SHANGHAI UNIVERSITY

2021-2022 学年秋季学期

上海大学　计算机学院

《汇编语言程序设计》

实验 5

实验名称 ： 屏幕窗口程序实验

专业 ： 19 级直招计科 2 班

姓名 ： 汪雨卿

学号 ： 19120191

实验名称：屏幕窗口程序实验

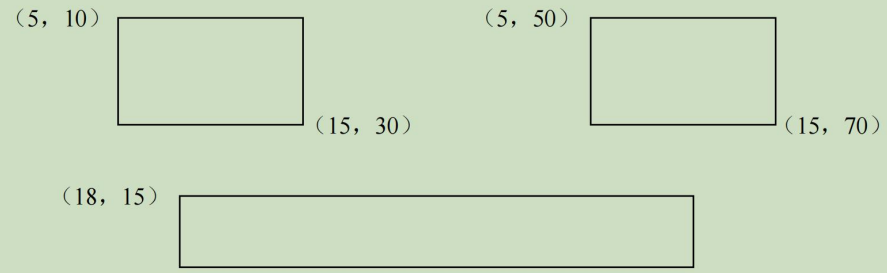
一、实验目的

（1） 掌握输入输出程序设计的概念和方法。
（2） 学习如何在 PC 机上编写具有输入输出功能的程序。
（3） 了解 DOS 和 BIOS 功能调用

二、实验内容

3. 实验内容：

在屏幕上开出三个窗口，它们的行列坐标如下图所示。



光标首先定位在右窗口最下面一行的行首（15，50），如从键盘输入字符，则显示在右窗口，同时也显示在下窗口的最下面一行。若需要将字符显示于左窗口，则先按下←键，接着再从键盘输入字符，字符就会从左窗口的最下行开始显示，同时下窗口也显示出左窗口的内容。如若再按下→键，输入字符就会接在先前输入的字符之后显示出来。当一行字符显示满后（左右窗口一行显示 20 个字符，下窗口一行显示 50 个字符），窗口自动向上卷动一行，输入字符继续显示于最低一行，窗口最高一行向上卷动后消失。

(1) 实验原理

①宏指令

宏意思是源程序中的一段有独立功能的程序代码宏指令，用户自定义的指令。编程序时候，将多次使用的功能用一条宏指令来代替。

基本格式：

```
宏名称    MACRO    参数 1，参数 2，.....
          local    标号 1，标号 2 .....
          标号 1：  xxxxx
                   xxxxx
                   xxxxx
          ENDM
```

宏的使用：

```
M1    MOV, 5, 1
```

这条指令会被展开如下结果大致上就是字符替换

```
??0001:  MOV  AX, BX
         MOV  AX, 5
         MOV  BX, 1
         TEST  AX, BX
         JNZ  ??0001
         JMP  ??0002
??0002:  MOV  AX, BX
```

其中值得注意的是标号被替换成了??0001 ??0002 这是因为宏如果在同一段代码里多次展开，标号也会跟着被多次展开，那么就会出现一段代码里同一个标号被多次定义，就会报错，所以如果在宏定义里面有标号定义，那么在开头要使用 LOCAL 指令声明本地标号，这样汇编器会自动累加数字来代替标号

另外，使用的时候参数多传少传都可以，多传了就舍弃，少传了就默认用空代替，之后介绍的命令 IFB 可以判断参数是否为空

②BIOS 功能调用

BIOS（Basic input/output system）固化在 ROM 中，包括 I/O 设备的处理程序和许多常用的例行程序。

对用户程序来说，可由特定指令 INT n（n 为中断号）通过软终端的方式调用。不管 DOS 是否装入系统，这些调用可以直接控制 I/O 设备。

主要的 BIOS 功能调用如下：

中断号	10	11	12	13	14	15	16	17
功能	视频服务	设备类型	内存容量	磁盘	I/O 串行口	磁带	I/O 键盘	打印机

举例：INT 1OH ;视频服务 BIOS 功能调用

⑵ 实验步骤

（1）启动 MASM 6.0 或 MASM for Windows 集成编程环境。

（2）分支指令形式编写.ASM 源程序。

（3）对其进行汇编及连接，产生.EXE 文件。

（4）作必要的调试。

⑶ 实验记录

a） 数据段部分：设置三个框和光标位置的变量，同时设置窗口位置状态值，

用于后序的窗口切换。

```
001 DATAS SEGMENT
002     ;光标的初始位置
003     inity   db  15
004     initx   db  50
005
006     ;记录左窗口当前的光标位置，初始化为做窗口的初始位置
007     leftx   db  10
008     lefty   db  15
009
010     ;记录右窗口当前的光标位置，初始化为做窗口的初始位置
011     rightx  db  50
012     righty  db  15
013
014     ;记录下窗口当前的光标位置，初始化为下窗口初始位置
015     nowx    db   15
016     nowy    db   22
017
018     ulrow   db  0
019     ulcow   db  0
020     lrrow   db  0
021     lrcol   db  0
022     windowstat  db   1      ;=1，右边窗口，=2，左边窗口
023 DATAS ENDS
024
```

