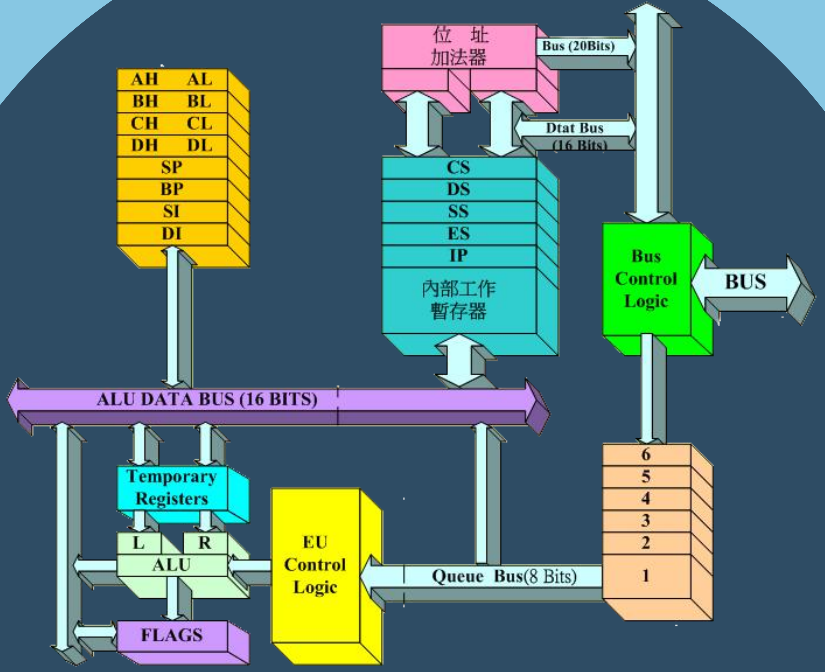
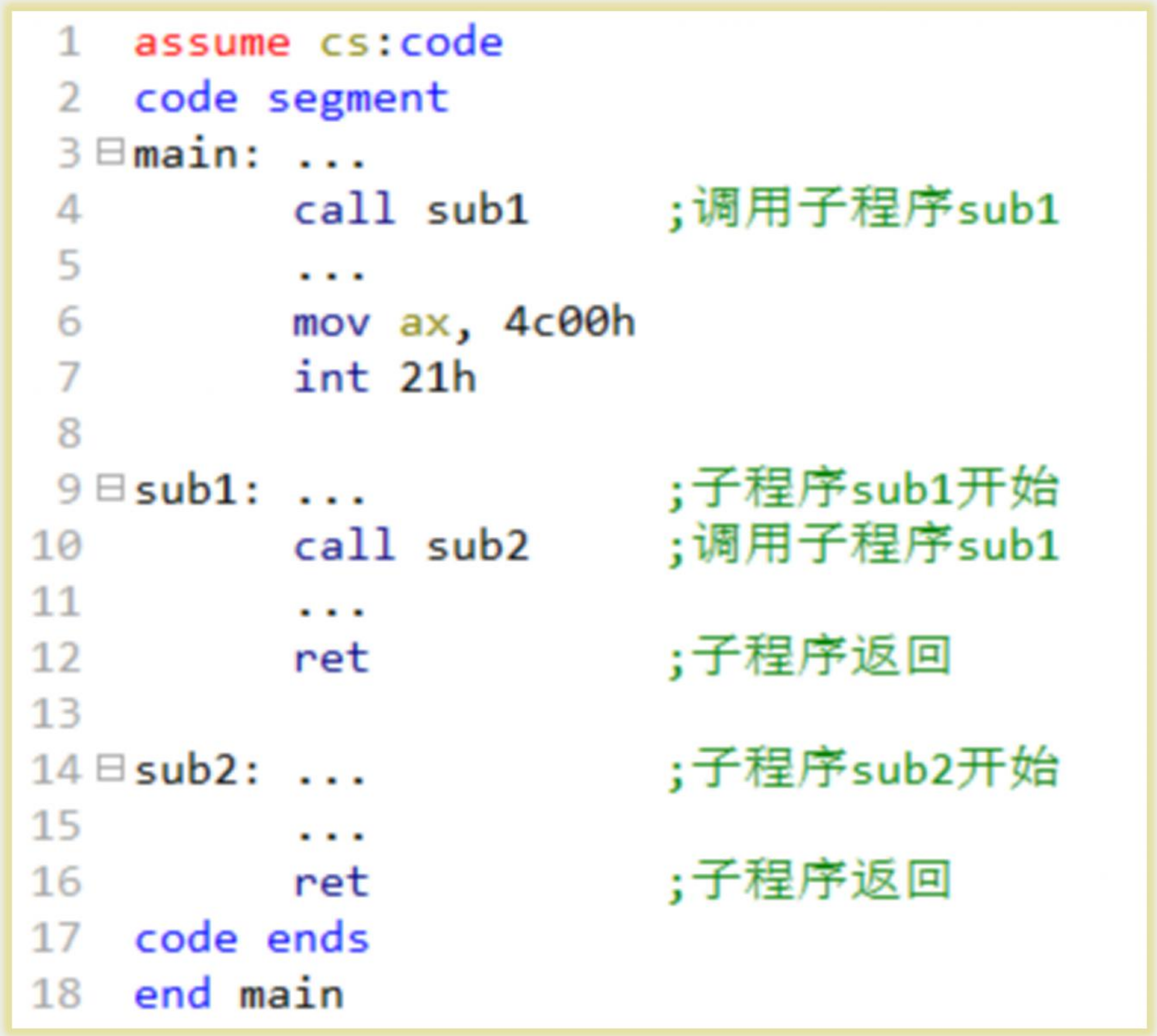
call 和 ret 的配合使用

