

# 浙江大学 实验报告

姓名：马璐瑶      学号：3190102205      日期：2020.04.23      成绩：\_\_\_\_\_

课程名称：C程序设计专题      指导老师：翁恺      实验名称：行编辑器

## 一、实验题目要求

实现一个GUI窗口内的单行的文本输入编辑器。

支持左右方向键、回退和删除键、ins切换插入和覆盖状态，编辑过程中有光标闪烁，插入和覆盖状态的光标形状不同。

回车后，结束输入，将输入的内容在标准输出输出。

## 二、实验思路 and 过程描述

为了实现行编辑器，首先我们需要在Setup函数中调用initWindow创建一个windows窗口，并注册键盘事件和计时器事件registerKeyboardEvent, registerTimerEvent，配合键盘、计时器消息接收函数keyboardListener和timerListener实现对键盘消息的处理以及光标的闪动，调用acllib中光标有关的函数来实现对光标的操作。

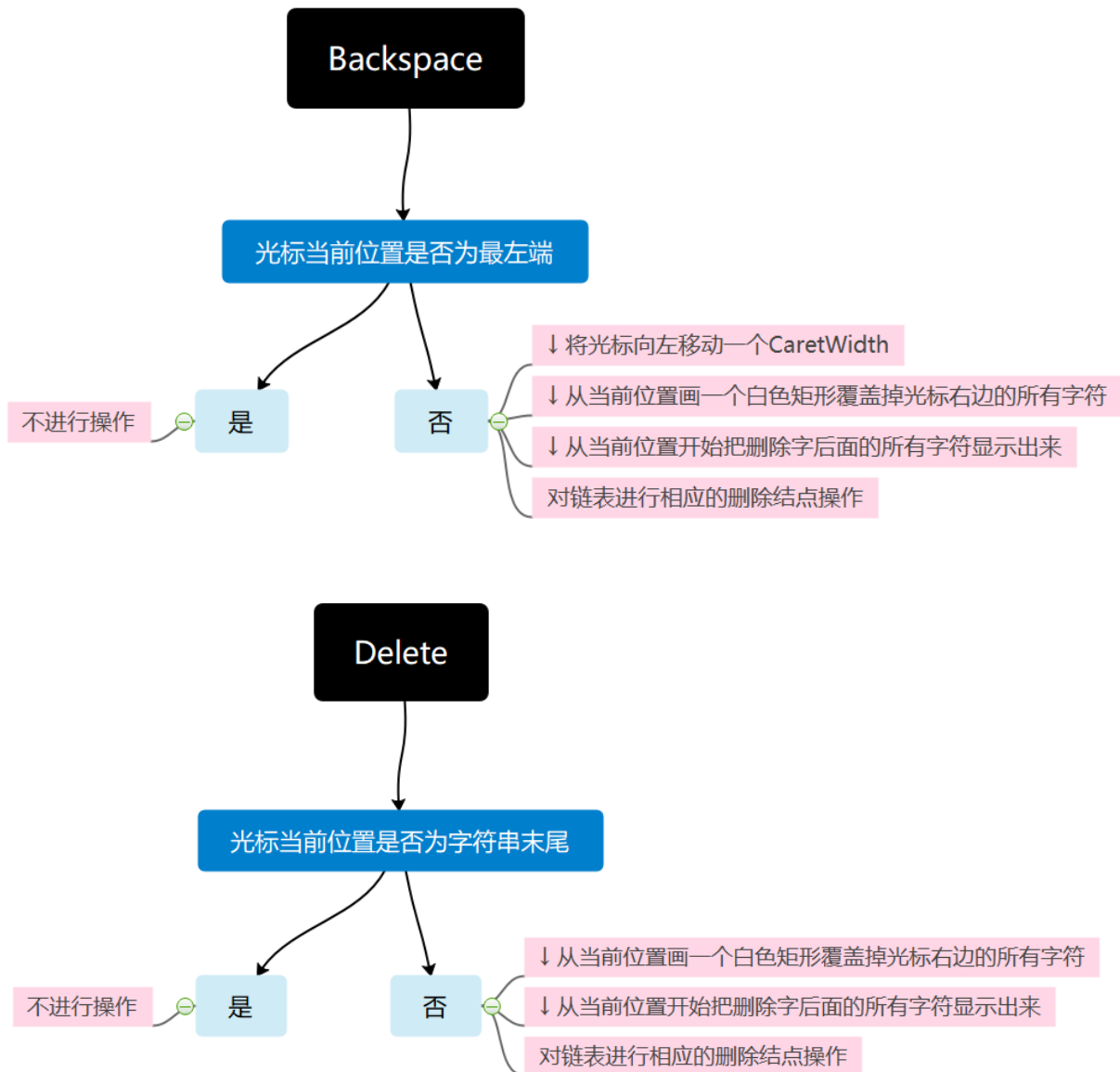
在本实验用到的 acllib.c和.h 中，由于光标(Caret)和文本(Text)是互相独立的，因此将光标和文本分开进行操作。至于我们在行编辑器中看到的光标和文本之间的左右移动、覆盖、退格和删除时的伴随移动，其实是人为添加的光标和文本间的关系，在本实验写出来的行编辑器中，光标和文本，都只是图像。

