汇编语言下的菜单技术

浙江台州地区人民银行 林荣庆

菜单技术是实现应用系统程序模块化的工具之一，也是人机交互的窗口，良好的菜单形式可使用户方便和有效地操作软件。

传统的菜单是全屏幕菜单，编程简单，但有明显的缺点，不仅缺缺乏层次感和动感，而且由于需要变更菜单须频繁清屏，导致屏幕闪烁，另外只有一个层次，不能实现多级菜单，不能跟踪功能的调用，显得呆板和陈旧。

目前结合窗口技术，出现了不少优秀的菜单方案。例如，条形菜单把菜单内容显示在屏幕的顶部或底部，在整个相关操作中固定保持在屏幕上，常作为主控菜单使用；下拉菜单和迭压菜效果相同，能够在屏幕上直观地表现出用户使用功能的踪迹，增强软件使用的透明度，子菜单都出现在父层菜单的选择项下面，并部分复盖父层菜单；弹出式菜单采用逻辑屏幕方法，将菜单处理成弹出窗口，速度快，能很好地解决屏幕的内烁问题；T 形菜单由顶部条形菜单和下拉菜单组成。上面这些菜单方案使界面轻松活泼友好，受到用户的欢迎。

选择方式上有，用序号或字母作为选择代码，分别对应于不同的功能，仅需按某个代码键就转入相应的功能，一般不按回车键，也可通过光标键移动光标或彩条到某个选择项上，按回车键给予确认，则转到相应的菜单功能上，或者上面二种方法兼而用之。

当执行具体功能时要撤消菜单显示，以便空出屏幕供其使用，同时在完成某功能后又要返回原来的菜单，把清除掉的屏幕恢复原样，以便用户跟踪选择，这就要用到屏幕的保存和恢复技术，屏幕可保存到磁盘或内存中。

汇编语言编制的程序易被其它高级语言调用，成为子程序。笔者用汇编语言，综合上述各种菜单形式，编制了一个菜单程序。菜单共有三级，主控菜单常驻屏幕顶部，通过按及光标键移动光标进行选择，当选中某项时按回车键则执行。第二级菜采用下拉式，按字母键进行选择。第三级菜单采用迭压式，迭加在第二级菜单上，按序号进行选择，执行具体功能。当执行第三级菜单上某功能前，首先保存屏幕上的菜单信息，然后可转到其它高级语言程序执行，执行完毕后恢复原菜单，屏幕的保存和恢复在显示缓冲区和内存缓冲区间进行，这样弹出屏幕快。为缩短程序长度，本示范程序的一级菜单只选择“输入数据”，第二级菜单只选择“输入月报”，第三级菜单只选择“格式一”，读者可根据需要进行扩展。源程序附在后面。

