检测点11.3

（1）

代码如下：

*mov* ax,0f000h

*mov* ds,ax

*mov* bx,0

*mov* dx,0

*mov* cx,32

s:*mov* al,[bx]

*cmp* al,32

*jb* s0

*cmp* al,128

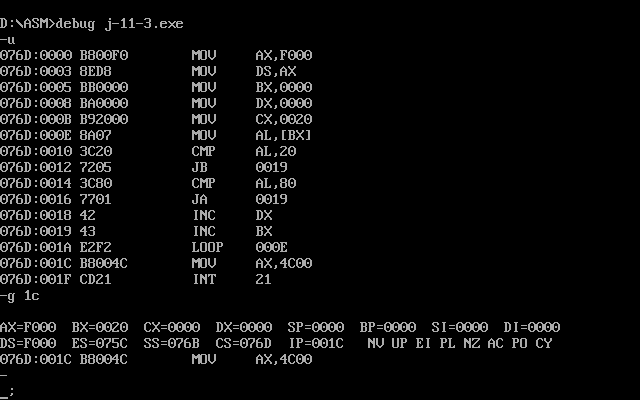
*ja* s0

*inc* dx

s0:*inc* bx

*loop* s

结果如下：



可以看到结果显示为dx=0000h

因为f000:0000处的值均为零，即被筛出的个数为bx=0020h=32

（2）

代码如下：

*mov* ax,0f000h

*mov* ds,ax

*mov* bx,0

*mov* dx,0

*mov* cx,32

s:*mov* al,[bx]

*cmp* al,32

*jna* s0

*cmp* al,128

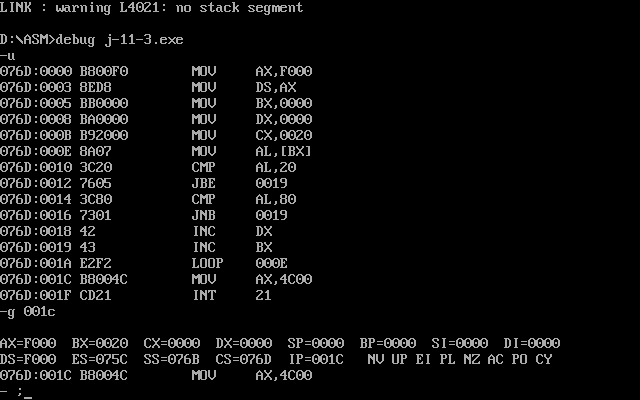
*jnb* s0

*inc* dx

s0:*inc* bx

*loop* s

结果如下：



可以看到结果显示为dx=0000h

因为f000:0000处的值均为零，即被筛出的个数为bx=0020h=32

实验11

代码如下：

*assume* cs:codesg,ds:data

data segment

    db "Beginner's All-purpose Symbolic Instruction Code.",0

data ends

codesg segment

begin:*mov* dx,data

*mov* ds,ax

*mov* si,0

*call* letterc

over:

*mov* ax,4c00h

*int* 21h

letterc:

*mov* cx,0

*mov* cl,ds:[si]

    jcxz over

*cmp* byte ptr [si],97*;大写*

*jb* next

*cmp* byte ptr [si],122*;小写*

*ja* next

*sub* byte ptr [si],20h

next:

*inc* si

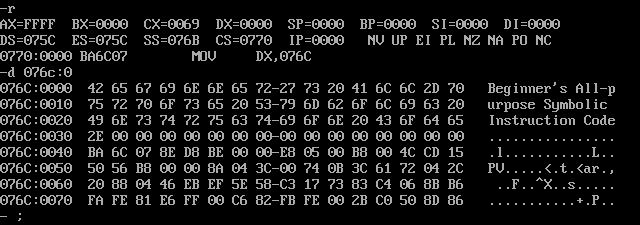
*jmp* short letterc

codesg ends

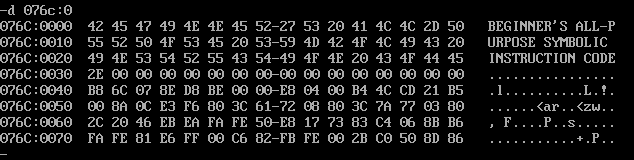
end begin

结果如下：

原数据



更改后数据：



实验12

代码如下：

*assume* cs:code

code segment

start:

*mov* ax,cs

*mov* ds,ax

*mov* si,offset do0

*mov* ax,0

*mov* es,ax

*mov* di,200h

*mov* cx,offset do0end-offset do0

*cld*

    rep *movsb*

*mov* ax,0

*mov* es,ax

*mov* word ptr es:[0\*4],200h

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

*;除法溢出*

*mov* ax,1000h

*mov* bx,0

*mov* bl,1

*div* bl

*mov* ax,4c00h

*int* 21h

do0:

*jmp* short do0start

    db "divide error!"

 do0start:

*mov* ax,cs

*mov* ds,ax

*mov* si,202h

*mov* ax,0b800h

*mov* es,ax

*mov* di,12\*160+35\*2

*mov* cx,13

*mov* ah,00000100b

s:

*mov* al,ds:[si]

*mov* es:[di],ax

*inc* si

*add* di,2

*loop* s

*mov* ax,4c00h

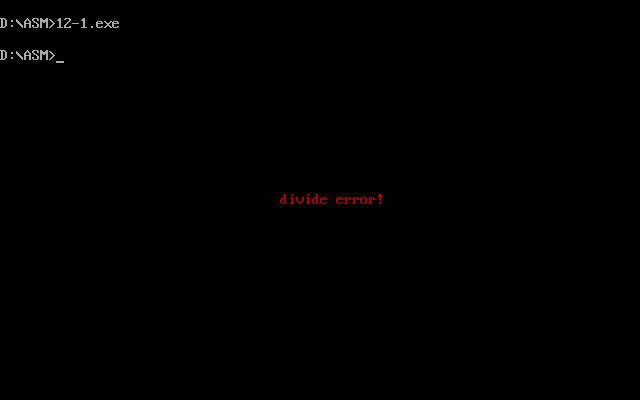
*int* 21h

do0end:*nop*

code ends

end start

结果如下：



检测点13.1

（1）

答：程序通过add [bp+2],bx实现位移，而bx作为无符号数时范围最大，其范围为[0，65535]