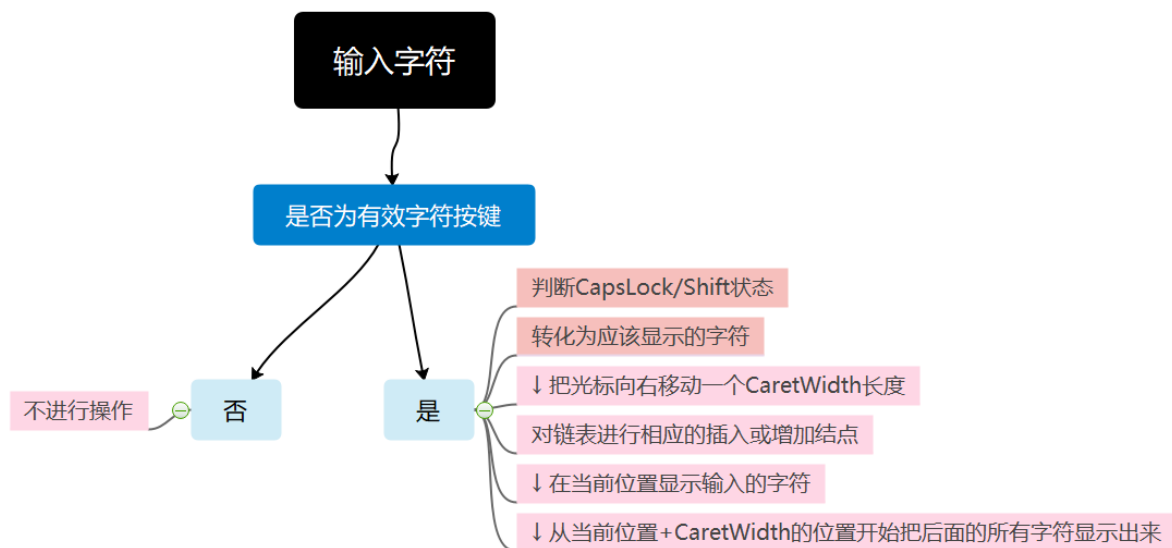
从不同按键的不同功能出发，考虑将整个程序分为以下几个部分：

- (1) 显示光标，呈现光标闪动效果；
- (2) 按下左右键时，光标左右移动；
- (3) 按下insert键时，切换插入和覆盖状态；
- (4) 按下Backspace时，实现“退格”和光标左移；
- (5) 按下Delete时，实现“删除”；
- (6) 按下普通字符键时，根据当前状态显示字符，并移动光标位置。

其中，为了实现退格、删除、插入字符的操作，我们需要将输入过的字符储存起来——不妨使用可增加、删除结点的链表来实现。需要注意的是，对显示的图像进行删减和插入时，也要对链表作相应的增加、删除结点操作，否则很容易出bug.

