16**,**DARKGRAY**);**

puthz**(**298**,**5**,"编辑",** 16**,** 16**,**DARKGRAY**);**

puthz**(**475**,**5**,"帮助",** 16**,** 16**,**DARKGRAY**);**

puthz**(**24**,**458**,"行",** 16**,** 16**,**DARKGRAY**);**

puthz**(**100**,**458**,"列",** 16**,** 16**,**DARKGRAY**);**

**}**

**void** drawfile**()** *//点击“文件”显示的子菜单*

**{**

setcolor**(**LIGHTGRAY**);**

bar**(**97**,**25**,**177**,**118**);**

setcolor**(**DARKGRAY**);**

line**(**97**,**25**,**177**,**25**);***//新建按钮*

line**(**97**,**48**,**177**,**48**);***//打开按钮*

line**(**97**,**71**,**177**,**71**);***//保存按钮*

line**(**97**,**94**,**177**,**94**);***//另存为按钮*

puthz**(**121**,**29**,"新建",** 16**,** 16**,**DARKGRAY**);**

puthz**(**121**,**52**,"打开",** 16**,** 16**,**DARKGRAY**);**

puthz**(**121**,**75**,"保存",** 16**,** 16**,**DARKGRAY**);**

puthz**(**117**,**98**,"另存为",** 16**,** 16**,**DARKGRAY**);**

**}**

**void** drawedit**()***//点击“编辑”显示的子菜单*

**{**

setcolor**(**LIGHTGRAY**);**

bar**(**274**,**25**,**354**,**166**);**

setcolor**(**DARKGRAY**);**

line**(**274**,**25**,**354**,**25**);***//剪切按钮*

line**(**274**,**48**,**354**,**48**);***//复制按钮*

line**(**274**,**71**,**354**,**71**);***//粘贴按钮*

line**(**274**,**94**,**354**,**94**);***//清除按钮*

line**(**274**,**117**,**354**,**117**);***//查找按钮*

line**(**274**,**140**,**354**,**140**);***//编译按钮*

puthz**(**298**,**29**,"剪切",** 16**,** 16**,**DARKGRAY**);**

puthz**(**298**,**52**,"复制",** 16**,** 16**,**DARKGRAY**);**

puthz**(**298**,**75**,"粘贴",** 16**,** 16**,**DARKGRAY**);**

puthz**(**298**,**99**,"替换",** 16**,** 16**,**DARKGRAY**);**

puthz**(**298**,**121**,"查找",** 16**,** 16**,**DARKGRAY**);**

puthz**(**298**,**144**,"清除",** 16**,** 16**,**DARKGRAY**);**

**}**

**void** drawhelp**()** *//点击“帮助”显示的子菜单*

**{**

setcolor**(**LIGHTGRAY**);**

bar**(**451**,**25**,**531**,**71**);**

setcolor**(**DARKGRAY**);**

line**(**451**,**25**,**531**,**25**);***//关于按钮*

line**(**451**,**48**,**531**,**48**);***//帮助按钮*

puthz**(**475**,**29**,"关于",** 16**,** 16**,**DARKGRAY**);**

puthz**(**475**,**52**,"帮助",** 16**,** 16**,**DARKGRAY**);**

**}**

**void** drawrc**(int** r**,** **int** c**)**

**{**

**char** row**[**3**];**

**char** col**[**3**];**

itoa**(**r**,**row**,**10**);**

itoa**(**c**,**col**,**10**);**

setfillstyle**(**1**,** LIGHTGRAY**);**

bar**(**40**,** 458**,** 90**,** 474**);**

bar**(**116**,**458**,**166**,**474**);**

settextstyle**(**0**,**0**,**1**);**

setcolor**(**WHITE**);**

outtextxy**(**48**,**464**,**row**);**

outtextxy**(**124**,**464**,**col**);**

**}**

**void** drawwin**(int** x**,int** y**)**

**{**

setfillstyle**(**1**,** LIGHTGRAY**);**

bar**(**x**,**y**,**x**+**63**,**y**+**51**);**

setcolor**(**DARKGRAY**);**

line**(**x**,**y**,**x**+**63**,**y**);**

line**(**x**+**63**,**y**,**x**+**63**,**y**+**51**);**

line**(**x**,**y**+**51**,**x**+**63**,**y**+**51**);**

line**(**x**,**y**,**x**,**y**+**51**);**

line**(**x**,**y**+**17**,**x**+**63**,**y**+**17**);**

line**(**x**,**y**+**34**,**x**+**63**,**y**+**34**);**

puthz**(**x**+**16**,**y**+**1**,"复制",** 16**,** 16**,**DARKGRAY**);**

puthz**(**x**+**16**,**y**+**18**,"剪切",** 16**,** 16**,**DARKGRAY**);**

puthz**(**x**+**16**,**y**+**35**,"粘贴",** 16**,** 16**,**DARKGRAY**);**

**}**

**void** full\_skbar**()***//画一个全覆盖的拖动条*

**{**

setfillstyle**(**1**,** LIGHTGRAY**);**

bar**(**624**,**25**,**640**,**41**);***//上移按钮*

bar**(**624**,**440**,**640**,**456**);***//下移按钮*

setcolor**(**WHITE**);**

line**(**632**,**25**,**624**,**33**);**

line**(**632**,**25**,**640**,**33**);**

line**(**632**,**25**,**632**,**41**);***//组成向上箭头*

line**(**632**,**456**,**624**,**448**);**

line**(**632**,**456**,**640**,**448**);**

line**(**632**,**456**,**632**,**440**);***//组成向下箭头*

setfillstyle**(**1**,** DARKGRAY**);**

bar**(**624**,**41**,**640**,**440**);***//拖动条占满预留位置*

**}**

7.file.c

#include<common.h>

#include<draw.h>

**char** filename**[**30**]={**0**};***//定义全局变量filename*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*【文件模块】*

*编写人：王思为*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**int** FreeAll**(**Hnode**\*** p**)** *//删除p及后面的行节点和节点*

