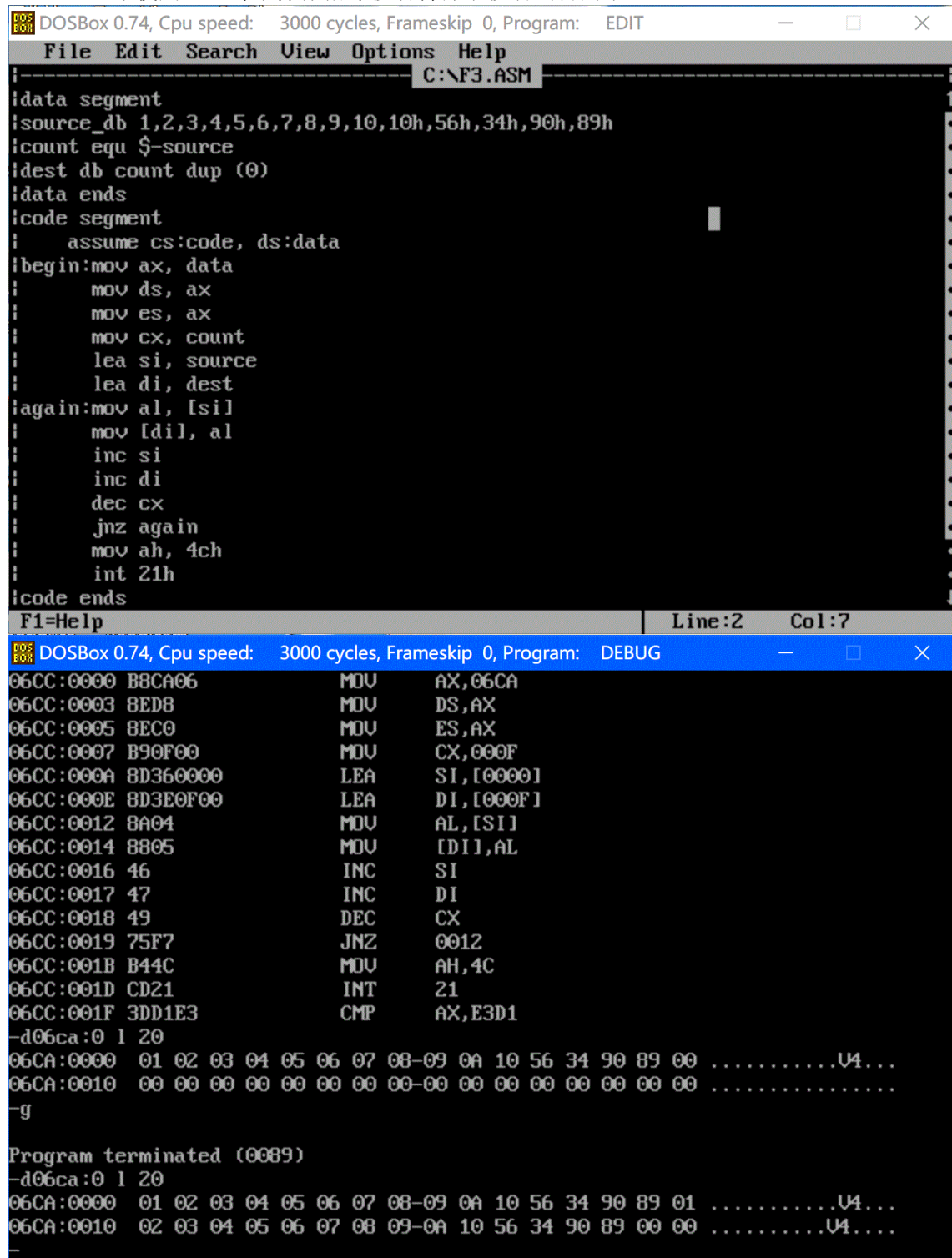


朱志儒-16337341-计算机组成原理（五）

一、（1）不使用movs等串操作指令移动将源串移动到目的串dest

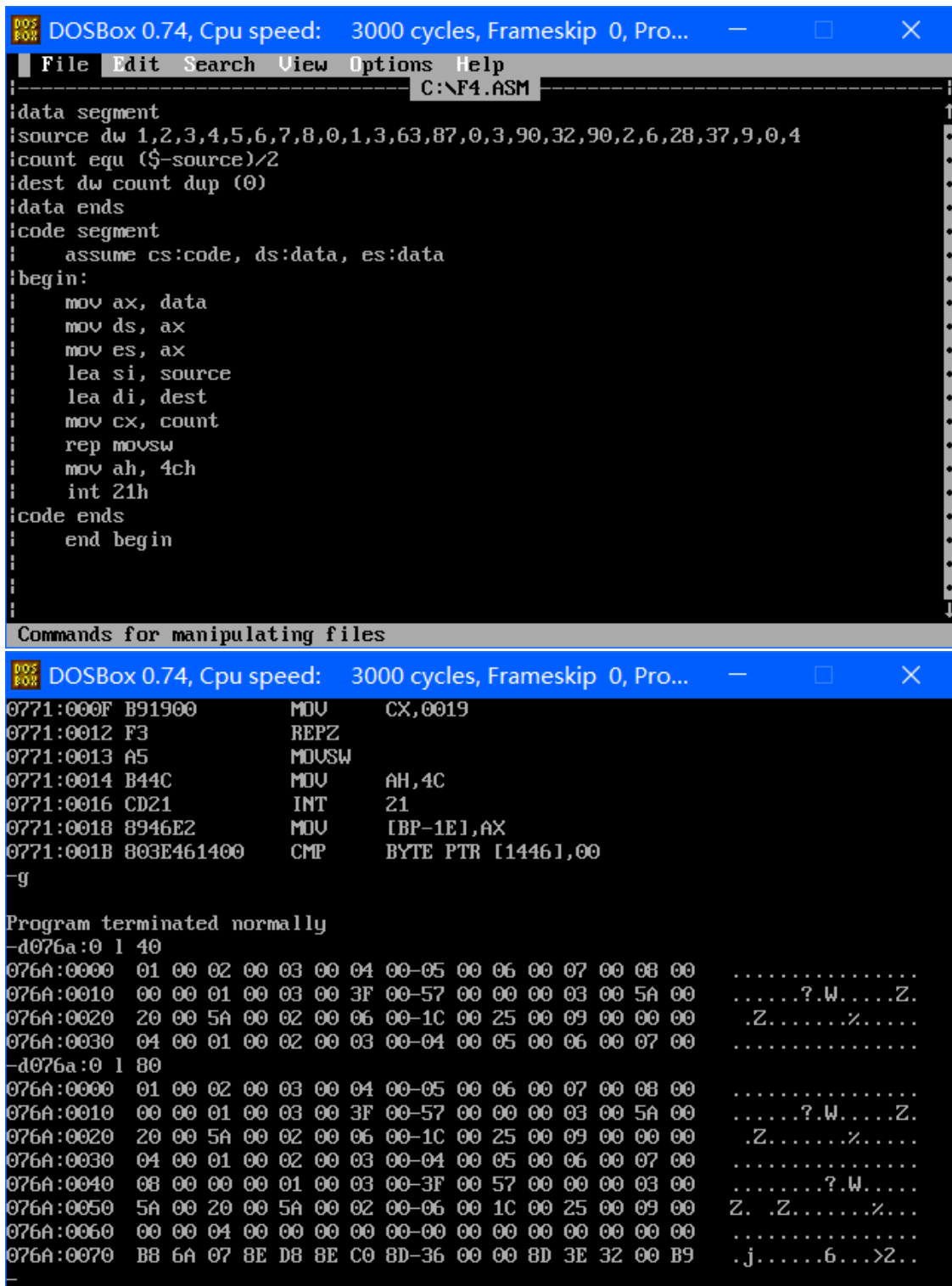


The image shows two windows from DOSBox 0.74. The top window is in EDIT mode, showing an assembly program named C:\NF3.ASM. The program defines a data segment with a source string '1,2,3,4,5,6,7,8,9,10,10h,56h,34h,90h,89h' and a destination buffer 'dest' of size 16. It then sets up registers and moves the source string to the destination. The bottom window is in DEBUG mode, showing the execution of the program. It displays the instruction stream, the state of registers, and the memory contents of the source and destination buffers. The source buffer contains the original string, and the destination buffer contains the copied string.