b） 宏指令部分：编写一些可复用的程序段。

i. 调用 BIOS 的 06 号功能，实现指定位置的窗口清屏。

指令格式：clear x, y, m, n

- x, y 依次是左上角的行号，列号
- m, n 依次是右下角的行号，列号

```
025 ;宏指令clear清屏（左上行，左上列，右下行，右下列）
026 clear  macro a,b,c,d
027          mov  al,0  ;整个窗口空白（设置上卷行数）
028          mov  bh,7  ;卷入行属性为07，是正常属性（设置卷入行属性）
029          mov  ch,a  ;左上角行号
030          mov  cl,b  ;左上角列号
031          mov  dh,c  ;右下角行号
032          mov  dl,d  ;右下角列号
033          mov  ah,6  ;设置BIOS的功能号
034          int 10h    ;显示器驱动程序
035 endm
036
```

ii. 调用同 i，实现页面的上卷操作

指令格式：scroll num, x, y, m, n

参数：

- num 上卷行数
- x, y 依次是左上角的行号，列号
- m, n 依次是右下角的行号，列号

```
037 ;-------------------------------------------------------------宏指令 s
038 scroll macro cont,ulrow,ulcol,lrrow,lrcol
039          mov al,cont                    ;上卷行数
040          mov bh,70h                     ;卷入行属性
041          mov ch,ulrow                   ;左上角行号
042          mov cl,ulcol                   ;左上角列号
043          mov dh,lrrow                   ;右下角行号
044          mov dl,lrcol                   ;右下角列号
045          mov ah,6
046          int 10h                        ;BIOS调用，上卷
047 endm
```

iii. 调用 BIOS 的 2 号功能，设置当前光标的位置

指令格式： locate_n y, x

参数：

- y, x 依次是左上角的行号，列号

```
;-------------------------------------------------------------宏指令locate_n 光标定位
locate_n macro y,x   ;设置光标在 (y, x)位置，设置光标行列位置
                     ;注意这里坐标是（行号，列号）的形式，行号是y，列号是x）
    mov bh,0         ;显示页号
    mov ah,2
    mov dh,y         ;int10h中，ah=2时，设置光标位置，dh为行号，dl为列号
    mov dl,x
    int 10h
    endm
```

iv. SHOW 显示功能：调用 BIOS 的字符显示功能。先将读入的字符输出到当前光标位置；然后调用 locate_n 宏指令，将该字符继续输出到新定位后的光标位置。

指令格式： show

```
059 ;-------------------------------------------------------------在当前窗口和下窗口显示
060 show macro
061     mov bh,0              ;显示页
062     mov cx,1              ;cx = 字符重复个数，即重复一次
063     mov ah,0ah            ;在当前的窗口的光标位置处打印字符和属性
064     int 10h
065
066     locate_n nowy,nowx    ;重新设置当前光标位置到下窗口的正确位置处
067     mov bh,0
068     mov cx,1
069     mov ah,0ah            ;在下窗口输出
070     int 10h
071 endm
```