**{**

**if** **(**p**->**next**)**

Freenode**(**p**->**next**);**

**if** **(**p**->**nextl**)**

FreeAll**(**p**->**nextl**);**

free**(**p**);**

**return;**

**}**

**int** Freenode**(**node**\*** p**)** *//删除p后节点*

**{**

**if** **(**p**->**next**)** Freenode**(**p**->**next**);**

free**(**p**);**

**return;**

**}**

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*函数：opens*

*功能：通过链表里读取文件里的字符，将其全部保存到链表里后通过view函数将其显示到屏幕上*

*打开失败会输出提示*

*参数：链表Hnode\*Hp*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

Hnode**\*** opens**(**Hnode**\*** Hp**,**Hnode **\*\***Chead**)** *//打开文件*

**{**

FILE**\*** fp**;**

Hnode**\*** q11**,** **\*** q22**;**

node**\*** p11**,** **\*** p22**;**

**int** temp**;** *//temp用于接收从文件中获得的字符*

**int** Hflg**=**0**;**

**int** Hflg2**=**0**;** *//Hflg，Hflg2用来判定是否是第一行*

**int** count **=** 0**;** *//count用于记录某行已经输入的字符数*

**int** flags **=** 1**;**

**int** i **=** 0**;**

cleardevice**();***//清屏*

drawmenu**();**

delcursor**();***//删除光标，准备输出提示*

puthz**(**50**,**50**,"输入一个文件名",**16**,**16**,**WHITE**);**

setcolor**(**WHITE**);**

outtextxy**(**163**,**50**,"(e.g. c:\\test.c)")** **;***//打开文件示例*

gettxtname**(**filename**,**310**,**50**);**

setcolor**(**WHITE**);**

fp **=** fopen**(**filename**,** **"r");***//以只读方式打开文件，filename必须要存在*

**if** **(**fp **==** NULL**)***//打开文件失败*

**{**

cleardevice**();**

puthz**(**288**,**232**,"打开失败",** 16**,** 16**,**WHITE**);**

getch**();**

drawmenu**();**

**return** Hp**;**

**}**

**else**

**{**

FreeAll**(**Hp**->**nextl**);** *//释放空间，准备把文件读入链表*

Hp**->**next **=** NULL**;**

Hp**->**nextl **=** NULL**;**

**}**

**while** **(!**feof**(**fp**))** *//feof（fp）文件结束返回0，反之非零*

**{**

count **=** 0**;**

flags **=** 1**;**

q22 **=** **(**Hnode**\*)**malloc**(sizeof(**Hnode**));***//新建一个行单链表中的节点*

p11 **=** **(**node**\*)**malloc**(sizeof(**node**));** *//新建一个列单链表中的节点*

p11**->**txt**[**0**]** **=** 13**;**

p11**->**sign**=**0**;**

q22**->**next **=** p11**;**

q22**->**nextl **=** NULL**;** *//将存储了字符的新列单链表与行单链表中的新节点建立关联*

**if(**Hflg**==**0**)***//这里判断是否是第一行，第一行操作结束后不再执行此行操作*

**{**

Hp**=**q22**;**

Hflg**=**1**;**

**}**

temp **=** fgetc**(**fp**);** *//获得文件中的字符*

**while** **((**temp **!=** 10 **)&&** **(**temp **!=** 13**)&&** **(**count **<=** 76**)** **&&** **(!**feof**(**fp**)))**

*//若循环结束，则说明在单链表中一行处理完毕，因此开始新行*

**{**

*//新建一个列单链表中的节点*

**if** **(**flags **==** 1**)** *//flags==1说明在第一个节点输入*

**{**

**if** **(**temp **==** 0x09**)** *//如果检测到tab，将其全部替换成空格*

**{**

p11**->**txt**[**0**]** **=** ' '**;**

p11**->**txt**[**1**]=**'\0'**;**

p11**->**sign**=**0**;**

count**++;**

**for** **(**i **=** 0**;** i **<** 3**;** i**++)**

**{**

p22 **=** **(**node**\*)**malloc**(sizeof(**node**));**

p22**->**txt**[**0**]** **=**' '**;**

p22**->**txt**[**1**]=**'\0'**;**

p22**->**next **=** NULL**;**

p11**->**next **=** p22**;**

p22**=**p22**->**next**;**

p11 **=** p11**->**next**;**

count**++;**

**}**

temp **=** fgetc**(**fp**);**

**}**

**else**

**{**

p11**->**txt**[**0**]** **=** temp**;**

p11**->**txt**[**1**]=**'\0'**;**

p11**->**sign**=**0**;**

p11**->**flag**=**WHITE**;**

p11**->**next **=** NULL**;**

count**++;**

temp **=** fgetc**(**fp**);**

**}**

flags**=**0**;**

**}**

**else** **if(**flags**!=**1**)** *//不在第一个节点输入*

**{**

**if** **(**temp **==** 0x09**)** */\*如果检测到tab，将其全部替换成空格\*/*

**{**

p22 **=** **(**node**\*)**malloc**(sizeof(**node**));**

**for** **(**i **=** 0**;** i **<** 4**;** i**++)**

**{**

**if(**i**!=**0**)** p22 **=** **(**node**\*)**malloc**(sizeof(**node**));**

p22**->**txt**[**0**]** **=**' '**;**

p22**->**txt**[**1**]=**'\0'**;**

p22**->**next **=** NULL**;**

p11**->**next **=** p22**;**

p22**=**p22**->**next**;**

p11 **=** p11**->**next**;**

count**++;**

**}**

temp **=** fgetc**(**fp**);**

**}**

**else**

**{**

**if((**temp **&** 0x80**)==**0**)***//temp与0x80作与，若其值为0，表示是非中文字符*

**{**

p22 **=** **(**node**\*)**malloc**(sizeof(**node**));**

p22**->**txt**[**0**]** **=** temp**;**

p22**->**txt**[**1**]=**'\0'**;**

p22**->**sign**=**0**;**

p22**->**flag**=**WHITE**;**

p22**->**next **=** NULL**;**

p11**->**next **=** p22**;**

p11 **=** p22**;**

count**++;**

**}**

**else** **if((**temp **&** 0x80**)!=**0**)***//其值不为0，表示是中文字符*

**{**

p22 **=** **(**node**\*)**malloc**(sizeof(**node**));**

p22**->**txt**[**0**]=**temp**;**

temp **=** fgetc**(**fp**);**

p22**->**txt**[**1**]=**temp**;**

p22**->**txt**[**2**]=**'\0'**;**

p22**->**sign**=**1**;**

p22**->**next **=** NULL**;**

p11**->**next **=** p22**;**

p11 **=** p22**;**

count**+=**2**;** *//一个中文字符占两个字节*

**}**

temp **=** fgetc**(**fp**);**

**}**

**}**

**}**

**if** **(**count **>** 0**)**

**{**

*//若为换行符，将其转换为回车符，因为在程序中，是按回车符处理的*

p22 **=** **(**node**\*)**malloc**(sizeof(**node**));**

p22**->**txt**[**0**]=** 13**;**

p22**->**sign**=**0**;**

p22**->**next **=** NULL**;**

p11**->**next **=** p22**;**

**if(**Hflg2**==**0**)**

**{**

q11 **=** q22**;**

Hflg2**=**1**;**

**}**

**else**

**{**

q11**->**nextl **=** q22**;**

q11 **=** q22**;**

**}**

**}**

**else**

**{**

*//处理单行只有一个空格或换行符的情况*

p11**->**next **=** NULL**;**

q22**->**next **=** p11**;**

q22**->**nextl **=** NULL**;**

q11**->**nextl **=** q22**;**

q11 **=** q22**;**

**}**

**}**

view**(**Hp**);***//将链表显示在可输入范围内*

**\***Chead**=**Hp**;**

**return** Hp**;**

**}**

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*函数：newbuild*

*功能：新建一个文件，通过 fp = fopen(filename, "wb+")实现*

*若输入的文件已存在，将提示该文件存在，并直接打开此文件*

*参数：链表Hnode\*Hp*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**int** newbuild**(**Hnode**\*** Hp**,**Hnode **\*\***Chead**)**