故转移位移结果为65535。

（2）

代码如下：

*assume* cs:code

data segment

    db 'coversation',0

data ends

code segment

start:

*mov* ax,cs

*mov* ds,ax

*mov* si,offset cst

*mov* ax,0

*mov* es,ax

*mov* di,200h

*mov* cx,offset ce - offset cst

*cld*

    rep *movsb*

*mov* ax,0

*mov* es,ax

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

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

*mov* ax,data

*mov* ds,ax

*mov* si,0

*mov* ax,0b800h

*mov* es,ax

*mov* di,12\*160+35\*2

s:

*cmp* byte ptr [si],0

*je* ok

*mov* al,ds:[si]

*mov* es:[di],al

*inc* si

*add* di,2

*mov* bx,offset s - offset ok

*int* 7ch

ok:

*mov* ax,4c00h

*int* 21h

cst:

*push* bp

*mov* bp,sp

*add* [bp+2],bx

*pop* bp

*iret*

ce: *nop*

code ends

end start

结果如下：



检测点13.2

1. 答：错误，因为bios是先于程序启动的，即使改掉了其硬件检测系统与初始化程序，它也早已运行结束，同时系统默认ffff:0000为可读，不可修改
2. 答：错误，因为19h是dos开始启动的程序，是先于dos启动的程序，所以不可以由dos提供

实验13

（1）

代码如下：

*assume* cs:code

code segment

start:*mov* ax,cs

*mov* ds,ax

*mov* si,offset show

*mov* ax,0

*mov* es,ax

*mov* di,200h

*mov* cx,offset showend-offset show

*cld*

    rep *movsb*

*;安装中断向量表*

*mov* ax,0

*mov* es,ax

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

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

*mov* ax,4c00h

*int* 21h

show:

*push* es

*push* di

*push* ax

*push* cx

*mov* ax, 0b800H

*mov* es, ax

*mov* ax, 160

*mul* dh

*mov* di, ax

*mov* ax, 2

*mul* dl

*add* di, ax           *; es=0b800H, di=dh\*160+dl\*2, es:di为显示区地址*

*mov* al, cl           *; 颜色存到al中*

s0:

*mov* ch, 0

*mov* cl, [si]

    jcxz printret        *; 判断字符串是否结束，结束则返回*

*mov* es:[di], cl      *; 显示字符*

*mov* es:[di+1], al    *; 字符颜色*

*inc* si

*add* di, 2

*jmp* short s0

printret:

*pop* cx

*pop* ax

*pop* di

*pop* es

*iret*

showend:

*nop*

code ends

end start

结果如下：



（2）

代码如下

*assume* cs:code

code segment

start:*mov* ax,cs

*mov* ds,ax

*mov* si,offset lp

*mov* ax,0

*mov* es,ax

*mov* di,200h

*mov* cx,offset lpend-offset lp

*cld*

    rep *movsb*

*;安装中断向量表*

*mov* ax,0

*mov* es,ax

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

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

*;测试程序*

*mov* ax, 0b800H

*mov* es, ax

*mov* di, 160\*12

*mov* bx, offset s-offset se      *; 设置从标号s到标号se的位移*

*mov* cx, 80

s:

*mov* byte ptr es:[di], '!'

*add* di, 2

*int* 7cH                         *; 如果(cx)!=0，转移到标号s处*

se:

*nop*

*mov* ax, 4c00H

*int* 21H

lp:*push* bp

*mov* bp,sp

*dec* cx

    jcxz lpret

*add* [bp+2],bx

lpret:*pop* bp

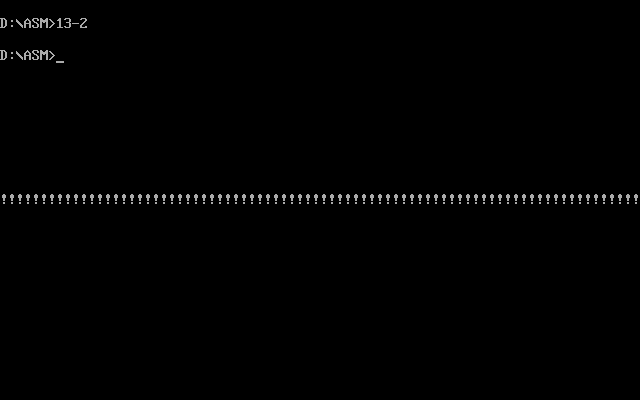
*iret*

lpend: *nop*

code ends

end

结果如下：



（3）

代码如下：

*assume* cs:code

code segment

s1: db 'Good,better,best,','$'

s2: db 'Never let it rest,','$'

s3: db 'Till good is better,','$'

s4: db 'And better,best.','$'

s:  dw offset s1,offset s2,offset s3,offset s4

row:db 2,4,6,8

start:  *mov* ax,cs

*mov* ds,ax

*mov* bx,offset s

*mov* si,offset row

*mov* cx,4

ok:     *mov* bh,0

*mov* dh,[si]

*mov* dl,0

*mov* ah,2

*int* 10h

*mov* dx,[bx]

*mov* ah,9

*int* 21h

*inc* si

*add* bx,2

*loop* ok

*mov* ax,4c00h

*int* 21h

code ends

end start

结果如下：

