监测点14.1

(1)

assume cs:code

code segment

start:

mov al,2

out 70h,al

in al,71h

mov ax,4c00h

int 21h

code ends

end start

(2)

assume cs:code

code segment

start:

mov al,2

out 70h,al

mov al,0

out 71h,al

mov ax,4c00h

int 21h

code ends

end start

监测点14.2

assume cs:code

code segment

start:

shl al,1

mov bx,ax

mov cl,2

shl al,cl

add ax,bx

mov ax,4c00h

int 21h

code ends

end start

监测点16.1

assume cs:code

code segment

a dw 1,2,3,4,5,6,7,8

b dd 0

start:

mov si,0

mov cx,8

s:mov ax,a[si]

add word ptr b[0],ax

adc word ptr b[2],0

add si,2

loop s

mov ax,4c00h

int 21h

code ends

end strat

监测点17.1

错误，因为设置IF=1是为了保证在读取键盘缓冲区的时候键盘中断可以相应，可以键入数据到缓冲区，但int9属于内部中断不受IF控制，所以无需特别设置IF=1

实验14

assume cs:code

data segment

db 9,8,7,4,2,0

data ends

code segment

start: mov ax,data

mov ds,ax

mov bx,0

mov cx,6

mov si,0

s: mov al,[bx]

out 70h,al

in al,71h

push cx

mov ah,al

mov cl,4

shr ah,cl

and al,00001111b

pop cx

add ah,30h

add al,30h

mov dx,0b800h

mov es,dx

mov byte ptr es:[160\*12+2\*30+si],ah

mov byte ptr es:[160\*12+2\*30+2+si],al

cmp bx,2

jna na2

cmp bx,5

jnb nb5

mov al,3ah

mov byte ptr es:[160\*12+2\*30+4+si],al

jmp ok

nb5: jmp endloop

na2: je equal2

mov al,2fh

mov byte ptr es:[160\*12+2\*30+4+si],al

jmp ok

equal2: mov al,20h

mov byte ptr es:[160\*12+2\*30+4+si],al

jmp ok

ok: add si,6

inc bx

jmp s

endloop:mov ax,4c00h

int 21h

code ends

end start

实验15

assume cs:code

stack segment

db 128 dup(0)

stack ends

code segment

start:

mov ax,stack

mov ss,ax

mov sp,128

mov ax,0

mov es,ax

push es:[9\*4]

pop es:[200h]

push es:[9\*4+2]

pop es:[202h]

mov ax,cs

mov ds,ax

mov si,offset int9

mov di,204h

mov cx,offset int9end-offset int9

cld

rep movsb

cli

mov word ptr es:[9\*4],204h

mov word ptr es:[9\*4+2],0

sti

mov ax,4c00h

int 21h

int9:

push ax

push es

push cx

push di

in al,60h

pushf

call dword ptr cs:[200h]

cmp al,1eh+80h

jne int9ret

mov ax,0b800h

mov es,ax

mov cx,2000

mov di,0

mov al,65

s:

mov es:[di],al

add di,2

loop s

int9ret:

pop di

pop cx

pop es

pop ax

iret

int9end: nop

code ends

end star

实验17

assume cs:code

code segment

start: mov ax,cs

mov ds,ax

mov si,offset int7cstart

mov ax,0

mov es,ax

mov di,200h

mov cx,offset int7cend-offset int7cstart

cld

rep movsb

cli

mov ax,0

mov ds,ax

mov word ptr ds:[4\*7ch],200h

mov word ptr ds:[4\*7ch+2],0

sti

mov ax,4c00h

int 21h

int7cstart: cmp ah,1

ja ok

push ax

push bx

push cx

push dx

push ax

mov ax,dx

mov dx,0

mov cx,1440

div cx

push ax

mov cx,18

mov ax,dx

mov dx,0

div cx

push ax

inc dx

push dx

pop ax

mov cl,al

pop ax

mov ch,al

pop ax

mob dh,al

mov dl,0

pop ax

mov al,1

cmp ah,0

je read

cmp ah,1

je write

read: mov ah,2

jmp short func

write: mov ah,3

jmp short func

func: int 13h

pop dx

pop cx

pop bx

pop ax

ok: iret

int7cend: nop

code ends

end start