**{**

FILE**\*** fp**;**

Hnode**\*** q11**,** **\*** q22**;**

node**\*** p11**,** **\*** p22**;**

**int** temp**;**

**int** Hflg**=**0**;**

**int** Hflg2**=**0**;** *//Hflg，Hflg2用来判定是否是第一行*

**int** count **=** 0**;**

**int** flags **=** 1**;**

**int** i **=** 0**;**

cleardevice**();**

drawmenu**();**

delcursor**();***//删除光标，准备输出提示*

puthz**(**50**,**50**,"请输入您想新建文件的文件名",**16**,**16**,**WHITE**);**

setcolor**(**WHITE**);**

gettxtname**(**filename**,**310**,**50**);**

setcolor**(**WHITE**);**

fp **=** fopen**(**filename**,** **"r");***//达到"若没有这个文件，则建立一个文件"的目的*

**if** **(**fp **==** NULL**)***//不存在该文件*

**{**

cleardevice**();***//清屏*

drawmenu**();**

setcolor**(**WHITE**);**

fp **=** fopen**(**filename**,** **"wb+");**

**}**

**else**

**{**

cleardevice**();***//清屏*

drawmenu**();**

delcursor**();***//删除光标，准备输出提示*

setcolor**(**WHITE**);**

puthz**(**222**,**202**,"新建失败",**16**,**16**,**WHITE**);**

puthz**(**222**,**218**,"因为该文件已经存在",**16**,**16**,**WHITE**);**

puthz**(**222**,**234**,"正在为您打开该文件",**16**,**16**,**WHITE**);**

puthz**(**222**,**250**,"请稍候",**16**,**16**,**WHITE**);**

outtextxy**(**270**,**254**,".");**

delay**(**500**);**

outtextxy**(**286**,**254**,".");**

delay**(**500**);**

outtextxy**(**302**,**254**,".");**

delay**(**500**);**

outtextxy**(**318**,**254**,".");**

delay**(**500**);**

fp **=** fopen**(**filename**,** **"r");***//以只读方式打开文件，filename必须要存在*

FreeAll**(**Hp**->**nextl**);**

Hp**->**next **=** NULL**;**

Hp**->**nextl **=** NULL**;**

**while** **(!**feof**(**fp**))** *//feof（fp）文件结束返回0，反之非零*

**{**

count **=** 0**;**

flags **=** 1**;**

q22 **=** **(**Hnode**\*)**malloc**(sizeof(**Hnode**));***//新建一个行单链表中的节点*

p11 **=** **(**node**\*)**malloc**(sizeof(**node**));** *//新建一个列单链表中的节点\**

p11**->**txt**[**0**]** **=** 13**;**

p11**->**sign**=**0**;**

q22**->**next **=** p11**;**

q22**->**nextl **=** NULL**;** *//将存储了字符的新列单链表与行单链表中的新节点建立关联*

**if(**Hflg**==**0**)***//这里判断是否是第一行，第一行操作结束后不再执行此行操作*

**{**

Hp**=**q22**;**

Hflg**=**1**;**

**}**

temp **=** fgetc**(**fp**);**

**while** **((**temp **!=** 10 **)&&** **(**temp **!=** 13**)&&** **(**count **<=** 76**)** **&&** **(!**feof**(**fp**)))**

