

浙江大学实验报告

专业：计算机科学与技术

姓名：唐皓翔

学号：3190104797

日期：2020.4.20

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 实验类型： 多文件编程

一、实验目的和要求

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

将功能划分为键盘和字符两个回调函数，定义一个全局字符数组来储存输入的内容，并定义一个字符指针指向当前编译位置。键盘回调函数主要用于判断删除、方向键等行为并移动字符指针；字符回调函数用于读入字符；

三、实验代码分析

1. function. h:

标准头文件，包含三个 callback 函数如下

```
void KeyListener(int key,int event);  
void CharListener(char c);  
void Mouselistener(int x,int y,int button,int event);
```

分别读取键盘、字符、鼠标的反馈。

2. funtion. c:

首先定义全局字符串 str、全局字符指针 p、全局的 x 坐标 x；

➤ **void KeyListener(int key,int event)**

①**回车输出**：回车键码为 13，用 if 语句判断 key 和 event 的值。如果判断为是，打印字符串 str，初始化光标位置，用 20*600 的白色长方形覆盖图形界面上输入的内容，最后用 memset 函数将 str 置零，p 指向 str[0]。

②**ins 键切换**：首先在函数外定义一个整型变量 ins，初值为零。当 key 等于 ins 键码时，通过 ins=! ins，改变 ins 是非性。若 ins 为非，则将光标宽度设为 2，否则设为 5。

③**方向键**：以左方向键为例，当 key=37 时，若*(p-1)不为 0，则 x 坐标减 11，p 指针指向前一个元素，并重新设置光标位置为 x。

④**回退键**：回退键 key=8，同样 if 判断。如果判断为是，定义一个字符指针 p1，用 while 循环从 p-1 处开始，把后面的字符整体向前移一格，用白色长方形遮盖 x-11(字符宽度)到右边界，并重新在界面上打印 str+(x/11-1)的字符串，代码如下：

```

if(key==8&&event==1){
    p1=p-1;
    while(*p1!=0){
        *p1=*(p1+1);
        p1++;
    }
    beginPaint();
    rectangle(x-11,0,600,20);
    setCaretPos(x-11,0);
    showCaret();
    paintText(x-11,0,str+(x/11-1));
    endPaint();
    x-=11;
    p-=1;
}

```

⑤删除键 Del：功能的实现和回退键基本相同，不再赘述。

➤ void CharListener(char c)

分为两种情况：

①插入：即 ins==0 时。若此时 p 指向字符串结尾，直接在 x 位置打印读入的字符，然后 p++，x+=11。若 p 指向字符串中间，和回退键处理类似，将 p-1 后面的字符都往后移一格，将读入的字符插入 p 位置，然后用白色长方形覆盖 x 后面的字符，并在 x 位置打印 p 开始的字符串，最后 p++，x+=11。代码如下：

```

if(*(p+1)==0){
    ch[0]=*p=c;
    beginPaint();
    paintText(x,0,ch);
    setCaretPos(x+11,0);
    showCaret();
    endPaint();
    x+=11;
    p++;
}else{
    while((*p0)!=0){
        p0++;
    }
    while(p0!=p){
        *p0=*(p0-1);
        p0--;
    }
    ch[0]=*p=c;
    beginPaint();
    rectangle(x,0,600,20);
    paintText(x,0,str+(x/11));
    setCaretPos(x+11,0);
    showCaret();
    endPaint();
    x+=11;
    p++;
}

```

②覆盖：将 p 位置元素替换成读入的字符，并如上用覆盖白色长方形并重新打印，不再详细说明。

➤ void Mousetlistener(int a,int y,int button,int event)

该函数的功能是通过鼠标的点击事件来改变编辑位置。

首先算出字符 str 所占的长度 x0。当鼠标点击位置的 y 坐标小于 25 时，若 x 坐标 a 大于 x0，则光标移至最末尾；若 a 小于 x0，计算 a 在哪一个字符上，并将光标移至该字符的前面，具体代码如下：

```
int len=0,x0=0;char *p2=str;
if(y<24&&event==0){
    while(*p2!=0){
        len++;
        p2++;
    }
    x0=3+len*11;
    if(a<=x0){
        x=((a-3)/11)*11+3;
        p=str+(x/11);
        setCaretPos(x,0);
        showCaret();
    }else{
        x=x0;
        p=str+(x/11);
        setCaretPos(x,0);
        showCaret();
    }
}
```

3. main. c:

创建图形窗口和控制台

```
initWindow("行编辑器",0,0,600,600);
initConsole();
```

将文本颜色设为黑色，画笔刷子设为白色

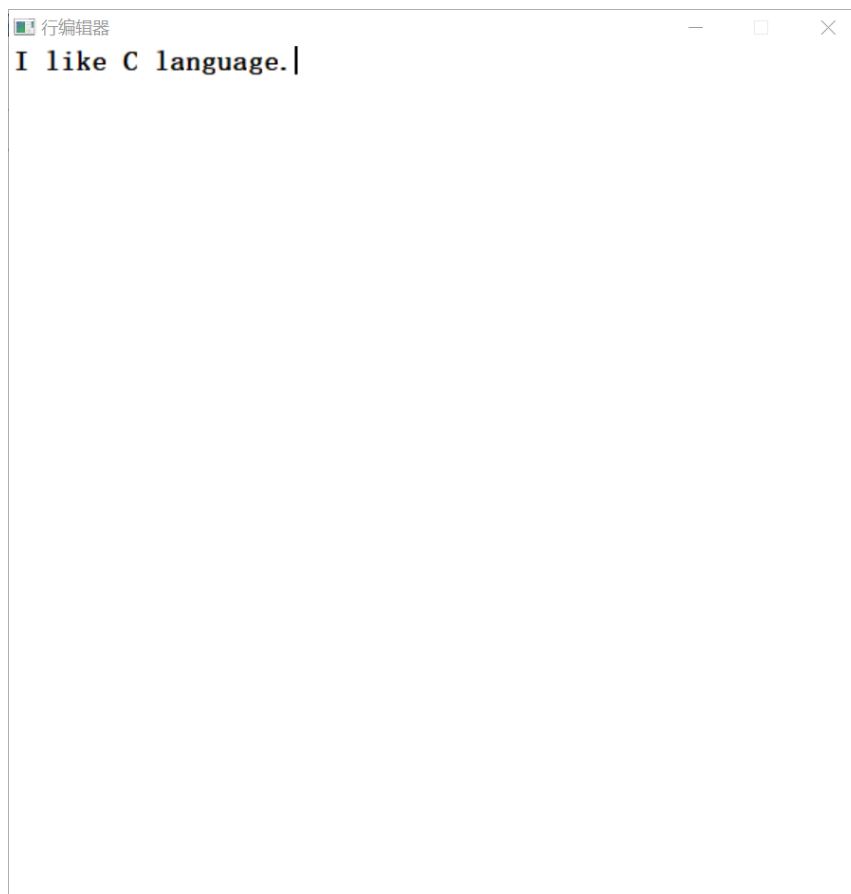
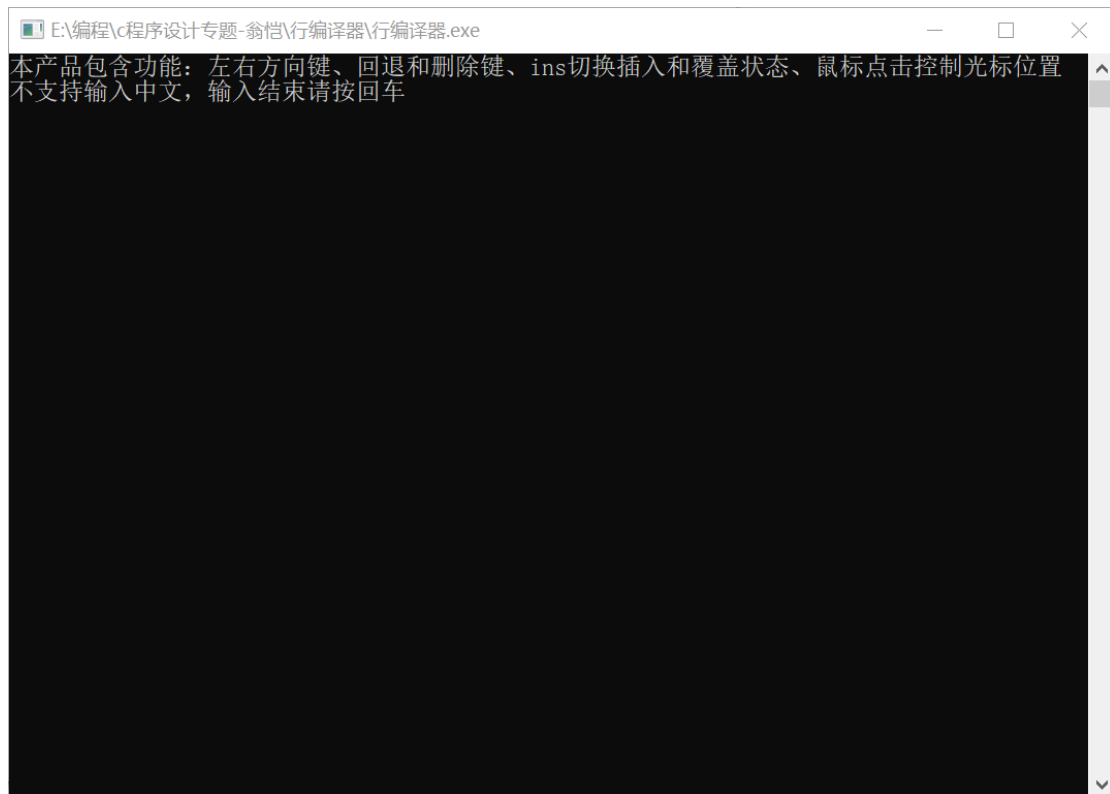
初始化光标位置、大小

