

实验五 串处理指令的使用

一、实验目的

学习串操作指令、循环指令、转移指令的用法；掌握数据定义伪指令的格式，会用 DEBUG 中的 D 命令观察 DB、DW、DD 存储数据的格式；熟悉汇编语言的基本框架，掌握编写汇编语言程序的基本方法。

二、示 例

1、 下列程序实现的功能是把字符串 string 中的 ' & ' 字符用空格符代替。

STRING DB 'THE DATA IS FEB&03'

程序如下：

DATA SEGMENT

STRING DB 'THE DATA IS FEB&03'

N DW \$-STRING

DATA ENDS

CODE SEGMENT

ASSUME DS: DATA, CS: CODE

START:

MOV AX,DATA

MOV DS,AX

MOV AL,'&'

MOV CX,N

LEA DI,STRING

CLD

REPNE SCASB

MOV STRING[DI],'

MOV AH,4CH

INT 21H

CODE ENDS

END START

要求：用 debug 命令来调试该程序，**找出程序中的错误并修改**实现题目要求的功

能。

源代码的问题:

- REPNE SCASB 扫描后, DI 会指向“下一个”字节(即找到的字符的下一个位置), 所以 DI 应该-1。
- 没有判断是否真的找到 &, 应该判断 ZF 标志 (Zero Flag), 只有找到时才替换
- ES 和 DS 需要都指向 data 数据段, SCASB 才能正确扫描 STRING 字符串。

修改后的代码:

```
DATA SEGMENT
    STRING DB 'THE DATA IS FEB&03'    ; 定义字符串数据
    N      DW $-STRING                  ; 计算字符串长度
DATA ENDS
CODE SEGMENT
    ASSUME DS: DATA, CS: CODE          ; 设置段寄存器
START:
    MOV AX, DATA                       ; 初始化数据段
    MOV DS, AX
    MOV AL, '&'                         ; 设置要查找的字符'&' (如果是字类型的话
需要送入 AX)
    MOV CX, N                           ; 设置循环计数器为字符串长度
    LEA DI, STRING                      ; 加载字符串首地址到 DI (目的串)
    CLD                                ; 清除方向标志, 使串操作向高地址方向进行
    REPNE SCASB                         ; 重复扫描直到找到'&' 或 CX=0
    JNZ NOT_FOUND
    DEC DI                              ; 扫描后, DI 会指向“下一个”字节, 所以 DI-1
    MOV STRING[DI], ' '                 ; 将找到的'&' 替换为空格
NOT_FOUND:
    MOV AH, 4CH                         ; 设置 DOS 功能号
    INT 21H                             ; 调用 DOS 中断返回
CODE ENDS
    END START                           ; 程序结束
```

Debug 记录

-U 命令查看指令是否正确导入

```
(pt2) D:\code\EXPERIMENTAL_REPORT>debug 汇编语言\E5_串处理指令的使用\E1.exe
-u
204F:0000 B84D20      MOV     AX,204D
204F:0003 8ED8      MOV     DS,AX
204F:0005 8EC0      MOV     ES,AX
204F:0007 B026      MOV     AL,26
204F:0009 8B0E1200  MOV     CX,[0012]
204F:000D 8D3E0000  LEA     DI,[0000]
204F:0011 FC        CLD
204F:0012 F2        REPNZ
204F:0013 AE        SCASB
204F:0014 7504      JNZ     001A
204F:0016 4F        DEC     DI
204F:0017 C60520    MOV     BYTE PTR [DI],20
204F:001A B44C      MOV     AH,4C
204F:001C CD21      INT     21
204F:001E 0000      ADD     [BX+SI],AL
```

&替换前

```
-d 204D:0
204D:0000 54 48 45 20 44 41 54 41-20 49 53 20 46 45 42 26 THE DATA IS FEB&
204D:0010 30 33 12 00 00 00 00 00-00 00 00 00 00 00 00 00 03.....
204D:0020 B8 4D 20 8E D8 B0 26 8B-0E 12 00 8D 3E 00 00 FC 8M .X0&.....>..|
204D:0030 F2 AE 75 04 4F C6 05 20-B4 4C CD 21 00 00 00 00 r.u.OF. 4LM!....
204D:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
204D:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
204D:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
204D:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
```