*//若循环结束，则说明在单链表中一行处理完毕，因此开始新行*

**{**

*//新建一个列单链表中的节点*

**if** **(**flags **==** 1**)** *//flags==1说明在第一个节点输入*

**{**

**if** **(**temp **==** 0x09**)** *//如果检测到tab，将其全部替换成空格*

**{**

p11**->**txt**[**0**]** **=** ' '**;**

p11**->**txt**[**1**]=**'\0'**;**

p11**->**sign**=**0**;**

count**++;**

**for** **(**i **=** 0**;** i **<** 3**;** i**++)**

**{**

p22 **=** **(**node**\*)**malloc**(sizeof(**node**));**

p22**->**txt**[**0**]** **=**' '**;**

p22**->**txt**[**1**]=**'\0'**;**

p22**->**next **=** NULL**;**

p11**->**next **=** p22**;**

p22**=**p22**->**next**;**

p11 **=** p11**->**next**;**

count**++;**

**}**

temp **=** fgetc**(**fp**);**

**}**

**else**

**{**

p11**->**txt**[**0**]** **=** temp**;**

p11**->**txt**[**1**]=**'\0'**;**

p11**->**next **=** NULL**;**

count**++;**

temp **=** fgetc**(**fp**);**

**}**

flags**=**0**;**

**}**

**else** **if(**flags**!=**1**)** *//不在第一个节点输入*

**{**

**if** **(**temp **==** 0x09**)** *//如果检测到tab，将其全部替换成空格*

**{**

p22 **=** **(**node**\*)**malloc**(sizeof(**node**));**

**for** **(**i **=** 0**;** i **<** 4**;** i**++)**

**{**

**if(**i**!=**0**)** p22 **=** **(**node**\*)**malloc**(sizeof(**node**));**

p22**->**txt**[**0**]** **=**' '**;**

p22**->**txt**[**1**]=**'\0'**;**

p22**->**next **=** NULL**;**

p11**->**next **=** p22**;**

p22**=**p22**->**next**;**

p11 **=** p11**->**next**;**

count**++;**

**}**

temp **=** fgetc**(**fp**);**

**}**

**else**

**{**

**if((**temp **&** 0x80**)==**0**)***//表示是非中文字符*

**{**

p22 **=** **(**node**\*)**malloc**(sizeof(**node**));**

p22**->**txt**[**0**]** **=** temp**;**

p22**->**txt**[**1**]=**'\0'**;**

p22**->**sign**=**0**;**

p22**->**flag**=**WHITE**;**

p22**->**next **=** NULL**;**

p11**->**next **=** p22**;**

p11 **=** p22**;**

count**++;**

**}**

**else** **if((**temp **&** 0x80**)!=**0**)***//表示是中文字符*

**{**

p22 **=** **(**node**\*)**malloc**(sizeof(**node**));**

p22**->**txt**[**0**]=**temp**;**

temp **=** fgetc**(**fp**);**

p22**->**txt**[**1**]=**temp**;**

p22**->**txt**[**2**]=**'\0'**;**

p22**=**p22**->**next**;**

p22**->**sign**=**1**;**

p22**->**next **=** NULL**;**

p11**->**next **=** p22**;**

p11 **=** p22**;**

count**++;**

count**++;**

**}**

temp **=** fgetc**(**fp**);**

**}**

**}**

**}**

**if** **(**count **>** 0**)**

**{**

*//若为换行符，将其转换为回车符，因为在程序中，是按回车符处理的*

p22 **=** **(**node**\*)**malloc**(sizeof(**node**));**

p22**->**txt**[**0**]=** 13**;**

p22**->**next **=** NULL**;**

p22**->**sign**=**0**;**

p22**->**flag**=**WHITE**;**

p11**->**next **=** p22**;**

**if(**Hflg2**==**0**)**

**{**

q11 **=** q22**;**

Hflg2**=**1**;**

**}**

**else**

**{**

q11**->**nextl **=** q22**;**

q11 **=** q22**;**

**}**

**}**

**else**

**{**

*//处理单行只有一个空格或换行符的情况*

p11**->**next **=** NULL**;**

q22**->**next **=** p11**;**

q22**->**nextl **=** NULL**;**

q11**->**nextl **=** q22**;**

q11 **=** q22**;**

**}**

**}**

view**(**Hp**);**

**\***Chead**=**Hp**;**

**}**

**}**

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*函数：save*

*功能：将修改的字符保存进链表（即修改链表）以达到保存文件的目的*

*参数：链表Hnode\*Hp*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**int** save**(**Hnode**\*** Hp**)**

**{**

FILE**\*** fp**;**

Hnode**\*** q**;**

node**\*** p**;**

q**=**Hp**;**

fp **=** fopen**(**filename**,** **"w");**

**if** **(**fp **==** NULL**)***//打开文件失败*

**{**

cleardevice**();**

puthz**(**288**,**232**,"打开失败",** 16**,** 16**,**WHITE**);**

getch**();**

drawmenu**();**

**return** 0**;**

**}**

**do**

**{**

p **=** q**->**next**;** *//指向node类型的数据*

**while** **(**p **!=** NULL**)**

**{**

**if(**p**->**sign**==**0**)**

**{**

**if(**p**->**txt**[**0**]==**13**)**

fputc**(**'\n'**,** fp**);**

**else**

fprintf**(**fp**,"%c",**p**->**txt**[**0**])** **;**

**}**

**else** **if(**p**->**sign**==**1**)**

**{**

fprintf**(**fp**,"%c",**p**->**txt**[**0**])** **;**

fprintf**(**fp**,"%c",**p**->**txt**[**1**])** **;**

**}**

p **=** p**->**next**;**

fflush**(**fp**);**

**}**

q **=** q**->**nextl**;**

**}** **while** **(**q**!=**NULL**);**

fclose**(**fp**);** *//关闭此文件*

drawmenu**();**

**return;**

**}**

**int** saveas**(**Hnode**\*** Hp**)**

**{**

FILE**\*** fp**;**

Hnode**\*** q**;**

node**\*** p**;**

**char** newname**[**15**];**

cleardevice**();***//清屏*

drawmenu**();**

puthz**(**50**,**50**,"请输入您想另存为文件的文件名",**16**,**16**,**WHITE**);**

setcolor**(**WHITE**);**

gettxtname**(**newname**,**280**,**50**);**

q**=**Hp**;**

*//通过键盘修改链表*

fp **=** fopen**(**newname**,** **"w");**

**if** **(**fp **==** NULL**)***//打开文件失败*

**{**

cleardevice**();**

puthz**(**288**,**232**,"打开失败",** 16**,** 16**,**WHITE**);**

getch**();**

drawmenu**();**

**return** 0**;**

**}**

**do**

**{**

p **=** q**->**next**;** *//指向node类型的数据*

**while** **(**p **!=** NULL**)**

**{**

**if(**p**->**sign**==**0**)**

**{**

**if(**p**->**txt**[**0**]==**13**)**

fputc**(**'\n'**,** fp**);**

**else**

fprintf**(**fp**,"%c",**p**->**txt**[**0**])** **;**

**}**

**else** **if(**p**->**sign**==**1**)**

**{**

fprintf**(**fp**,"%c",**p**->**txt**[**0**])** **;**

fprintf**(**fp**,"%c",**p**->**txt**[**1**])** **;**

**}**

p **=** p**->**next**;**

fflush**(**fp**);**

**}**

q **=** q**->**nextl**;**

**}** **while** **(**q**!=**NULL**);**

fclose**(**fp**);** *//关闭此文件*

drawmenu**();**

**return;**

**}**

8.help.c

#include<common.h>

#include<HELP.h>

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*【画图模块】*

*编写人：王思为*

*该板块保存了汉化bc的帮助功能，显示该软件的主要功能*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**void** help**()**

**{**

cleardevice**();**

setfillstyle**(**1**,** WHITE**);**

bar**(**0**,**0**,**640**,**480**);**

setfillstyle**(**1**,** DARKGRAY**);**

bar**(**5**,**5**,**61**,**45**);**

puthz**(**17**,**17**,"返回",**16**,**16**,**WHITE**);**

delcursor**();**

puthz**(**17**,**60**,"欢迎使用",**16**,**16**,**DARKGRAY**);**

outtextxy**(**82**,**64**,"BC");**

puthz**(**99**,**60**,"，本程序由鼠标和键盘共同控制。",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**80**,"鼠标点击屏幕位置，将移动光标；点击菜单栏，通过点击菜单弹出框进行对应操作。",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**100**,"新建、打开文件：",**16**,**16**,**DARKGRAY**);**

puthz**(**49**,**120**,"点击“文件”菜单，在菜单框中点击对应按钮，输入要打开或新建的文件路径及文",**16**,**16**,**DARKGRAY**);**

puthz**(**49**,**140**,"件名，再按回车即可",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**160**,"保存文件：",**16**,**16**,**DARKGRAY**);**

puthz**(**49**,**180**,"打开文件编辑之后，点击文件菜单，点击保存即可。",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**200**,"另存为：",**16**,**16**,**DARKGRAY**);**

puthz**(**49**,**220**,"点击文件菜单，点击另存为，输入另存为的路径与文件名按下回车即可。",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**240**,"按普通按键：在光标位置输入字符；",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**260**,"按上下左右键：移动光标。",**16**,**16**,**DARKGRAY**);**

