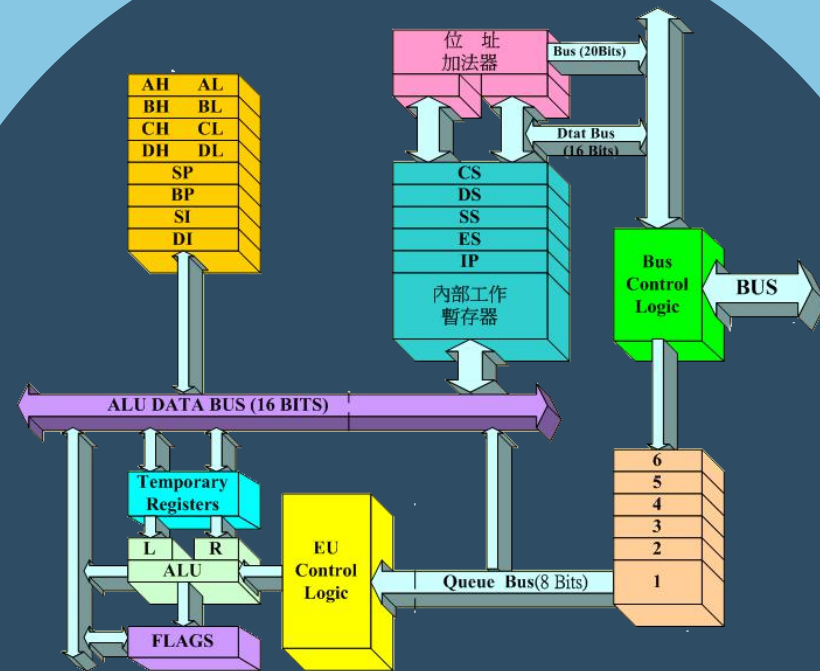


# 在代码段中使用栈

贺利坚 主讲



汇编语言程序设计  
Assembly Language

# 在代码段中使用栈：以数据逆序存放为例

💻问题：完成下面的程序，利用栈，将程序中定义的数据逆序存放。

```
assume cs:codesg
codesg segment
    dw 0123h,0456h,0789h,0abch,0defh,0fedh,0cbah,0987h
    ?
code ends
end
```

💻程序的思路大致如下：

- 📁 程序运行时，定义的数据存放在cs:0~cs:F单元中，共8个字单元。
- 📁 依次将这8个字单元中的数据入栈，然后再依次出栈到这8个字单元中，从而实现数据的逆序存放。
- 📁 栈需要的内存空间，在程序中通过定义“空”数据来取得。

0987H
0cbaH
0fedH
0defH
0abch
0789H
0456H
0123H

入栈后的数据

# 数据逆序存放程序

问题：完成下面的程序，利用栈，将程序中定义的数据逆序存放。

```
assume cs:codesg
codesg segment
    dw 0123H,0456H,0789H,0abcH,0defH,0fedH,0cbaH,0987H
    dw 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
start: mov ax,cs
      mov ss,ax
      mov sp,30h
      ; 入栈
      ; 出栈
      mov ax,4c00h
      int 21h
codesg ends
end start
```

```
mov bx,0
mov cx,8
s: push cs:[bx]
   add bx,2
   loop s
```

```
mov bx,0
mov cx,8
s0: pop cs:[bx]
    add bx,2
    loop s0
```

0123H	CS:0	0987H	CS:0
0456H	CS:2	0cbaH	CS:2
0789H	CS:4	0fedH	CS:4
0abcH	CS:6	0defH	CS:6
0defH	CS:8	0abcH	CS:8
0fedH	CS:a	0789H	CS:a
0cbaH	CS:c	0456H	CS:c
0987H	CS:e	0123H	CS:e
0	CS:10	0	CS:10
0	...	0	...
0	CS:20	0987H	CS:20
0	CS:22	0cbaH	CS:22
0	CS:24	0fedH	CS:24
0	CS:26	0defH	CS:26
0	CS:28	0abcH	CS:28
0	CS:2a	0789H	CS:2a
0	CS:2c	0456H	CS:2c
0	CS:2e	0123H	CS:2e
	CS:30		CS:30

# 在Debug中的执行结果

```
1  assume cs:codesg
2  codesg segment
3      dw 0123H,0456H,0789H,0abcH,0defH,0fedH,0cbaH,0987H
4      dw 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
5
6  start: mov ax,cs
7         mov ss,ax
8         mov sp,30h
9         mov bx,0
10        mov cx,8
11  s:    push cs:[bx]
12        add bx,2
13        loop s
14
15        mov bx,0
16        mov cx,8
17  s0:   pop cs:[bx]
18        add bx,2
19        loop s0
20
21        mov ax,4c00h
22        int 21h
23 codesg ends
24 end start
```

C:\>debug p6-3.exe

-r

AX=FFFF BX=0000 CX=005B DX=0000 SP=0000 BP=0000 SI=0000 DI=0000  
DS=075A ES=075A SS=0769 CS=076A IP=0030 NV UP EI PL NZ NA PO NC  
076A:0030 BCCB MOV AX,CS

-u cs:0030

076A:0030	BCCB	MOV	AX,CS
076A:0032	8ED0	MOV	SS,AX
076A:0034	BC3000	MOV	SP,0030
076A:0037	BB0000	MOV	BX,0000
076A:003A	B90800	MOV	CX,0008
076A:003D	2E	CS:	
076A:003E	FF37	PUSH	[BX]
076A:0040	83C302	ADD	BX,+02
076A:0043	E2F8	LOOP	003D
076A:0045	BB0000	MOV	BX,0000
076A:0048	B90800	MOV	CX,0008
076A:004B	2E	CS:	
076A:004C	8F07	POP	[BX]
076A:004E	83C302	ADD	BX,+02

-d CS:0 2f

076A:0000	23 01 56 04 89 07 BC 0A-EF 0D ED 0F BA 0C 87 09	#.U.....
076A:0010	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
076A:0020	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....

-g

Program terminated normally

-d cs:0 2f

076A:0000	87 09 BA 0C ED 0F EF 0D-BC 0A 89 07 56 04 23 01	.....U.#.
076A:0010	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
076A:0020	87 09 BA 0C ED 0F EF 0D-BC 0A 58 00 6A 07 12 72	.....X.j..r