行编辑器实验报告

一、 编译运行方法

在本压缩包的 dev 和 ACLlib/sample/cpp 中有可用 dev-cpp 和 VS 运行的两个版本。运行后，可在图形界面进行输入。

二、 程序功能

本程序可在图形窗口实现一个行编辑器，显示输入的可显示字符。本程序支持的编辑动作及输入字符如下：

支持的编辑动作：

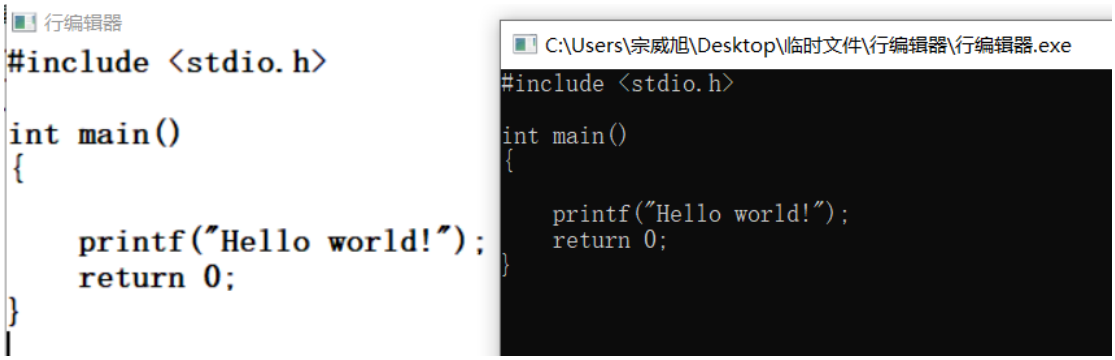
1) ←/→：移动光标位置；
2) Backspace/Delete：删除光标前/后的字符；
3) Enter：换行，并在 CLI 显示输入内容；
4) Esc：退出程序；
5) Caps Lock：大小写切换；
6) Shift：输入状态切换；
7) Tab：制表符；
8) Home：将光标移动到行首；
9) End：将光标移动到行尾；
10) F1~F4：更换该行所有显示字符颜色；
11) F5~F8：更换该行所有显示字符背景颜色；

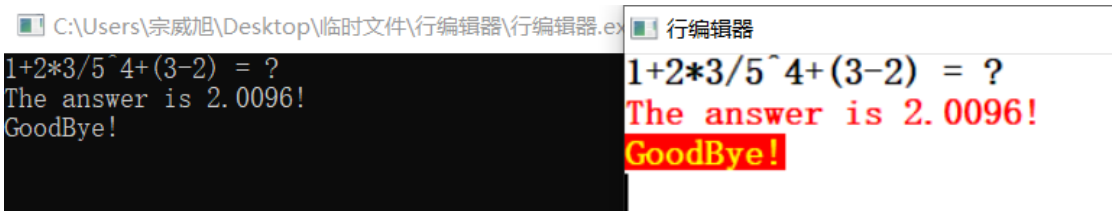
支持的输入字符：

数字，字母（大小写）、标点、空白符、符号。注意：输入字符不支持小键盘输入、中文和中文符号。

程序存在一些不能支持的功能。比如程序不支持自动换行，需要用户自行换行。每行的最大字符容量为 110。程序的最大字符处理量为约 106 个。

运行中的样例如下：





三、 心得体会

第一次完成代码最大的障碍是如何实现光标移动后的删除。通过数组来实现。完成后突然想到，其实如果利用链表实现存储字符的话，由于链表的插入和删除十分方便，具有优越性。而且，通过链表读入可以实现换行、返回上一行、将字体颜色等字体信息与字体绑定等功能。然而……由于时间问题，只能先提交数组版本，希望在假期有时间，把链表版本做出来。

ACLlib 库还算比较对我这种萌新友善的图形库，但是还有一个 bug 想不出来，就是最初我在实现 backspace 和 delete 删除时，选择将数组结尾的字符变为'\0'，然而那个字符还是会被刷出来。后来发现需要换成空白符才能实现这个功能，并不明白为什么会这样。猜想可能是输入文本传入的字符串没有被改变，或者有某种机制使其不能被改为'\0'。

本次实现符号的读入的方法是 switch case，虽然实现了，但感觉方法太低端了，在讨论课要向大家学习。我也会把我的大作业推到 gitHub 上，以便后续的改进工作。