outtextxy**(**17**,**284**,"CTRL+");**

puthz**(**57**,**280**,"左：向左选中一个文本",**16**,**16**,**DARKGRAY**);**

outtextxy**(**17**,**304**,"CTRL+");**

puthz**(**57**,**300**,"右：向右选中一个文本",**16**,**16**,**DARKGRAY**);**

outtextxy**(**17**,**324**,"CTRL+X");**

puthz**(**65**,**320**,"：剪切键快捷键",**16**,**16**,**DARKGRAY**);**

outtextxy**(**17**,**344**,"CTRL+C");**

puthz**(**65**,**340**,"：复制键快捷键",**16**,**16**,**DARKGRAY**);**

outtextxy**(**17**,**364**,"CTRL+V");**

puthz**(**65**,**360**,"：粘贴键快捷键",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**380**,"按",**16**,**16**,**DARKGRAY**);**

outtextxy**(**33**,**384**,"F4");**

puthz**(**50**,**380**,"键进入汉字输入模式，按",**16**,**16**,**DARKGRAY**);**

outtextxy**(**245**,**384**,"ESC");**

puthz**(**270**,**384**,"退出",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**400**,"查找：点击编辑菜单栏，点击“查找”，输入要查找的字符串，按回车键",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**420**,"替换：点击编辑菜单栏，点击“替换”，输入要替换的字符串，按回车键，",**16**,**16**,**DARKGRAY**);**

puthz**(**66**,**440**,"输入新的字符串，按回车即可",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**460**,"单击鼠标右键可以从弹出框快捷选择复制、粘贴、剪切。",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**480**,"也可以点击编辑菜单栏，在弹出框里进行复制、粘贴、剪切、查找",**16**,**16**,**DARKGRAY**);**

**return;**

**}**

**void** about**()**

**{**

cleardevice**();**

setfillstyle**(**1**,** WHITE**);**

bar**(**0**,**0**,**640**,**480**);**

setfillstyle**(**1**,** DARKGRAY**);**

bar**(**5**,**5**,**61**,**45**);**

puthz**(**17**,**17**,"返回",**16**,**16**,**WHITE**);**

delcursor**();**

puthz**(**17**,**60**,"关于该汉化",**16**,**16**,**DARKGRAY**);**

outtextxy**(**97**,**65**,"BorlandC++");**

puthz**(**178**,**60**,"的说明",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**80**,"鉴于传统的",**16**,**16**,**DARKGRAY**);**

outtextxy**(**97**,**85**,"BorlandC++");**

puthz**(**178**,**80**,"编辑器不支持中文",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**100**,"我组编写的编辑器在一定程度上弥补了这一缺陷",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**120**,"我们实现了在界面里输入中文的功能",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**140**,"另外在此编辑器中用户可以做到诸如复制，粘贴，剪切等基本操作功能",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**160**,"我们还添加了文件的相关操作",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**180**,"用户可以选择保存，另存为，打开文件等操作",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**200**,"为了使得用户操作更方便，我们添加了查找功能",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**240**,"班级：",**16**,**16**,**DARKGRAY**);**

puthz**(**66**,**240**,"自动化",**16**,**16**,**DARKGRAY**);**

outtextxy**(**115**,**244**,"2208");**

puthz**(**148**,**240**,"班",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**260**,"主题：",**16**,**16**,**DARKGRAY**);**

outtextxy**(**65**,**264**,"BorlandC++");**

puthz**(**147**,**260**,"编辑器的汉化",**16**,**16**,**DARKGRAY**);**

puthz**(**17**,**280**,"小组成员：",**16**,**16**,**DARKGRAY**);**

puthz**(**104**,**280**,"胡耀文",**16**,**16**,**DARKGRAY**);**

puthz**(**161**,**280**,"王思为",**16**,**16**,**DARKGRAY**);**

**return;**

**}**

9.seekbar.c

#include<common.h>

#include"seekbar.h"

**int** count**=**0**;***//记录up和down的次数用以在不同位置显示拖动条*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*【滑动条模块】*

*编写人：王思为*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**void** drawskbar**(**Hnode**\*** Hhead**)**

