

# DF标志和串传送指令

贺利坚 主讲



汇编语言程序设计  
Assembly Language

# 问题的提出

💻编程：将data段中的第一个字符串复制到它后面的空间中。

```
data segment
```

```
    db 'Welcome to masm!'
```

```
    db 16 dup (0)
```

```
data ends
```



还能再简洁吗？

```
code segment
```

```
start: mov ax,data
```

```
        mov ds,ax
```

```
        mov si,0
```

```
        mov di,16
```

```
        mov cx,8
```

```
s: mov ax,[si]
```

```
        mov [di],ax
```

```
        add si,2
```

```
        add di,2
```

```
        loop s
```

```
        mov ax,4c00h
```

```
        int 21h
```

```
code ends
```

```
end start
```

# DF标志和串传送指令

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

DF-方向标志位 ( Direction Flag )

## 功能

- 在串处理指令中，控制每次操作后si，di的增减。
- DF = 0：每次操作后si，di递增；
- DF = 1：每次操作后si，di递减。

## 对DF位进行设置的指令：

- cld指令：将标志寄存器的DF位设为0(clear)
- std指令：将标志寄存器的DF位设为1(setup)

```
data segment
db 'Welcome to masm!'
db 16 dup (0)
data ends
```

```
mov cx,16
s: movsb
loop s
```

```
code segment
start:
;设置寄存器
;循环传送
mov ax,4c00h
int 21h
code ends
```

```
mov ax,data
mov ds,ax
mov si,0
mov es,ax
mov di,16
cld
```

串传送指令1： movsb

功能： ( 以字节为单位传送 )

- (1)  $((es) \times 16 + (di)) = ((ds) \times 16 + (si))$
- (2) 如果DF = 0则：  $(si) = (si) + 1$   
 $(di) = (di) + 1$   
如果DF = 1则：  $(si) = (si) - 1$   
 $(di) = (di) - 1$

串传送指令2： movsw

功能： ( 以字为单位传送 )

- (1)  $((es) \times 16 + (di)) = ((ds) \times 16 + (si))$
- (2) 如果DF = 0则：  $(si) = (si) + 2$   
 $(di) = (di) + 2$   
如果DF = 1则：  $(si) = (si) - 2$   
 $(di) = (di) - 2$

# rep指令

💻 rep指令常和串传送指令搭配使用

💻 功能：根据cx的值，重复执行后面的指令

💻 用法：

rep movsb



s : movsb  
loop s

rep movsw



s : movsw  
loop s

```
1  assume cs:code, ds:data
2  data segment
3      db 'Welcome to masm!'
4      db 16 dup (0)
5  data ends
6  code segment
7  start:
8      mov ax,data
9      mov ds,ax
10     mov si,0
11     mov es,ax
12     mov di,16
13     cld
14     mov cx,8
15     rep movsw
16
17     mov ax,4c00h
18     int 21h
19 code ends
20 end start
```

# 应用实例

任务：用串传送指令，将F000H段中的最后16个字符复制到data段中。

data segment

db 16 dup (0)

data ends

F000H段的最后一个字符  
的位置：F000:FFFF

```
-d f000:ffff0 ffff
F000:FFF0 EA C0 12 00 F0 30 31 2F-30 31 2F 39 32 00 FC 55 .....01/01/92..U
-d 076a:0 f
076A:0000 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
-g 16

AX=076A BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=FFEF DI=FFFF
DS=F000 ES=076A SS=0769 CS=076B IP=0016 NV DN EI PL NZ NA PO NC
076B:0016 B8004C      MOV     AX,4C00
-d 076a:0 f
076A:0000 EA C0 12 00 F0 30 31 2F-30 31 2F 39 32 00 FC 55 .....01/01/92..U
```

```
1  assume cs:code, ds:data
2  data segment
3      db 16 dup (0)
4  data ends
5  code segment
6  start:
7      mov ax,0f000h
8      mov ds,ax
9      mov si,0ffffh
10     mov ax,data
11     mov es,ax
12     mov di,15
13     mov cx,16
14     std
15     rep movsb
16
17     mov ax,4c00h
18     int 21h
19 code ends
20 end start
```