四、 源代码

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "acllib.h"
```

```
#define length 1210
#define width 800
#define MAX 1000000
#define deltax 11
#define deltay 22

#define caretwidth 2
#define caretheight 22

#define textsize 20

#define backspace 8
#define table 9
#define enter 13
#define shift 16
#define capslock 20
#define esc 27
#define space 32
#define end 35
#define home 36
#define leftarrow 37
#define rightarrow 39
```

```
#define Delete 46

void initCaret();
void MoveCaret(int dx);
void scanText(int key, int event);
void Print(int key,int event);
void Capslock(int key,int event);
void Backspace(int key,int event);
void DeLeTe(int key,int event);
void LeftArrow(int key,int event);
void RightArrow(int key,int event);
void Table(int key,int event);
void Home(int key,int event);
void End(int key,int event);
void Shift(int key,int event);
void TextConfig(int key,int event);
void Exit(int key,int event);
void insertText(int key,int event);
void deFault(int key,int event);

int p = 0,len = 0;
int x = 0, y = 0, dx = 0;
```

```
int Caps = 1,SHIFT = 1;
int textcolor = BLACK,textbkcolor = WHITE;
char ch[MAX] = { 0 };

int Setup()
{
    initConsole();
    initWindow("zwx", DEFAULT, DEFAULT, length, width);
    initCaret();
    registerKeyboardEvent(scanText);
    return 0;
}
void initCaret()
{
    setCaretSize(caretwidth, caretheight);
    showCaret();
    setCaretPos(x, y);
}
void MoveCaret(int dx)
{
    setCaretSize(caretwidth, caretheight);
    showCaret();
```

```
        setCaretPos(x + dx, y);
        dx = 0;
    }
    void scanText(int key, int event)
    {
        int j = 0;
        void(*point[20])(int key,int event) =
        {

    Print,Capslock,Backspace,DeLeTe,LeftArrow,RightArrow,Table,
Home,End,Shift,TextConfig,Exit,insertText,deFault
        };
        if (!event)
        {
            if (key == enter) j = 0;
            else if (key == capslock) j = 1;
            else if (key == backspace) j = 2;
            else if (key == Delete) j = 3;
            else if (key == leftarrow) j = 4;
            else if (key == rightarrow) j = 5;
            else if (key == table) j = 6;
            else if (key == home) j = 7;
```

```
        else if (key == end) j = 8;
        else if (key == shift) j = 9;
        else if (key >= 112 && key <= 123) j = 10;
        else if (key == esc) j = 11;
        else if (key == space) j = 12;
        else j = 12;
    }
    else
    {
        if (key == shift) j = 9;
        else j = 13;
    }
    (*point[j])(key,event);
}
void showText()
{
    beginPaint();
    setTextSize(textsize);
    setTextColor(textcolor);
    setTextBkColor(textbkcolor);
    paintText(x, y, ch);
    endPaint();
```

```
}
void Print(int key,int event)
{
    int j = 0;
    for (j = 0;j < len;j++) printf("%c", ch[j]);
    printf("\n");
    for (j = 0;j < len;j++) ch[j] = 0;
    p = 0;dx = 0;len = 0;y += deltay;
    initCaret();
}
void Capslock(int key,int event)
{
    Caps *= -1;
}
void Backspace(int key,int event)
{
    int j = 0;
    if (p != 0)
    {
        dx -= deltax;p--;len--;
        for (j = p;j < len;j++) ch[j] = ch[j + 1];
        ch[len] = ' ';
```

```
        showText();
        MoveCaret(dx);
    }
}
void DeLeTe(int key,int event)
{
    int j = 0;
    if (p != len)
    {
        len--;
        for (j = p;j < len;j++) ch[j] = ch[j + 1];
        ch[len] = ' ';
        showText();
        MoveCaret(dx);
    }
}
void LeftArrow(int key,int event)
{
    if (p)
    {
        dx -= deltax;
        p--;
```

```
        MoveCaret(dx);
    }


}
void RightArrow(int key,int event)
{
    if (p != len)
    {
        dx += deltax;
        p++;
        MoveCaret(dx);
    }
}
void Table(int key,int event)
{
    int m = 0, n = 0;
    for (m = 0;m < 4;m++)
    {
        len++;dx += deltax;
        for (n = len - 1;n > p;n--)
        {
            ch[n] = ch[n - 1];
```