贺利坚　主讲



汇编语言程序设计
Assembly Language

具有子程序的源程序的框架



```
assume cs:code
code segment
main: ...
        call sub1       ;调用子程序sub1
        ...
        mov ax, 4c00h
        int 21h

sub1: ...               ;子程序sub1开始
        call sub2       ;调用子程序sub1
        ...
        ret             ;子程序返回

sub2: ...               ;子程序sub2开始
        ...
        ret             ;子程序返回
code ends
end main
```

调用程序的框架

```
… …
call 标号
… …
```

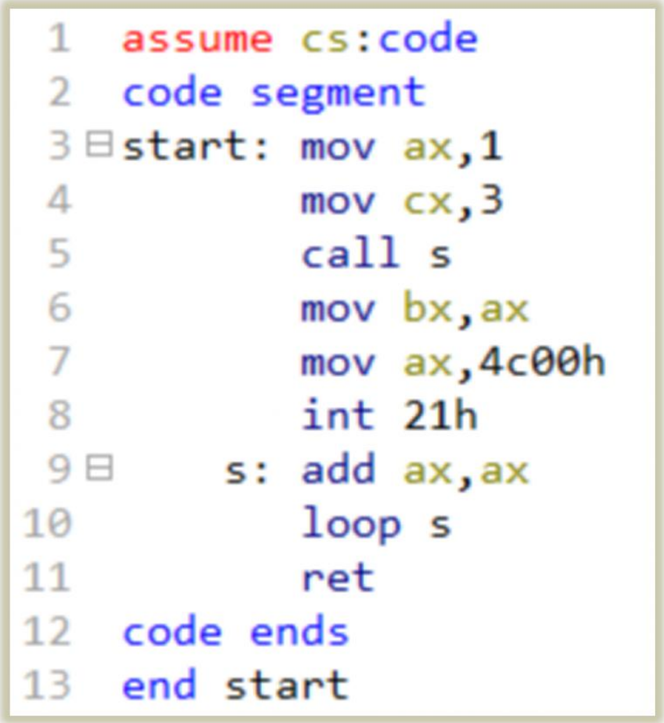
子程序的框架

```
标号:
    指令
    ret
```

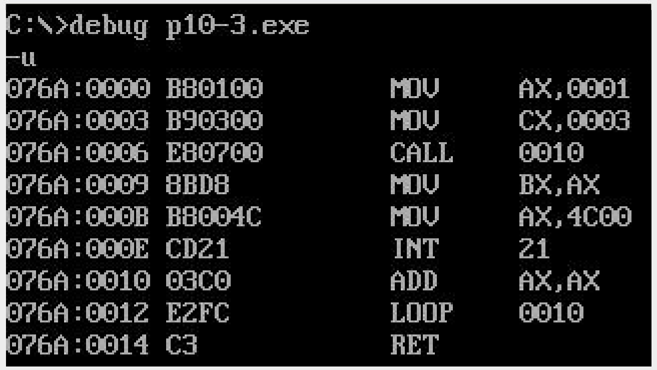
call 和 ret 的配合使用

- 例：

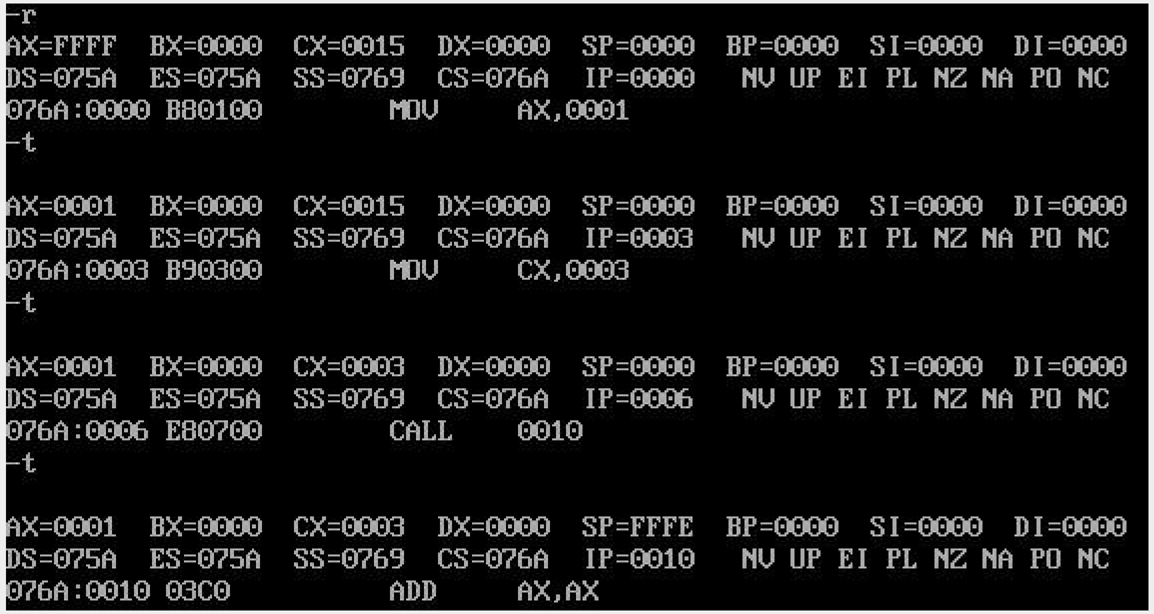
计算2的N次方，计算前，N的值由CX提供。



