```
TITLE    A06MOVE (EXE)  Repetitive move operations
         .MODEL SMALL
         .STACK 64
         .DATA
HEADNG1 DB       'S09222017=>',0ah,0dh
Len     EQU      $ - HEADNG1 - 2
HEADNG2 DB       Len DUP ('*'), '$'
         .CODE
A10MAIN PROC     FAR
         MOV     AX, @data    ;Initialize segment
         MOV     DS,AX        ;  registers
         MOV     ES,AX

         MOV     CX,len               ;Initialize to move 9 chars(len)
         LEA     SI,HEADNG1           ;Initialize offset addresses
         LEA     DI,HEADNG2+len-1     ;  of HEADNG1 and HEADNG2
A20:
         MOV     AL,[SI]              ;Get character from HEADNG1,
         MOV     [DI],AL              ;  move it to HEADNG2
         INC     SI           ;Increment next char in HEADNG1
         DEC     DI           ;Increment next position in HEADNG2
         DEC     CX           ;Decrement count for loop
         JNZ     A20          ;Count not zero? Yes, loop finished
         MOV     AH,09H               ;Request display
         LEA     DX,HEADNG2   ;  of HEADNG2
         INT     21H

         MOV     AX,4C00H     ;End processing
         INT     21H
A10MAIN ENDP
         END     A10MAIN
```

從 HEADNG2 開始顯示

```
C:\MASM611\BIN>ml a06asm.asm
Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993.  All rights reserved.

 Assembling: a06asm.asm

Microsoft (R) Segmented Executable Linker  Version 5.31.009 Jul 13 1992
Copyright (C) Microsoft Corp 1984-1992.  All rights reserved.

Object Modules [.obj]: a06asm.obj
Run File [a06asm.exe]: "a06asm.exe"
List File [nul.map]: NUL
Libraries [.lib]:
Definitions File [nul.def]:

C:\MASM611\BIN>a06asm.exe
>=71022290S
```

```
TITLE    A06MOVE (EXE)  Repetitive move operations
         .MODEL SMALL
         .STACK 64
         .DATA
HEADNG1 DB       'S09222017=>',0ah,0dh
Len     EQU      $ - HEADNG1 - 2
HEADNG2 DB       Len DUP ('*'), '$'
         .CODE
A10MAIN PROC     FAR
         MOV     AX, @data      ;Initialize segment
         MOV     DS,AX          ;  registers
         MOV     ES,AX

         MOV     CX,len                ;Initialize to move 9 chars(len)
         LEA     SI,HEADNG1            ;Initialize offset addresses
         LEA     DI,HEADNG2+len-1     ;  of HEADNG1 and HEADNG2
A20:
         MOV     AL,[SI]              ;Get character from HEADNG1,
         MOV     [DI],AL              ;  move it to HEADNG2
         INC     SI             ;Increment next char in HEADNG1
         DEC     DI             ;Increment next position in HEADNG2
         DEC     CX             ;Decrement count for loop
         JNZ     A20            ;Count not zero? Yes, loop finished
         MOV     AH,09H                ;Request display
         LEA     DX,HEADNG1     ;  of HEADNG2
         INT     21H

         MOV     AX,4C00H       ;End processing
         INT     21H
A10MAIN ENDP
         END     A10MAIN
```

從 HEADNG1 開始顯示