源程序清单：

```
B>type cd·asm
data segment para public 'data'
b11 db0
b12 db12
b13 db4
buff1 db '输入数据  统计数据  输出数据  退出运行'$
buff2 db '现在输入月报格式一，按任一键后返回!'$
srcd0 db '                                    '
srcd1 db '输入月报------(A)'
srcd2 db '输入年报------(B)'
srcd3 db '退出--------(Q)'
srcd4 db '请选择----------'
srcd5 db '                                    '
srgs0 db '                                    '
srgs1 db '格式一--------(1)'
srgs2 db '格式二--------(2)'
srgs3 db '退出--------(q)'
srgs4 db '请选择----------'
srgs5 db '----------------'
x       db 0
y       db 0
buff db 8192 dup (0)
data ends
code segment para public 'code'
        assume cs:code,ds:data
zcd     proc far
        push ds
        mov ax,0
        push ax
        mov ax。data
        mov ds,ax
        mov es,ax
        mov ah,0
```

```
        mov al, 3
        int 10h
        mov ah, 02h
        mov bh, 0
        mov dx, 0108h
        int 10h
        mov ah, 09h
        mov dx, offset buff1
        int 21h
        mov ah, 02h
        mov bh, 0
        mov dx, 0108h
        int 10h
        mov bll , 1
xz:     mov ah, 0
        int 16h
        cmp ax, 4800h
        jz xzl
        cmp ax, 5000h
        jz xz2
        cmp al, 0dh
        jz xz3
        jmp xz
xzl:    call upp
        jmp xz
xz2:    call dwp
        jmp xz
xz3 :   cmp bll , 4
        jz xz4
        call reu
        jmp xz
xz4:    mov ah, 0
        mov al, 3
        int 10h
        ret
zcd     endp
upp     proc near
        cmp bll , 1
        jz e4
        dec bll
        mov ah, 02h
        mov bh, 0
        mov dh, 01
        sub dl, 16
        int 10h
        ret
e4:     ret
upp     endp
dwp     proc near
        mov al, bl3
        cmp bll , al
        jz e5
        inc bll
        mov ah, 02h
        mov bh, 0
        mov dh, 01
        add dl, 16
        int 10h
        ret
e5:     ret
dwp     endp
reu     proc near
        mov ah, 03h
        mov bh, 0
        mov y , dh
        mov x , dl
        int 10h
        cmp bll , 1
        jz srsj
        ret
srsj:   mov ah, 13h
        mov dx, data
        mov ds, dx
        mov bp, offset srcd0
        mov cx, 18
        mov bx, 000fh
        mov dx, 0505h
        mov dl, 0
        int 10h
        mov bp, offset srcd1
        mov dx, 0605h
        int 10h
        mov bp, offset srcd2
        mov dx, 0705h
        int 10h
        mov bp, offset srcd3
        mov dx, 0805h
        int 10h
        mov bp, offset srcd4
        mov dx, 0805h
        int 10h
        mov bp, offset srcd4
        mov dx, 0905h
        int 10h
        mov bp, offset srcd5
        mov dx, 0a05h
srxz:   mov ah, 02h
        mov dx, 0915h
        mov bh, 0
        int 10h
        mov ah, 01h
        int 21h
        cmp al, 'a '
        jz yb
        cmp al, 'q '
        jz tu
        jmp srxz
tu:     mov ah, 06h
        mov al, 0
        mov cx, 0505h
        mov dx, 0albh
        mov bh, 7
        int 10h
```

```
        mov ah,02h
        mov bh,0
        mov dh,y
        mov dl,x
        int 10h
        ret
yb:     mov ah,13h
        mov dx,data
        mov es,dx
        mov bp,offset srgs0
        mov cx,20
        mov bx,000fh
        mov al,0
        mov dx,0705h
        int 10h
        mov hp,offset srgs1
        mov dx,0805h
        int 10h
        mov bp,offset srgs2
        mov dx,09005h
        int 10h
        mov bp,offset srgs3
        mov dx,0a05h
        int 10h
        mov bp,offset srgs4
        mov dx,0b05h
        int 10h
        mov bp,offset srgs5
        mov dx,0c05h
        int 10h
xzll:   mov ah,02h
        mov bh,0
        mov dx,0b17h
        int 10h
        mov ah,01h
        int 21h
        cmp al,'1'
        jz xzl2
        cmp al,'q'
        jz tul
        jmp xzll
xzl2:   call save
        mov ah,02h
        mov bh,0
        mov dx,0e18h
        int 10h
        mov ah,09h
        mov dx,data
        mov ds dx
        mov dx,offset buff2
        int 21h
        mov ah,01h
        int 21h
        call read
        jmp xzll
tul:    mov ah,06h
        mov al,0
        mov cx,0705h
        mov dx,0c1bh
        mov bh,7
        int 10h
        jmp srsj
reu     endp
save    proc near
        mov dx,0b000h
        mov ds,dx
        mov bx,data
        mov es,bx
        mov si,offset 0000h
        mov di,offset buff
        mov cx,1000h
        repz movsb
        mov si,offset 8000h
        mov di,offset buff+4096
        mov cx,1000h
        repz movsb
        mov ah,06h
        mov al,0
        mov cx,0303h
        mov dx,1850h
        mov bh,7
        int 10h
        ret
save    endp
read    proc near
        mov dx,data
        mov ds,dx
        mov bx,0b000h
        mov es,bx
        mov si,offset buff
        mov di,offset 000h
        mov cx,1000h
        repz movsb
        mov si,offset buff+4096
        mov di,offset 8000h
        mov cx,100h
        repz movsb
        mov ah,05h
        mov al,0
        int 10h
        ret
read    endp
code    ends
end     zcd
```