**{**

*//拖动条栏设计为16×432的长方形板块。上下各取一个16×16的正方形作为上下按键*

*//故实际拖动条图形占空间为16×400的长方形*

**int** r**;***//记录总行数*

**int** x**;***//拖动条应有长度*

r**=**getHlen**(**Hhead**);**

setfillstyle**(**1**,** LIGHTGRAY**);**

bar**(**624**,**25**,**640**,**41**);***//上移按钮*

bar**(**624**,**440**,**640**,**456**);***//下移按钮*

setcolor**(**BLUE**);**

line**(**632**,**25**,**624**,**33**);**

line**(**632**,**25**,**640**,**33**);**

line**(**632**,**25**,**632**,**41**);***//组成向上箭头*

line**(**632**,**456**,**624**,**448**);**

line**(**632**,**456**,**640**,**448**);**

line**(**632**,**456**,**632**,**440**);***//组成向下箭头*

line**(**624**,**25**,**640**,**25**);**

line**(**640**,**25**,**640**,**41**);**

line**(**624**,**456**,**640**,**456**);**

line**(**640**,**440**,**640**,**456**);**

**if(**r**<=**27**)**

**{**

setfillstyle**(**1**,** DARKGRAY**);**

bar**(**624**,**41**,**640**,**440**);***//拖动条占满预留位置*

**}**

**else** **if(**r**>**27**)**

**{**

x**=(int)**10773**/**r**;** *//获得拖动条长度 (10773=27\*339)*

setfillstyle**(**1**,** DARKGRAY**);**

bar**(**624**,**41**,**640**,**41**+**x**);**

**}**

**}**

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*函数：up\_page/down\_page*

*功能： 由于文本框的大小只能容纳27行文本，需要进行翻页的操作，也就是更改显示的第一个行单链表。*

*行链表的头结点称为Hhead。在行链表头结点之外新建一个行链表指针，称作Chead。*

*用Chead指向的位置来指定当前应当显示的第一行。因为屏幕大小的限制，只会显示Chead指向的行链表及其后26行的文本。*

*在传递参数给view函数等函数时，将Chead代替Hhead传入。*

*以上两个函数分别是指将Chead上移/下移一个单位*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

**int** up\_page**(**Hnode**\*** Hhead**,**Hnode **\*\***Chead**)**

**{**

**int** r**;***//记录总行数*

**int** x**;***//拖动条应有长度*

Hnode**\*** q**;** *//按行显示保存在单链表中的文本字符,q为指向行单链表中第一个节点的指针*

**int** i**=**0**;**

q**=**Hhead**;**

r**=**getHlen**(**Hhead**);**

x**=(int)**10773**/**r**;**

**if((((int)(**41**+**count**\***399**/**r**))>=**41**)&&(**count**>**0**))**

**{**

**for(**i**=**0**;**i**<(**count**-**1**);**i**++)**

**{**

q**=**q**->**nextl**;**

**}**

**\***Chead**=**q**;**

newskbar**(**Hhead**,**r**);**

count**--;**

**}**

**return;**

**}**

**int** down\_page**(**Hnode**\*** Hhead**,**Hnode **\*\***Chead**)**

**{**

**int** r**;***//记录总行数*

**int** x**;***//拖动条应有长度*

r**=**getHlen**(**Hhead**);**

x**=(int)**10773**/**r**;**

**if((((int)(**41**+**x**+**count**\***399**/**r**))<=**440**)&&(**count**<(**r**-**27**)))**

**{**

**\***Chead**=(\***Chead**)->**nextl**;**

newskbar**(**Hhead**,**r**);**

count**++;**

**}**

**return;**

**}**

**int** newskbar**(**Hnode**\*** Hhead**,int** r**)***//画新的拖动条*

**{**

**int** x**;***//拖动条应有长度*

r**=**getHlen**(**Hhead**);**

x**=(int)**10773**/**r**;**

setfillstyle**(**1**,** LIGHTGRAY**);**

bar**(**624**,**41**,**640**,**440**);**

**if((((int)(**41**+**x**+**count**\***399**/**r**)<=**440**))&&(((int)(**41**+**count**\***399**/**r**)>=**41**)))**

**{**

setfillstyle**(**1**,** DARKGRAY**);**

bar**(**624**,(int)(**41**+**count**\***399**/**r**),**640**,(int)(**41**+**x**+**count**\***399**/**r**));***//单位长度为399/r (像素每行）*

**}**

**else** **if((((int)(**41**+**x**+**count**\***399**/**r**))>**440**))**

**{**

setfillstyle**(**1**,** DARKGRAY**);**

bar**(**624**,**440**-**x**,**640**,**440**);**

**}**

**else** **if((((int)(**41**+**count**\***399**/**r**))<**41**))**

**{**

setfillstyle**(**1**,** DARKGRAY**);**

bar**(**624**,**41**,**640**,**41**+**x**);**

**}**

**return;**

**}**

**int** click\_skbar**(**Hnode**\*** Hhead**,**Hnode **\*\***Chead**)**

**{**

**int** r**;***//记录总行数*

**int** x**;***//拖动条应有长度*

**int** y1\_skbar**;** *//目前拖动条上端位置*

**int** y2\_skbar**;***//目前拖动条下端位置*

**int** change\_count1**;***//记录点击后count的改变量*

**int** change\_count2**;***//记录点击后count的改变量*

**int** i**=**0**;**

Hnode**\*** q**;**

r**=**getHlen**(**Hhead**);**

x**=(int)**10773**/**r**;**

q**=**Hhead**;**

y1\_skbar**=(int)(**41**+**x**+**count**\***399**/**r**-**x**);**

y2\_skbar**=(int)(**41**+**x**+**count**\***399**/**r**);**

change\_count1**=(int)((**y1\_skbar**-**MouseY**)\***r**/**399**);**

change\_count2**=(int)((**MouseY**-**y2\_skbar**)\***r**/**399**);**

**if(**MouseY**<**y1\_skbar**)**

**{**

**for(**i**=**0**;**i**<(**change\_count1**-**1**);**i**++)**

**{**

q**=**q**->**nextl**;**

**}**

**\***Chead**=**q**;**

newskbar**(**Hhead**,**r**);**

count**--;**

**}**

**else** **if(**MouseY**>**y2\_skbar**)**

**{**

**for(**i**=**0**;**i**<**change\_count2**;**i**++)**

**{**

**\***Chead**=(\***Chead**)->**nextl**;**

newskbar**(**Hhead**,**r**);**

count**++;**

**}**

**}**

**else** **if((**MouseY**>=**y1\_skbar**)&&(**MouseY**<=**y2\_skbar**))**

**{**

**return;**

**}**

**return;**

**}**

10.view.c

#include<common.h>

#include<view.h>

*//打印一次当前屏幕显示区域的代码*

**void** view**(**Hnode**\*** Chead**)**

**{**

**int** x**=**17**,**y**=**25**,**r**=**1**,**c**=**1**;**

node**\*** p**;** */\*p为保存列单链表节点元素地址的指针\*/*

Hnode**\*** q**;** */\*按行显示保存在单链表中的文本字符,q为指向行单链表中第一个节点的指针\*/*

clrmous**(**MouseX**,**MouseY**);** *//清除鼠标*

setfillstyle**(**1**,** BLUE**);**

bar**(**17**,**25**,**624**,**455**);***//字符区域呈蓝色*

**for(**q**=**Chead**;**q**!=**NULL**&&**r**<=**27**;**r**++)**

**{**

p**=**q**->**next**;**

**for(;(**p**!=**NULL**)&&(**p**->**txt**[**0**]!=**13**);**c**++)**

**{**

**if(**p**->**sign**==**1**)**

**{**

setcolor**(**WHITE**);**

puthz**(**x**,**y**,**p**->**txt**,**16**,** 16**,**WHITE**);**

x**+=**16**;**

c**++;**

**}**

**else** **if(**p**->**sign**==**0**)**

**{**

setcolor**(**p**->**flag**);**

settextstyle**(**0**,**0**,**1**);**

outtextxy**(**x**,**y**+**6**,**p**->**txt**);**

x**+=**8**;**

**}**

p **=** p**->**next**;** *//列指针指向下一个节点*

**}**

x**=**17**;**

y**+=**16**;**

q **=** q**->**nextl**;** *//行指针指向下一个节点*

**}**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

**return** 0**;**

**}**

*//用于将屏幕上m行n列的字符的底色改变*

**int** colorview**(int** m**,int** n**,**Hnode**\*** Chead**)**

**{**

**int** i**=**0**;**

Hnode**\*** q1**;**

node**\*** p1**;**

q1**=**Chead**;**

**for(**i**=**1**;**i**<**m**;**i**++)**

q1**=**q1**->**nextl**;***//循环使得q1移动到第m行*

p1**=**q1**->**next**;**

**for(**i**=**0**;**i**<**n**-**1**;**i**++)**

**{**

p1**=**p1**->**next**;**

**}**

delcursor**();**

clrmous**(**MouseX**,**MouseY**);**

setfillstyle**(**1**,**LIGHTGRAY**);**

bar**(**17**+**8**\*(**n**-**1**),**25**+**16**\*(**m**-**1**),**24**+**8**\*(**n**-**1**),**40**+**16**\*(**m**-**1**));**

settextstyle**(**0**,**0**,**1**);**

setcolor**(**WHITE**);**

p1**->**txt**[**1**]=**'\0'**;**

outtextxy**(**17**+**8**\*(**n**-**1**),**31**+**16**\*(**m**-**1**),**p1**->**txt**);**