v. GETCHAR 窗口切换功能：通过对读入字符进行判断，利用子程序跳转实现不通过窗口的显示切换，以及退出程序功能。

指令格式： getchar

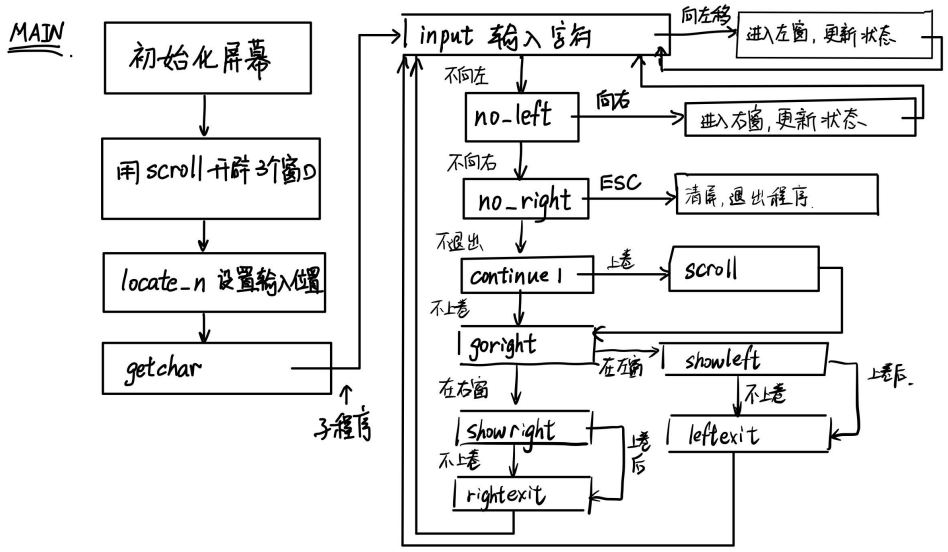
```
;-------------------------------------------------------------------接收字符并判断
getchar macro                   ;一个一个读入，判断，输出
   input:                       ;输入键盘上的字符
       mov ah,0
       int 16h                  ;BIOS调用，从键盘读字符

       cmp ah,4bh               ;判断输入字符是否为左向键扫描码
       jnz no_left              ;输入为左向键时，继续执行；否则跳转no_left
       locate_n lefty,leftx     ;向左：光标重新定位到左窗口
       mov windowstat,2         ;更新窗口状态
       jmp input                ;继续输入

   no_left:                     ;不跳转：已跳转至左窗口/保持在右窗口
       cmp ah,4dh               ;判断输入字符是否为右向键
       jnz no_right             ;输入为右向键时，继续执行；否则跳转no_right
       locate_n righty,rightx   ;向右：光标重新定位到有窗口
       mov windowstat,1         ;更新窗口状态
       jmp input                ;继续输入

   no_right:
       cmp ah,01h               ;判断输入字符是否为ESC
       jnz  continue            ;没有结束，继续执行
       clear 0,0,24,79          ;屏幕总大小：25×80
       ret
```

```
098     continue:
099        show                    ;不是左右键和ESC,则嵌套调用宏show（显示输入的可打印的字符）
100        inc nowx                ;先显示字符，再将下窗口该行的光标的列号加1，再来比较是否要上卷
101
102        cmp nowx,65             ;判断下窗口是否上卷
103        jle goright             ;<=65：isright表示下窗口中当前光标行可以向右走
104        scroll 1,18,15,22,65    ;>65：  到了一行的最右端，要向上卷一行
105        mov nowx,15             ;置位：上卷一行后，当前列号变为当前窗口最左边的列的列号
106
107    goright:                    ;判断左窗口/右窗口的右侧输入
108        cmp windowstat,1        ;判断是否在右窗口
109        jnz showleft            ;在右窗口：showright;否则，showleft
110
111    showright:                  ;判断右窗口是否上卷
112        inc rightx              ;列数+1
113        cmp rightx,70
114        jle rightexit           ;<= 70：rightexit表示右窗口中当前光标行可以向右走，不用上卷
115        scroll 1,5,50,15,70     ;> 70 ：上卷一行
116        mov rightx,50               ;置位，恢复列号
117
118    rightexit:                  ;右窗口输出
119       locate_n righty,rightx       ;定位到右窗口的光标位置
120       jmp input                ;继续输入
```

```
122    showleft:                   ;判断左窗口是否上卷  （同right的操作）
123       inc leftx
124       cmp leftx,30
125       jle leftexit
126       scroll 1,5,10,15,30
127       mov leftx,10
128
129    leftexit:                   ;左窗口输出
130       locate_n lefty,leftx
131       jmp input                ;继续输入
132
133       endm        ;getchar macro endm
```