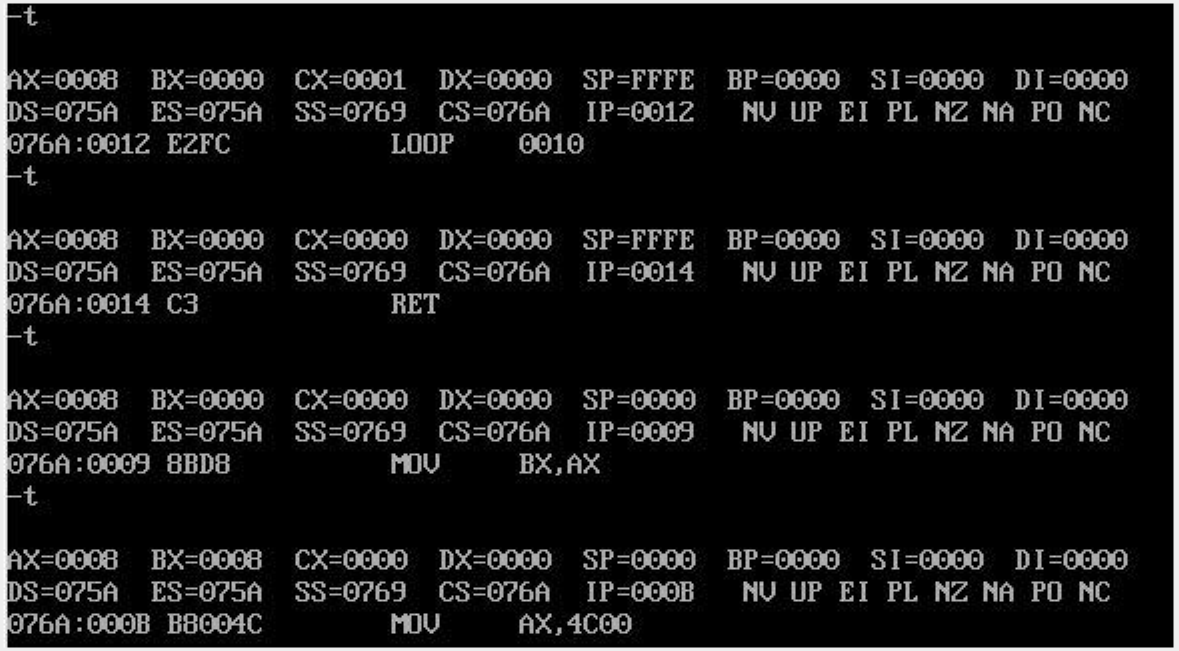
```
assume cs:code
code segment
start: mov ax,1
       mov cx,3
       call s
       mov bx,ax
       mov ax,4c00h
       int 21h
    s: add ax,ax
       loop s
       ret
code ends
end start
```



```
C:\>debug p10-3.exe
-u
076A:0000 B80100        MOV     AX,0001
076A:0003 B90300        MOV     CX,0003
076A:0006 E80700        CALL    0010
076A:0009 8BD8          MOV     BX,AX
076A:000B B8004C        MOV     AX,4C00
076A:000E CD21          INT     21
076A:0010 03C0          ADD     AX,AX
076A:0012 E2FC          LOOP    0010
076A:0014 C3            RET
```

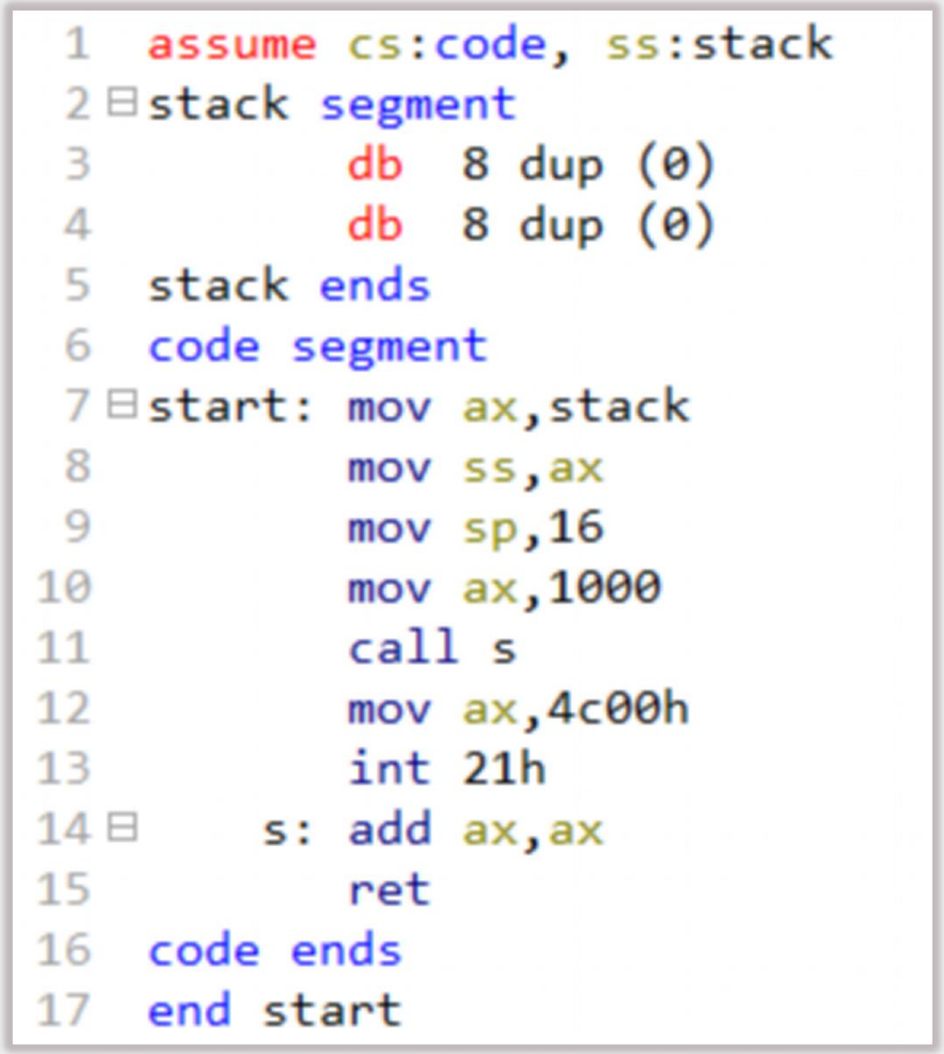


```
-r
AX=FFFF  BX=0000  CX=0015  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0000   NV UP EI PL NZ NA PO NC
076A:0000 B80100        MOV     AX,0001
-t

AX=0001  BX=0000  CX=0015  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0003   NV UP EI PL NZ NA PO NC
076A:0003 B90300        MOV     CX,0003
-t

AX=0001  BX=0000  CX=0003  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0006   NV UP EI PL NZ NA PO NC
076A:0006 E80700        CALL    0010
-t

AX=0001  BX=0000  CX=0003  DX=0000  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0010   NV UP EI PL NZ NA PO NC
076A:0010 03C0          ADD     AX,AX
```