&替换后结果

```
-d 204d:0
204D:0000 54 48 45 20 44 41 54 41-20 49 53 20 46 45 42 20 THE DATA IS FEB
204D:0010 30 33 12 00 00 00 00 00-00 00 00 00 00 00 00 00 03.....
204D:0020 B8 4D 20 8E D8 8E C0 B0-26 8B 0E 12 00 8D 3E 00 8M .X.@0&.....>..
204D:0030 00 FC F2 AE 75 04 4F C6-05 20 B4 4C CD 21 00 00 .|r.u.OF. 4LM!..
204D:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
204D:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
204D:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
204D:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
```

2 下面的程序是比较两个字符串 STRING1 和 STRING2 所含字符是否相同，若相同则显示‘MATCH’，若不相同则显示‘NO MATCH’。该程序是用串比较功能来完成程序所要求的功能。

DATA SEGMENT

```
STRING1 DB 'Visual C++ '
STRING2 DB 'Visual Basic'
COUNT EQU $-STRING2 ;计算串长度
MESS1 DB 'MATCH!',13,10,'$'
MESS2 DB 'NO MATCH!',13,10,'$'
```

```

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA,ES:DATA

MAIN PROC FAR

START:
    PUSH DS
    SUB AX,AX
    PUSH AX
    MOV AX, DATA
    MOV DS, AX
    MOV ES, AX
    ; MAIN PROGRAM
    MOV CX, COUNT
    LEA SI, STRING1
    LEA DI, STRING2
    CLD
    REPZ CMPSB
    JZ MES1
    MOV DX, OFFSET MESS1
    JMP DISP0

MES1: MOV DX, OFFSET MESS2

DISP0: MOV AH,9
    INT 21H

EXIT: RET ;返回 DOS

MAIN ENDP

CODE ENDS

END START

```

运行程序，观察是否得到预期结果，如果没有，**请修改程序得到正确结果**，调试程序，记录调试过程，**并注释程序**。

源代码的问题:

- 当字符串匹配时，REPZ CMPSB 指令会让 ZF 标志位为 1，程序会跳转到 mes1 标签。但在 mes1 标签处，代码加载了 mess2("NO MATCH!")消息，与预期相反。

修改后的代码:

```
DATA SEGMENT
```

```

STRING1 DB 'Visual C++ ' ;定义第一个字符串
STRING2 DB 'Visual Basic' ;定义第二个字符串
COUNT EQU $-STRING2 ;计算串长度,EQU 表示给COUNT 赋值为STRING2 的
长度,伪指令
MESS1 DB 'MATCH! ',13,10,'$' ;定义匹配成功时的提示信息,13->回车,10->换
行.$->字符串结束标志
MESS2 DB 'NO MATCH! ',13,10,'$' ;定义匹配失败时的提示信息
DATA ENDS
CODE SEGMENT
ASSUME CS:CODE,DS:DATA,ES:DATA ;设置段寄存器
MAIN PROC FAR ;远调用,把主程序部分以过程的形式定义,有多个子
程序的时候可以使用上面的方法
START:
    PUSH DS ;压栈,保存寄存器
    SUB AX, AX ;AX=0,清空AX
    PUSH AX ;压栈,保存寄存器
    MOV AX, DATA ;初始化数据段
    MOV DS, AX ;设置DS 指向数据段
    MOV ES, AX ;设置ES 指向数据段
; MAIN PROGRAM
    MOV CX, COUNT ;将字符串长度送入CX
    LEA SI, STRING1 ;将STRING1 的偏移地址送入SI
    LEA DI, STRING2 ;将STRING2 的偏移地址送入DI
    CLD ;清除方向标志位,使串操作按递增方向进行
    REPZ CMPSB ;重复比较两个字符串,直到不相等或CX=0
    JZ MATCH ;如果ZF=1(字符串相等),跳转到MESS1
    MOV DX, OFFSET MESS2 ;否则,将MESS1 的偏移地址送入DX
    JMP DISP0 ;跳转到DISP0
MATCH: MOV DX, OFFSET MESS1 ;将MESS2 的偏移地址送入DX
DISP0: MOV AH,9 ;设置DOS 功能号9(显示字符串)
    INT 21H ;调用DOS 中断
EXIT: RET ;返回DOS,程序结束
MAIN ENDP
CODE ENDS
    END START ;程序结束,从START 开始执行