前三个部分的实现都比较简单，不需要考虑过多的情况，只是对光标的图像做一下简单变换。而后三个部分是本次实验的难点，因为其设计较多的操作情况，需要使用诸多判断语句。下面对Backspace, Delete和输入字符三种状态分别进行梳理：





### 三、实验代码解释

1. 首先定义了若干个宏，经过实验调整，窗口大小、文字的大小、光标的高度和宽度都比较合理。

```

1 #define TEXTSIZE 40
2 #define CARETWIDTH 25
3 #define CARETHEIGHT 40
4 #define HEIGHT 100
5 #define WIDTH 1000
6 #define MAX 1000
  
```

2. 使用链表来存放输入的字符，由于传入paintText的需要是字符串，因此把Node的Payload设置为char s[2], 方便字符的显示. 并且由于存放字符的链表list和指示当前光标位置下一个结点的指针ptr需要在多个函数中被调用，因此创建全局变量List\*list和Node\*ptr.

3. 本实验写了8个函数：

```

1 void AppendTail(List*list,char a);//当在字符串末尾输入时，在链表尾部添加结点
2 void InsertNode(List*list,char a);//当在字符串中间输入时，在链表中间插入结点
3 void PrintText(int x,int y,Node*ptr);//打印出从ptr开始到链表尾部的所有字符
4 int FollowingSize(Node*ptr);//返回当前光标后面所有字符的个数(即从ptr开始计算的链表长度)，便于确定画出的白色矩形的大小
5 int ischar(int key);//判断键入内容是否为有效字符
6 char convert(int key);//处理按住shift时同一按键的字符转化
7 void keyListener(int key,int event);//对键入信息进行综合处理，整个程序的灵魂
8 void timerListener(int id);//计时器，实现光标的闪动
  
```

其中AppendTail, InsertNode, PrintText, FollowingSize都是链表应用的基本函数, ischar和convert中通过调用switch-case来实现对多种情况的处理, 在此不做赘述。下面介绍 timerListener 和 keyListener.

#### 4. timerListener

```
1 void timerListener(int id)
2 {
3     static int count = 1;
4     beginPaint();
5     if(count%2)showCaret();
6     else hideCaret();
7     endPaint();
8     count++;
9 }
```

在Setup中创建了一个计时器 startTimer(0, 500); 即创建0号计时器, 每500ms启动一次, 用count记录启动次数, 在奇数次显示光标, 偶数次隐藏光标, 从而实现了光标的闪烁(每隔0.5s闪烁一次).

#### 5. keyboardListener

在该函数中, 首先定义了一些表示状态的变量

```
1 static int x=0,y=0;//当前光标位置
2 static int middle = 0;//判断光标是否在字符串中间
3 static int shift = 0;//判断是否按住shift
4 static int Caps = 32;//Capslock
5 static int Caret = 1;//Width of the Caret
6 int k = ischar(key);//(字母为1,数字和空格为3,符号为2)
7 char c[2];//当前字符
```

如果键入的是字符, 则对键入的key结合k, shift, Caps的值进行判断, 将key转化为实际显示的key, 赋值给表示当前的字符c.

```
1 c[0] = key;
2 c[1] = '\\0';
```