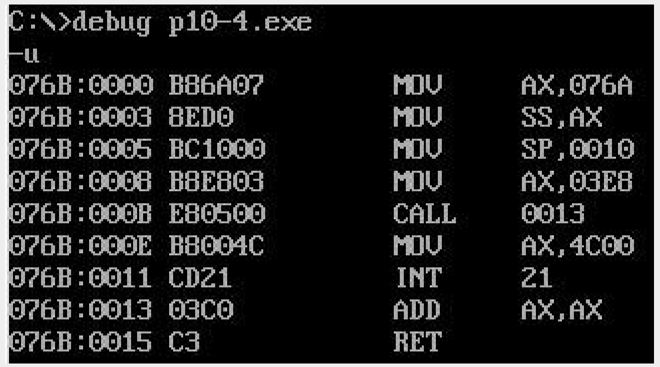


```
-t

AX=0008  BX=0000  CX=0001  DX=0000  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0012   NV UP EI PL NZ NA PO NC
076A:0012 E2FC          LOOP    0010
-t

AX=0008  BX=0000  CX=0000  DX=0000  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0014   NV UP EI PL NZ NA PO NC
076A:0014 C3            RET
-t

AX=0008  BX=0000  CX=0000  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0009   NV UP EI PL NZ NA PO NC
076A:0009 8BD8          MOV     BX,AX
-t

AX=0008  BX=0008  CX=0000  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=000B   NV UP EI PL NZ NA PO NC
076A:000B B8004C        MOV     AX,4C00
```