c） 编写主入口程序，利用宏指令实现程序要求功能。

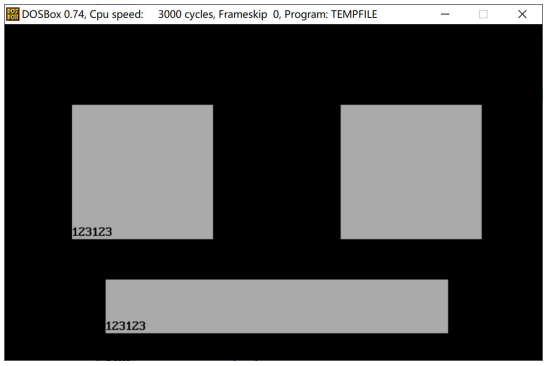
（4）主程序和子程序框图



(5) 数据处理

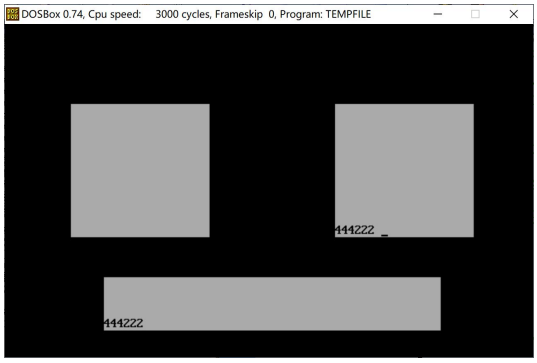
① 输入：←123123

结果：



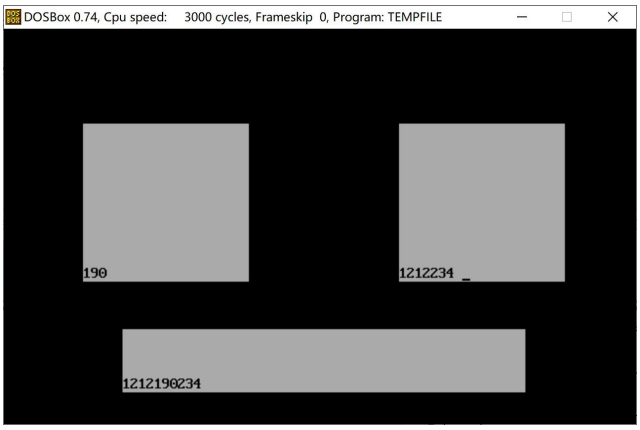
② 输入：→444222

结果：



③ 输入：1212←190→234

结果：



三、实验体会

本次实现主要学习了如何在汇编语言中调用 BIOS 功能实现输入输出的可视化窗口。此外，在编写代码的过程中也尝试引入了宏指令的编程方式，利用宏指令的方式增强代码的复用性，简化主程序的长度，使得整个程序可维护性和可阅读性得到了显著的提升。

在学习 BIOS 功能调用的同时，我由 BIOS 功能和 DOS 功能调用的区别产生好奇，简单了解了两个调用的区别。BIOS 功能会直接和一些 IO 设备进行调用，通过软终端实现。更偏向于硬件层面的调用。而 DOS 的调用，则是基于 BIOS 之上。因此，更好的掌握和理解 BIOS 的使用，会极大程度提升程序员使用 IO 设备的灵活性，并且基于用户更好的感官体验。

最后，关于宏指令部分的学习，让我更加重视编写代码过程中的代码复用。一个更为灵活，可维护的代码，它的要求并不能仅仅局限于一份代码是否能够被正常运行，且实现要求的功能；更重要的是这份代码可以被易于的理解，以及减少不必要的冗余，同时能够长期的维护。