imgflag**=**0**;**

save\_bk\_mou**(**MouseX**,**MouseY**);**

drawmous**(**MouseX**,**MouseY**);**

newcursor**();**

**}**

*//在XY位置输入一串字符并返回字符串长度*

**int** gettxtname**(char\*** s**,int** x**,int** y**)**

**{**

**char** a**;**

**char** bio**[**2**];**

**int** i**=**0**;**

**while(**1**)**

**{**

**if** **(**bioskey**(**1**))***//检测键盘输入*

**{**

a**=**bioskey**(**0**);**

**if(**a**>=**32**&&**a**<**127**)**

**{**

bio**[**0**]=**a**;**

bio**[**1**]=**'\0'**;**

setcolor**(**WHITE**);**

settextstyle**(**0**,**0**,**1**);**

outtextxy**(**x**,**y**,**bio**);**

x**+=**8**;**

s**[**i**]=**a**;**

s**[**i**+**1**]=**'\0'**;**

i**++;**

**}**

**else** **if(**a**==**8**)***//点退格*

**{**

setfillstyle**(**1**,** BLUE**);**

bar**(**x**-**8**,**y**,**x**-**1**,**y**+**16**);**

i**--;**

x**-=**8**;**

**}**

**else** **if(**a**==**13**)***//按回车结束输入*

**{**

**break;**

**}**

**}**

**}**

**return** i**+**1**;**

**}**

*//作用:检查某个坐标位置是否有字符，即节点是否为空 ,返回0为空，返回1为非空；*

**int** checktxt**(int** x**,int** y**,**Hnode**\*** Chead**)**

**{**

**int** i**=**0**;**

Hnode**\*** q1**;**

node**\*** p1**;**

q1**=**Chead**;**

**for(**i**=**1**;**i**<**y**;**i**++)**

**{**

q1**=**q1**->**nextl**;**

**if(**q1**==**NULL**)***//如果这一行不存在返回0*

**return** 0**;**

**}**

p1**=**q1**->**next**;**

**for(**i**=**1**;**i**<**x**;**i**++)**

**{**

**if(**p1**==**NULL**)***//如果这一行存在但是这个位置没有字符返回2*

**return** 2**;**

p1**=**p1**->**next**;**

**if(**p1**->**sign**==**1**)**

i**++;**

**}**

**return** 1**;**

**}**

*//功能：检查一行有多少字符并返回这个值*

**int** getlinelen**(**Hnode **\***q**)**

**{**

**int** i**;**

node**\*** p**;**

p**=**q**->**next**;**

**for(**i**=**0**;(**p**->**txt**[**0**]!=**13**)&&(**p**!=**NULL**);**i**++)**

**{**

p**=**p**->**next**;**

**}**

**return** i**;**

**}**

*//功能：检查一共有多少行并返回行数*

**int** getHlen**(**Hnode **\***Hhead**)**

**{**

Hnode **\***q**;**

**int** i**;**

q**=**Hhead**;**

**for(**i**=**1**;**q**->**nextl**!=**NULL**;**i**++)**

**{**

q**=**q**->**nextl**;**

**}**

**return** i**;**

**}**

11.INTI.h

#ifndef \_\_INIT\_H\_\_

#define \_\_INIT\_H\_\_

#define LEFT 0x4b00 */\*←：光标左移\*/*

#define RIGHT 0x4d00 */\*→：光标右移\*/*

#define DOWN 0x5000 */\*↓：光标下移\*/*

#define UP 0x4800 */\*↑：光标上移\*/*

#define ESC 0x011b */\*ESC键：取消菜单打开操作\*/*

#define ENTER 0x1c0d */\*回车键：换行\*/*

#define TAB 0x0f09 */\*TAB键：插入四个空格\*/*

#define DEL 21248 */\*DEL键：删除当前字符\*/*

#define BACK 3592 */\*BackSpace键：删除当前光标位置前一个字符\*/*

#define CL 29440 */\*ctrl+←键：从右至左，选定文本\*/*

#define CR 29696 */\*ctrl+→键：从左到右，选定文本\*/*

#define Cc 11779 */\*ctrl+c 键：将选定文本，复制一份到剪贴板中\*/*

#define Cv 12054 */\*ctrl+v 键：将剪贴板中的内容复制到当前位置\*/*

#define Cx 11544 */\*ctrl+x 键：对选定文本，执行剪切操作\*/*

#define F1 15104 */\*F1键：打开文件菜单\*/*

#define F2 15360 */\*F2键：打开编辑菜单\*/*

#define F3 15616 */\*F3键：打开帮助菜单\*/*

#define F4 15872 */\*F4键：汉化菜单\*/*

#define F10 17408 */\*F10键:刷新\*/*

#define SPACE 0x3920

#define FIRST 0x0231

#define SECOND 0x0332

#define THIRD 0x0433

#define FOURTH 0x0534

#define FIFTH 0x00635

#define SIXTH 0x0736

#define SEVENTH 0x0837

#define EIGHTH 0x0938

#define NINTH 0x0A39

#define ZIERO 0x0B30

#define MAXPY 7

#define SUCCESS 1

#define FAIL 0

#define LASTLINE 0x4800

#define NEXTLINE 0x5000

**typedef** **struct** node

**{**

**char** txt**[**3**];**

**int** flag**;**

**unsigned** **short** sign**;** *//用于指示节点储存的是中文还是英文：英文为0，汉字为1。*

**}**node**;**

**typedef** **struct** Hnode

**{**

node**\*** next**;**

**struct** Hnode**\*** nextl**;**

**}**Hnode**;**

#endif

# 

# 第九部分 个人总结

**一、组员总结**  
选题的时候对各个课题并没有太多的了解，和队友商量之后决定选择BC汉化这个要求很明确的题目。刚开始深入研究这个课题的需求分析的时候，我发现这个题目并没有选题的时候想的那样简单，看似清晰明了的核心功能实际上并不容易实现。

我们观阅了往届学长的代码，发现他们是基于文本界面的输入，虽然有很多方便的函数，但是并不能有效的进行汉字的编辑，仅仅能通过汉化按键显示原来是乱码的汉字。于是我找了几位老师咨询这个课题的需求，老师们的都提到了汉字的输入与显示。于是我们决定在图形界面下进行文本的操作，这样可以更好的完成汉化的任务。

需求分析之后，开始了正式编辑代码。第一次接触庞大的工程，我和队友都很手足无措，不知道从哪里入手，我们先是确定了明确的分工，确定了程序的大框架。之后我负责文本的编辑工作，于是就从画出可以移动的闪烁的光标入手开始组织代码，一开始进度很慢，光标调试用了一周的时间，文本的插入删除两周都没有稳定的实现，但是在这个过程中我发现自己写代码的能力在不断提升，于是我重新组织了我代码的逻辑，逻辑构建好之后代码的实现就容易了许多，终于在七周的周末基本调好了插入。在前几周卡住的同时我也研究了汉字的输入，并构思了复制粘贴等操作的实现逻辑。在调整好基本的输入以后，我用了半天就实现了复制粘贴的基本操作，后面又进一步优化了代码，并加入了剪切功能。功能的快速实现激发了我写代码的兴趣，于是我在之后的一周又完成了查找和替换功能。之前曾经让我焦头烂额的课设忽然就变得明朗起来了，之后的进度便自然推进了。在和队友合并完所有代码以后，看着能正常运行的程序，不禁感叹：世上无难事，只怕有心人。从最开始的完全不知道从何处开始，到一步一步完成程序的主要内容，也只花了一个多月的时间。