```
        }
        ch[n] = 32;
        p++;
    }
    showText();
    MoveCaret(dx);
}
void Home(int key,int event)
{
    dx -= p * deltax;
    p = 0;
    MoveCaret(dx);
}
void End(int key,int event)
{
    dx += (len - p)*deltax;
    p = len;
    MoveCaret(dx);
}
void Shift(int key,int event)
{
    if(!event) SHIFT = -1;
```

```
    else SHIFT = 1;
}
void TextConfig(int key,int event)
{
    switch(key)
    {
        case 112: textcolor = BLACK; break;
        case 113: textcolor = RED; break;
        case 114: textcolor = YELLOW; break;
        case 115: textcolor = CYAN; break;
        case 116: textbkcolor = WHITE; break;
        case 117: textbkcolor = RED; break;
        case 118: textbkcolor = YELLOW; break;
        case 119: textbkcolor = CYAN; break;
    }
    showText();
}
void Exit(int key,int event)
{
    exit(0);
}
void insertText(int key,int event)
```

```
{
    char temp;
    int j = 0;
    if (key == space) temp = key;
    else if (key >= 48 && key <= 57)
    {
        switch(key)
        {
            case 49:
            if(SHIFT == 1) temp = '1';
            else if(SHIFT == -1) temp = '!';
            break;
            case 50:
            if(SHIFT == 1) temp = '2';
            else if(SHIFT == -1) temp = '@';
            break;
            case 51:
            if(SHIFT == 1) temp = '3';
            else if(SHIFT == -1) temp = '#';
            break;
            case 52:
            if(SHIFT == 1) temp = '4';
```

```
else if(SHIFT == -1) temp = '$';
break;
case 53:
if(SHIFT == 1) temp = '5';
else if(SHIFT == -1) temp = '%';
break;
case 54:
if(SHIFT == 1) temp = '6';
else if(SHIFT == -1) temp = '^';
break;
case 55:
if(SHIFT == 1) temp = '7';
else if(SHIFT == -1) temp = '&';
break;
case 56:
if(SHIFT == 1) temp = '8';
else if(SHIFT == -1) temp = '*';
break;
case 57:
if(SHIFT == 1) temp = '9';
else if(SHIFT == -1) temp = '(';
break;
```

```
            case 48:
            if(SHIFT == 1) temp = '0';
            else if(SHIFT == -1) temp = ')';
            break;
        }
    }
    else if (key >= 65 && key <= 90)
    {
        if(SHIFT == 1)
        {
            if(Caps == -1) temp = key;
            else if (Caps == 1) temp = key - 'A' + 'a';
        }
        else if(SHIFT == -1)
        {
            if(Caps == -1) temp = key - 'A' + 'a';
            else if (Caps == 1) temp = key;
        }
    }
    else if (key >= 186 && key <= 192)
    {
        switch(key)
```

```
{
    case 186:
    if(SHIFT == 1) temp = ';';
    else if(SHIFT == -1) temp = ':';
    break;
    case 187:
    if(SHIFT == 1) temp = '=';
    else if(SHIFT == -1) temp = '+';
    break;
    case 188:
    if(SHIFT == 1) temp = ',';
    else if(SHIFT == -1) temp = '<';
    break;
    case 189:
    if(SHIFT == 1) temp = '-';
    else if(SHIFT == -1) temp = '_';
    break;
    case 190:
    if(SHIFT == 1) temp = '.';
    else if(SHIFT == -1) temp = '>';
    break;
    case 191:
```

```
            if(SHIFT == 1) temp = '/';
            else if(SHIFT == -1) temp = '?';
            break;
            case 192:
            if(SHIFT == 1) temp = '`';
            else if(SHIFT == -1) temp = '~';
            break;
        }
    }
    else if (key >= 219 && key <= 222)
    {
        switch(key)
        {
            case 219:
            if(SHIFT == 1) temp = '[';
            else if(SHIFT == -1) temp = '{';
            break;
            case 220:
            if(SHIFT == 1) temp = '\\';
            else if(SHIFT == -1) temp = '|';
            break;
            case 221:
```

```
                if(SHIFT == 1) temp = ']';
                else if(SHIFT == -1) temp = '}';
                break;
                case 222:
                if(SHIFT == 1) temp = '\'';
                else if(SHIFT == -1) temp = '"';
                break;
            }
        }
        else return;
        dx += deltax;
        len++;
        for (j = len - 1;j > p;j--)
        {
            ch[j] = ch[j - 1];
        }
        ch[j] = temp;
        p++;
        showText();
        MoveCaret(dx);
    }
    void deFault(int key,int event) {
```

}