这次实验也让我学习到了很多关于 BIOS 的调用方式，让我受益匪浅。

附上代码如下：

```
001 DATAS SEGMENT
002     ;光标的初始位置
003     inity   db  15
004     initx   db  50
005
006     ;记录左窗口当前的光标位置，初始化为做窗口的初始位置
007     leftx   db  10
008     lefty   db  15
009
010     ;记录右窗口当前的光标位置，初始化为做窗口的初始位置
011     rightx  db  50
012     righty  db  15
013
014     ;记录下窗口当前的光标位置，初始化为下窗口初始位置
015     nowx    db   15
016     nowy    db   22
017
018     ulrow   db  0
019     ulcow   db  0
020     lrrow   db  0
021     lrcol   db  0
022     windowstat  db   1      ;=1，右边窗口，=2，左边窗口
023 DATAS ENDS
024
025 ;宏指令clear清屏（左上行，左上列，右下行，右下列）
026 clear  macro a,b,c,d
027         mov  al,0  ;整个窗口空白（设置上卷行数）
028         mov  bh,7  ;卷入行属性为07，是正常属性（设置卷入行属性）
029         mov  ch,a  ;左上角行号
030         mov  cl,b  ;左上角列号
031         mov  dh,c  ;右下角行号
032         mov  dl,d  ;右下角列号
033         mov  ah,6  ;设置BIOS的功能号
034         int 10h    ;显示器驱动程序
035 endm
036
037 ;-----------------------------------------------------------------宏指令 scroll向上卷动一行
```

```
037 ;-----------------------------------------------------------------宏指令 scroll向上卷动一行
038 scroll macro cont,ulrow,ulcol,lrrow,lrcol
039         mov al,cont                 ;上卷行数
040         mov bh,70h                  ;卷入行属性
041         mov ch,ulrow                ;左上角行号
042         mov cl,ulcol                ;左上角列号
043         mov dh,lrrow                ;右下角行号
044         mov dl,lrcol                ;右下角列号
045         mov ah,6
046         int 10h                     ;BIOS调用，上卷
047 endm
048
049 ;-----------------------------------------------------------------宏指令locate_n 光标定位
050 locate_n macro y,x   ;设置光标在（y，x)位置，设置光标行列位置
051                      ;注意这里坐标是（行号，列号）的形式，行号是y，列号是x）
052     mov bh,0         ;显示页号
053     mov ah,2
054     mov dh,y         ;int10h中，ah=2时，设置光标位置，dh为行号，dl为列号
055     mov dl,x
056     int 10h
057     endm
058
059 ;-----------------------------------------------------------------在当前窗口和下窗口显示
060 show macro
061     mov bh,0                ;显示页
062     mov cx,1                ;cx = 字符重复个数，即重复一次
063     mov ah,0ah              ;在当前的窗口的光标位置处打印字符和属性
064     int 10h
065
066     locate_n nowy,nowx      ;重新设置当前光标位置到下窗口的正确位置处
067     mov bh,0
068     mov cx,1
069     mov ah,0ah              ;在下窗口输出
070     int 10h
071 endm
072
073 ;-----------------------------------------------------------------接收字符并判断
```

```
getchar macro                    ;一个一个读入，判断，输出
   input:                        ;输入键盘上的字符
       mov ah,0
       int 16h                   ;BIOS调用，从键盘读字符

       cmp ah,4bh                ;判断输入字符是否为左向键扫描码
       jnz no_left               ;输入为左向键时，继续执行；否则跳转no_left
       locate_n lefty,leftx      ;向左，光标重新定位到左窗口
       mov windowstat,2          ;更新窗口状态
       jmp input                 ;继续输入

   no_left:                      ;不跳转，已跳转至左窗口/保持在右窗口
       cmp ah,4dh                ;判断输入字符是否为右向键
       jnz no_right              ;输入为右向键时，继续执行；否则跳转no_right
       locate_n righty,rightx    ;向右，光标重新定位到有窗口
       mov windowstat,1          ;更新窗口状态
       jmp input                 ;继续输入

   no_right:
       cmp ah,01h                ;判断输入字符是否为ESC
       jnz  continue             ;没有结束，继续执行
       clear 0,0,24,79           ;屏幕总大小：25*80
       ret

    continue:
       show                      ;不是左右键和ESC,则嵌套调用宏show（显示输入的可打印的字符）
       inc nowx                  ;先显示字符，再将下窗口该行的光标的列号加1，再来比较是否要上卷

       cmp nowx,65               ;判断下窗口是否上卷
       jle goright               ;<=65: isright表示下窗口中当前光标行可以向右走
       scroll 1,18,15,22,65      ;>65:  到了一行的最右端，要向上卷一行
       mov nowx,15               ;置位：上卷一行后，当前列号变为当前窗口最左边的列的列号

   goright:                      ;判断左窗口/右窗口的右侧输入
       cmp windowstat,1          ;判断是否在右窗口
       jnz showleft              ;在右窗口：showright;否则，showleft

   showright:                    ;判断右窗口是否上卷
       inc rightx                ;列数+1
       cmp rightx,70
       jle rightexit             ;<= 70: rightexit表示右窗口中当前光标行可以向右走，不用上卷
       scroll 1,5,50,15,70       ;> 70 : 上卷一行
       mov rightx,50                ;置位，恢复列号

   rightexit:                    ;右窗口输出
      locate_n righty,rightx        ;定位到右窗口的光标位置
      jmp input                  ;继续输入

   showleft:                     ;判断左窗口是否上卷  （同right的操作）
      inc leftx
      cmp leftx,30
      jle leftexit
      scroll 1,5,10,15,30
```

```
   showleft:                     ;判断左窗口是否上卷  （同right的操作）
      inc leftx
      cmp leftx,30
      jle leftexit
      scroll 1,5,10,15,30
      mov leftx,10

   leftexit:                     ;左窗口输出
      locate_n lefty,leftx
      jmp input                  ;继续输入

      endm        ;getchar macro endm
CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS
START:
    main proc far
        push ds
        sub ax,ax
        push ax
        mov ax,DATAS
        mov ds,ax
        ;初始化显示
        clear 0,0,24,79

        ;利用scroll设置登录时的三个窗口
        scroll 10,5,10,15,30
        scroll 10,5,50,15,70
        scroll 4,18,15,22,65 ;上卷4行，是因为下窗口一共能容纳4行，得到一个空内容的窗口
        locate_n inity,initx    ;设置初始化的光标位置
        getchar
        ret
    main endp
CODES ENDS
    END START
```