```

Debug 记录

运行结果

```
(pt2) D:\code\EXPERIMENTAL_REPORT\汇编语言\E5_串处理指令的使用\c2>masm c2.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.
```

```
Object filename [c2.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:
c2.asm(24): error A2105: Expected: instruction or directive
```

```
51502 + 458258 Bytes symbol space free
```

```
0 Warning Errors
```

```
1 Severe Errors
```

```
(pt2) D:\code\EXPERIMENTAL_REPORT\汇编语言\E5_串处理指令的使用\c2>masm c2.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.
```

```
Object filename [c2.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:
```

```
51502 + 458258 Bytes symbol space free
```

```
0 Warning Errors
```

```
0 Severe Errors
```

```
(pt2) D:\code\EXPERIMENTAL_REPORT\汇编语言\E5_串处理指令的使用\c2>link c2.obj
```

```
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.
```

```
Run File [C2.EXE]:
```

```
List File [NUL.MAP]:
```

```
Libraries [LTD]:
```

```
LINK : warning L4021: no stack segment
```

```
(pt2) D:\code\EXPERIMENTAL_REPORT\汇编语言\E5_串处理指令的使用\c2>c2.exe
NO MATCH!
```

```
(pt2) D:\code\EXPERIMENTAL_REPORT>debug 汇编语言\E5_串处理指令的使用\c2\c2.exe
-g
NO MATCH!
```

-U 命令查看指令是否正确导入

```
(pt2) D:\code\EXPERIMENTAL_REPORT>debug 汇编语言\E5_串处理指令的使用\c2\c2.exe
-u
2050:0000 1E          PUSH     DS
2050:0001 2BC0          SUB      AX,AX
2050:0003 50           PUSH     AX
2050:0004 B84D20        MOV      AX,204D
2050:0007 8ED8          MOV      DS,AX
2050:0009 8EC0          MOV      ES,AX
2050:000B B90C00        MOV      CX,000C
2050:000E 8D360000      LEA      SI,[0000]
2050:0012 8D3E0C00      LEA      DI,[000C]
2050:0016 FC          CLD
2050:0017 F3          REPZ
2050:0018 A6          CMPSB
2050:0019 7406          JZ       0021
2050:001B BA1800      MOV      DX,0018
2050:001E EB04          JMP      0024
```

DX 中存储的要显示的信息

```
-r dx
DX 0021
:
-d 0021
204D:0021  4E 4F 20 4D 41 54 43-48 21 0D 0A 24 00 00 00 00  NO MATCH! .$....
204D:0030  1E 2B C0 50 B8 4D 20 8E-D8 8E C0 B9 0C 00 8D 36  .+@P8M .X.@9...6
204D:0040  00 00 8D 3E 0C 00 FC F3-A6 74 06 BA 21 00 EB 04  ...>..|s&t.:!.k.
204D:0050  90 BA 18 00 B4 09 CD 21-CB 00 00 00 00 00 00 00  :...4.M!K.....
204D:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
204D:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
204D:0080  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
204D:0090  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
204D:00A0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  ..
```

三 编写程序

1 假设数据段定义如下：

CONAME DB 'SPACE EXPLORERS INC.'

PRLINE DB 20 DUP('')

STUDENT_NAME DB 30 DUP(?)

STUDENT_ADDR DB 9 DUP('')

PRINT_LINE DB 50 DUP(?)

请完成以下功能：

- (1) 从右到左把 CONAME 中的字符串传送到 PRLINE。使用 debug 命令观察实验结果。
- (2) 把 CONAME 中的第 3 个字节装入 AX。使用 debug 命令观察实验结果。
- (3) 把 STUDENT_NAME 和 STUDENT_ADDR 的内容改成自己的姓名和学号，再把 STUDENT_NAME 移到 PRINT_LINE 的前 30 个字节中，把 STUDENT_ADDR 移到 PRINT_LINE 的后 9 个字节中。使用 debug 命令观察实验结果。