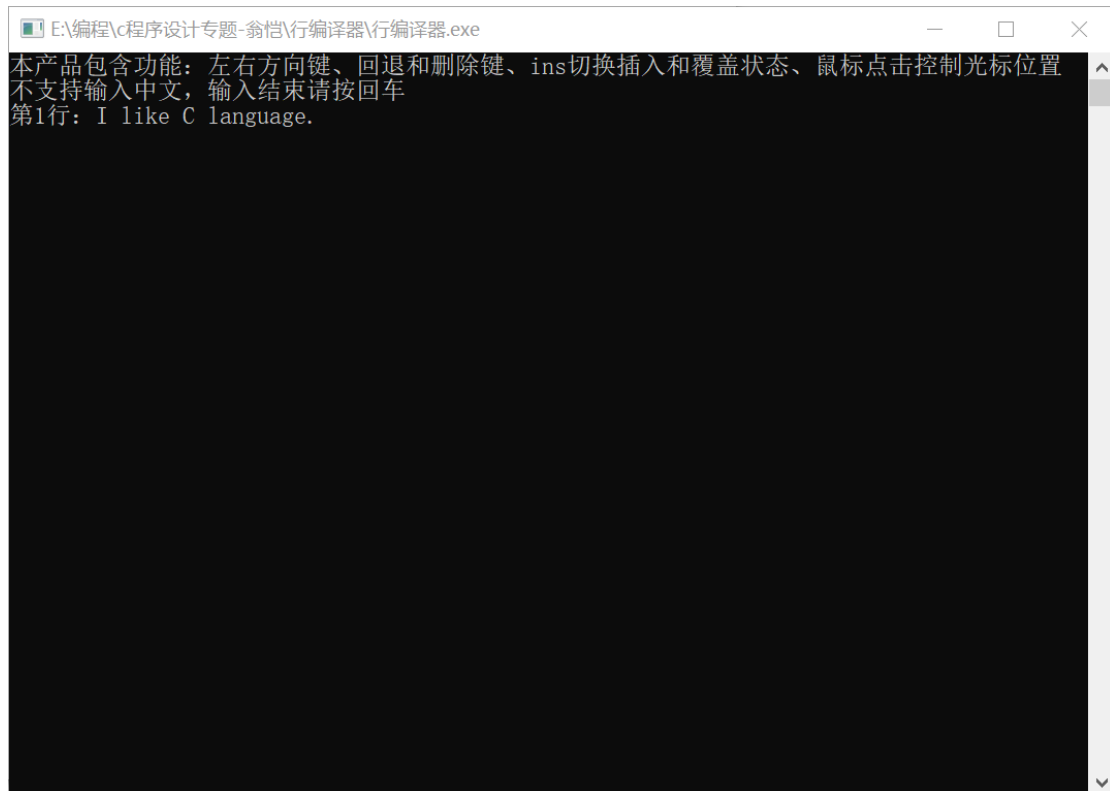
```
setTextSize(20);
setCaretSize(2,20);
setCaretPos(0,0);
showCaret();
```

三个事件回调函数

```
registerCharEvent(CharListener);
registerKeyboardEvent(KeyListener);
registerMouseEvent(Mousetlistener);
```

四、成果展示





五、实验心得

这次实验主要用到的是 `acllib` 库中的事件回调函数。其实理解了回调函数的运行规则之后，代码写起来并不是很难。

回退、方向键这些功能本质上差不多，都是接收一个来自键盘或者鼠标的信号，然后进行 `if` 的判断，最后做出相应的动作。在写完题目要求的功能之后，我想到了平时我们用编辑器时，可以用鼠标调整光标位置。这个功能很实用而且代码写起来也挺简单的，所以就花了一点时间加上了这个功能。

这次实验最大的感触是让我感觉到自己写的程序有实用价值。这也让我对后面的学习产生了更大的兴趣。