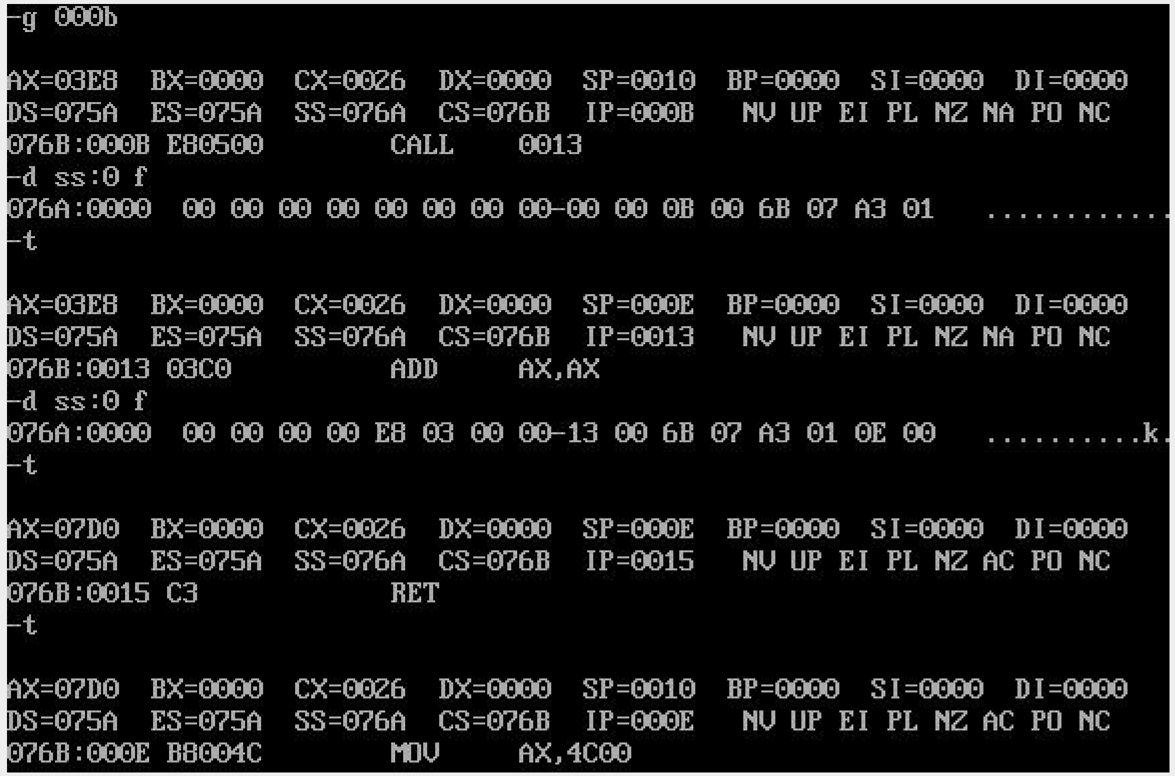
例： 为call和ret指令设置栈



```
assume cs:code, ss:stack
stack segment
        db   8 dup (0)
        db   8 dup (0)
stack ends
code segment
start: mov ax,stack
       mov ss,ax
       mov sp,16
       mov ax,1000
       call s
       mov ax,4c00h
       int 21h
    s: add ax,ax
       ret
code ends
end start
```



```
C:\>debug p10-4.exe
-u
076B:0000 B86A07        MOV     AX,076A
076B:0003 8ED0          MOV     SS,AX
076B:0005 BC1000        MOV     SP,0010
076B:0008 B8E803        MOV     AX,03E8
076B:000B E80500        CALL    0013
076B:000E B8004C        MOV     AX,4C00
076B:0011 CD21          INT     21
076B:0013 03C0          ADD     AX,AX
076B:0015 C3            RET
```



```
-g 000b

AX=03E8  BX=0000  CX=0026  DX=0000  SP=0010  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=076A  CS=076B  IP=000B   NV UP EI PL NZ NA PO NC
076B:000B E80500        CALL    0013
-d ss:0 f
076A:0000  00 00 00 00 00 00 00 00-00 00 0B 00 6B 07 A3 01   ...........
-t

AX=03E8  BX=0000  CX=0026  DX=0000  SP=000E  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=076A  CS=076B  IP=0013   NV UP EI PL NZ NA PO NC
076B:0013 03C0          ADD     AX,AX
-d ss:0 f
076A:0000  00 00 00 00 E8 03 00 00-13 00 6B 07 A3 01 0E 00   .........k.
-t

AX=07D0  BX=0000  CX=0026  DX=0000  SP=000E  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=076A  CS=076B  IP=0015   NV UP EI PL NZ AC PO NC
076B:0015 C3            RET
-t

AX=07D0  BX=0000  CX=0026  DX=0000  SP=0010  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=076A  CS=076B  IP=000E   NV UP EI PL NZ AC PO NC
076B:000E B8004C        MOV     AX,4C00
```