代码：

```
DATA SEGMENT
CONAME DB 'SPACE EXPLORERS INC.'
PRLINE DB 20 DUP(' ')
STUDENT_NAME DB 'TanKe',25 DUP(' ')
STUDENT_ADDR DB '202306630'
PRINT_LINE DB 50 DUP(' '), '$' ; 初始化为空格并添加结束符
DATA ENDS

CODE SEGMENT
```

```

ASSUME CS:CODE, DS:DATA
MAIN PROC FAR
START:
    ; 设置数据段
    PUSH DS
    SUB AX, AX
    PUSH AX
    MOV AX, DATA
    MOV DS, AX
    MOV ES, AX

    ; (1) 从右到左把CONAME 中的字符串传送到PRLINE
    LEA DI, PRLINE+19
    LEA SI, CONAME+19
    MOV CX, 20
    STD                ; 设置方向标志为递减
    REP MOVSB
    CLD                ; 恢复方向标志为递增

    ; (2) 把CONAME 中的第3 个字节装入AX
    MOV AL, CONAME+2
    MOV AH, 0

    ; (3) 把STUDENT_NAME 移到PRINT_LINE 的前30 个字节中
    LEA SI, STUDENT_NAME
    LEA DI, PRINT_LINE
    MOV CX, 30        ; 移动30 个字节
    CLD                ; 确保方向是向前的
    REP MOVSB

    ; 把STUDENT_ADDR 移到PRINT_LINE 的后9 个字节中
    LEA SI, STUDENT_ADDR
    LEA DI, PRINT_LINE+41 ; 50-9=41, 从PRINT_LINE 的第41 个位置开始放
STUDENT_ADDR
    MOV CX, 9          ; STUDENT_ADDR 长度为9
    CLD                ; 确保方向是向前的
    REP MOVSB

    ; 打印PRINT_LINE
    LEA DX, PRINT_LINE
    MOV AH, 9
    INT 21H

EXIT:

```



```

    RET
MAIN  ENDP
CODE  ENDS
END  START

```

Debug 记录

● 目标 1 运行结果

```
-d 204d:0
204D:0000 53 50 41 43 45 20 45 58-50 4C 4F 52 45 52 53 20 SPACE EXPLORERS
204D:0010 49 4E 43 2E 53 50 41 43-45 20 45 58 50 4C 4F 52 INC.SPACE EXPLOR
204D:0020 45 52 53 20 49 4E 43 2E-54 61 6E 4B 65 0D 0A 32 ERS INC.TanKe..2
204D:0030 30 32 33 30 36 36 33 30-20 20 20 20 20 20 20 20 202306630
204D:0040 20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
204D:0050 20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20
204D:0060 20 20 20 20 20 20 20 20-20 20 24 00 00 00 00 00 $. ....
204D:0070 1E 2B C0 50 B8 4D 20 8E-D8 8E C0 8D 3E 27 00 8D .+@P8M .X.@.>'..
```

● 目标 2 运行结果

```
AX=0041 BX=0000 CX=0000 DX=0000 SP=FFF0 BP=0000 SI=FFFF DI=0013
DS=204D ES=204D SS=204D CS=2054 IP=001F NV UP DI PL ZR NA PE NC
2054:001F 8D362800 LEA SI,[0028] DS:0028=6154
-
```

'A'=(65)D=(41)H

● 目标3 运行结果

```
(pt2) D:\code\EXPERIMENTAL_REPORT\汇编语言\E5_串处理指令的使用\c3>debug c3.exe
-g
TanKe
202306630
Program terminated normally

-d 204d:28
204D:0028  54 61 6E 4B 65 20 20 20
204D:0030  20 20 20 20 20 20 20 20 20-20 20 20 20 20 20 20
204D:0040  20 20 20 20 20 20 32 30-32 33 30 36 36 33 30 54
204D:0050  61 6E 4B 65 20 20 20 20-20 20 20 20 20 20 20
204D:0060  20 20 20 20 20 20 20 20-20 20 20 20 20 20 20
204D:0070  20 20 20 20 20 20 20 20-32 30 32 33 30 36 36 33
204D:0080  30 24 00 00 00 00 00 00-00 00 00 00 00 00 00
204D:0090  1E 2B C0 50 B8 4D 20 8E-D8 8E C0 8D 3E 27 00 8D
204D:00A0  36 13 00 B9 14 00 FD F3
```

四、实验报告

要求： 1.每步操作过程及所用指令；2.用截图方式反应指令执行前后的相关寄存器或存储单元内容的变化，并对所得结果进行分析。