接着，判断当前光标位置是在字符串末尾还是中间，是插入还是覆盖状态，并对链表进行相应的添加尾巴或插入结点操作。

```
1  if(middle==1 && Caret==CARETWIDTH){//光标在中间，覆盖状态
2    // 1.在光标所在位置paintText(x,y,c);
3    // 2. 对链表作改动(ptr->s)[0] = c[0]; ptr=ptr->next;
4    // 3. 如果ptr改动后指向NULL，则middle=0，表示光标移到了末尾
5  else{
6    if(ptr==NULL);
7    else PrintText(x+CARETWIDTH,y,ptr);
8
9    if(middle==0)AppendTail(&list,c[0]);
10   else InsertNode(&list,c[0]);
11
12   beginPaint();
13   setTextSize(TEXTSIZE);
14   paintText(x,y,c);
15   endPaint();
16   if(middle==0 && ptr)ptr = ptr->next;
17 }
18 x+=CARETWIDTH;
19 setCaretPos(x,y);//x has changed.
```

如果按下的是shift,则改变shift状态，按下CapsLock则改变Caps状态，按下Insert则用setCaretSize改变光标大小，同时改变变量Caret的值。

按下左键时，如果光标已在最左端则不作操作，否则认为光标被移动到了中间，通过x-=CARETWIDTH实现向左移动光标，并调整ptr指向位置. 按下右键时，如果光标已在最右端则不做处理，其他操作与按左键时类似，如果ptr调整后为NULL则认为光标在末尾，middle=0.

```
1  else if(key==37 && event==0){//left
2    if(x==0);
3    else{
4      middle = 1;
5      x-=CARETWIDTH;
6      if(ptr==NULL)ptr = list.tail;
7      else ptr = ptr->prev;
8      setCaretPos(x,y);
9    }
10 }
```

重点处理按下Backspace和Delete的情况，按下Backspace时，如果光标在最左端则不做操作，否则判断当前ptr方向，并据此对链表结点进行调整，然后移动光标，画白色矩形覆盖后面字符，显示剩下的字符。

```
1  else if(key==8 && event==0 && x!=0){//Backspace
2      if(x==0);
3      else{
4          if(ptr==NULL){
5              //删除最后一个结点
6          }
7          else{
8              //删除掉中间结点
9          }
10         x-=CARETWIDTH;//光标左移
11         setCaretPos(x,y);
12         beginPaint();
13         setPenColor(WHITE);
14         setBrushColor(WHITE);
15         rectangle(x,0,x+CARETWIDTH*(FollowingSize(ptr)+1),CARETHEIGHT);
16         endPaint();
17         PrintText(x,y,ptr);//显示出剩下的字符
18     }
19 }
```

对Delete的操作也是类似的，先从光标位置画一个白色矩形覆盖后面字符，然后根据ptr是否为NULL来改动链表结点，在光标位置显示剩下的字符。

#### 四、实验体会和心得

通过本次实验，我自己写出了单行编辑器，它能够显示输入的大小写字符，各种符号，并且对按住的shift、按下的CapsLock都能做出相应调整。但我也发现它的很多不足：一是程序运行之初的大小写状态是无法识别的，默认输出的是小写字母；二是只能在单行编辑，Enter键和上下键是没有作用的；三是对Tab键没有反应（时间关系就没有加入Tab键了）。

实验中显示的字符其实都是一些图像，真正存放字符的是自己创建的链表，这就和电脑上的记事本有很大不同，记事本创建的是txt文件，存放内容的是字符串，能够在程序中调用fscanf来读取txt文件的内容，而这次Project中写的GUI程序显然不具备这种特点——它就是个图形程序嘛。

如果要在本次Project的基础上再加以改进，我想可以

①加入多行编辑功能：仍然用一个链表储存字符，只是在PrintText的时候，加入输出到行末自动换行的功能。按下Enter的时候也进行换行操作。

②加入复制粘贴功能，思路是再创建两个Node指针A,B，两指针中间的内容是复制或者粘贴的内容。按住Ctrl和左右键的时候改变光标形态(此时光标应停止闪动)，按下C的时候保存字符到A,B之间，Ctrl+V的时候把粘贴内容显示到当前光标后面，同时对list进行调整。