在课设的这一段时间里，我对C语言有了更为深入的理解，从最开始在图形界面加画图的句子调试到后来加断点一步一步运行，并用watch监测指针的位置、字符串的值，我不仅复习了很多上个学期学到的的C语言知识，而且对于没有学的链表的知识也有了较深入的认识，更重要的是学会了面对一个庞大的程序如何去组织结构，设计函数。这次课设，是我大学生活中目前遇到的最为艰难的一次磨砺，也是最有价值的一段回忆。

**二、组员个人总结：**

这次的C语言课程设计是我第一次实际使用C语言进行程序编辑，所以在编写过程中自然是遇到了不小的挑战，在这里，我将记录自己的学习过程和在完成C课设过程中的所思所想。

在选题阶段，我们组选择了“仿BC编辑器的汉化”这一课题。明确选题前后，我对往届学长学姐们的代码进行了浏览，了解了这个课题需要做些什么，此外，我还仔细阅读了学长学姐建议学习的C课设相关知识，如BC的使用，小组内的分工方法，debug的方法技巧等。在了解完这些知识架构后，我们便开始着手完成本次课程设计。

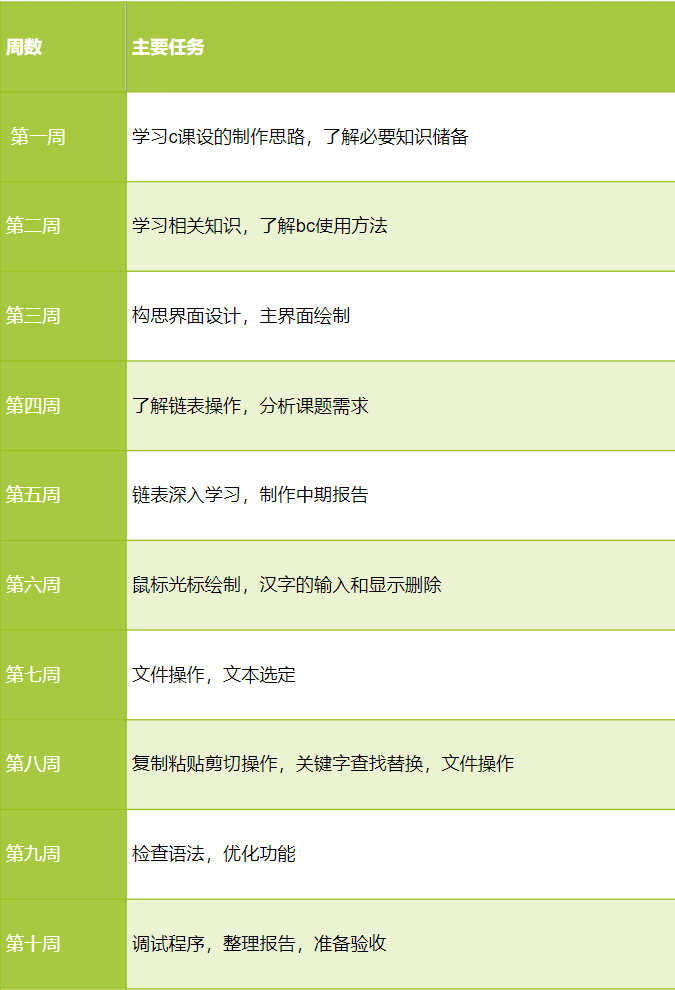
正式推进任务阶段，缺乏经验的我屡屡碰壁，也意识到应该在在上学期学习内容的基础上学习一些可以简化我们的代码的工具。因此，趁着前期任务较轻松（以画图为主），我和小组成员抓紧时间学习文件操作和链表的有关知识，在此期间，我会经常遇到不理解的地方，每当此时，我都会向队友，学长或者老师请教，我相信自己的思考和老师的点拨可以让我们更快地学习到这些知识。在完成文件操作的过程中，我犯了一个错误：我花费了大量的时间研究往届学长的代码而疏于自己亲自尝试写代码，自以为学到了很多，实际上手操作时却发现因为缺乏经验而出现许多错误，并且还不清楚是为何而错，正因为这种错误的学习方法让我耽误了很多时间，最终在老师的点拨和我的尝试后，我重新完成了这方面的学习并吸取教训，以自己实践操作为主，参考学习为辅，只有多次试错、反思错误并改正错误才能取得真正的进步。

C语言课程设计给我带来的又一个收获是学会如何与队友配合，共同攻克难关。本次课程设计是一个团队任务，想要圆满完成绝不是一个人的努力可以做到的。然而，如何与队友紧密合作，共同克服困难本身也是一个挑战。在开始写代码前，我们就听说过存在往届学长学姐因为组内成员没有进行充分且有效的沟通导致代码无法合并的问题。为了避免这种情况的发生，我与队友频繁地交流代码进度并养成了每写一段代码就写上较为详细的注释的习惯。因为毕竟无法将代码内容事无巨细地全部讲给队友，只能通过较为详尽的注释让队友明白自己的思路和逻辑，我认为这对于共同推进课设进度是很有必要的。另外，我和队友几乎每周都会合2~3次代码，确保不会出现“在验收前夜合代码发现完全合不上”的情况。我认为，与队友配合同样重要的是任务规划，只有提前规划好任务分工和时间安排，才能高效、合理地完成课设任务。我和队友早早地做好了任务规划和时间安排，虽然中途存在因为任务难度较大而超出了一点点时间规划的情况，但这份规划的存在也对保持我们的完成节奏起到了正面作用。

总的来说，C课设在提高我的C语言水平的同时，也在培养我时间规划、团队合作、调整学习方法的能力。这个项目得以完成，也离不开队友之间的配合和帮助，离不开老师的点拨指导。为期半个学期的C语言程序设计确实让我受益匪浅。

# 第十部分 时间安排与代码分工

## 时间安排



## 代码分工

主要负责字符文字编辑功能模块，主要负责文件操作绘图模块。最后，两人一起编译、调试、Debug并撰写分析报告。

完成的文件是main.c、mainloop.c、insert.c、func.c、cursor.c、mainloop .h、insert .h、func.h、cursor.h，代码行数为1439行。

完成的文件是file.c、draw.c、seekbar.c、view.c、help.c、file.h、draw.h、seekbar.h、view.h、help.ch、common.h、init.h，代码行数为1183行。

引用的往届代码为mouse.c，hz.c，hz\_put.c；引用代码均未计入代码量