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT
File Edit Search View Options Help
----- C:\NF3.ASM -----
;data segment
;source_db 1,2,3,4,5,6,7,8,9,10,10h,56h,34h,90h,89h
;count equ $-source
;dest db count dup (0)
;data ends
;code segment
; assume cs:code, ds:data
;begin:mov ax, data
;      mov ds, ax
;      mov es, ax
;      mov cx, count
;      lea si, source
;      lea di, dest
;again:mov al, [si]
;      mov [di], al
;      inc si
;      inc di
;      dec cx
;      jnz again
;      mov ah, 4ch
;      int 21h
;code ends
F1=Help Line:2 Col:7

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
06CC:0000 B8CA06      MOV     AX,06CA
06CC:0003 8ED8          MOV     DS,AX
06CC:0005 BEC0          MOV     ES,AX
06CC:0007 B90F00      MOV     CX,000F
06CC:000A 8D360000      LEA     SI,[0000]
06CC:000E 8D3E0F00      LEA     DI,[000F]
06CC:0012 8A04          MOV     AL,[SI]
06CC:0014 8805          MOV     [DI],AL
06CC:0016 46           INC     SI
06CC:0017 47           INC     DI
06CC:0018 49           DEC     CX
06CC:0019 75F7          JNZ     0012
06CC:001B B44C          MOV     AH,4C
06CC:001D CD21          INT     21
06CC:001F 3DD1E3      CMP     AX,E3D1
-d06ca:0 1 20
06CA:0000 01 02 03 04 05 06 07 08-09 0A 10 56 34 90 89 00 .....U4...
06CA:0010 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
-g
Program terminated (0089)
-d06ca:0 1 20
06CA:0000 01 02 03 04 05 06 07 08-09 0A 10 56 34 90 89 01 .....U4...
06CA:0010 02 03 04 05 06 07 08 09-0A 10 56 34 90 89 00 00 .....U4....
-
```

(2) 使用movs等串操作指令以及附加段将源串移动到目的串dest



The image shows two screenshots of the DOSBox 0.74 interface. The top screenshot displays the assembly code for a program that moves a string from 'source' to 'dest' using the 'rep movsw' instruction. The bottom screenshot shows the execution results, including the disassembled instructions and a memory dump.

Assembly Code (C:\F4.ASM):

```
data segment
source dw 1,2,3,4,5,6,7,8,0,1,3,63,87,0,3,90,32,90,2,6,28,37,9,0,4
count equ ($-source)/2
dest dw count dup (0)
data ends
code segment
assume cs:code, ds:data, es:data
begin:
mov ax, data
mov ds, ax
mov es, ax
lea si, source
lea di, dest
mov cx, count
rep movsw
mov ah, 4ch
int 21h
code ends
end begin
```

Execution Results:

Commands for manipulating files

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro...

0771:000F B91900 MOV CX,0019
0771:0012 F3 REPZ
0771:0013 A5 MOUSW
0771:0014 B44C MOV AH,4C
0771:0016 CD21 INT 21
0771:0018 8946E2 MOV [BP-1E],AX
0771:001B 803E461400 CMP BYTE PTR [1446],00
-g

Program terminated normally

-d076a:0 l 40

Address	Hex	ASCII
076A:0000	01 00 02 00 03 00 04 00-05 00 06 00 07 00 08 00
076A:0010	00 00 01 00 03 00 3F 00-57 00 00 00 03 00 5A 00?.W.....Z.
076A:0020	20 00 5A 00 02 00 06 00-1C 00 25 00 09 00 00 00	.Z.....%.....
076A:0030	04 00 01 00 02 00 03 00-04 00 05 00 06 00 07 00

-d076a:0 l 80

Address	Hex	ASCII
076A:0000	01 00 02 00 03 00 04 00-05 00 06 00 07 00 08 00
076A:0010	00 00 01 00 03 00 3F 00-57 00 00 00 03 00 5A 00?.W.....Z.
076A:0020	20 00 5A 00 02 00 06 00-1C 00 25 00 09 00 00 00	.Z.....%.....
076A:0030	04 00 01 00 02 00 03 00-04 00 05 00 06 00 07 00
076A:0040	08 00 00 00 01 00 03 00-3F 00 57 00 00 00 03 00?.W.....
076A:0050	5A 00 20 00 5A 00 02 00-06 00 1C 00 25 00 09 00	Z. .Z.....%...
076A:0060	00 00 04 00 00 00 00 00-00 00 00 00 00 00 00
076A:0070	B8 6A 07 8E D8 8E C0 BD-36 00 00 8D 3E 32 00 B9	.j.....6...>2..

二、随机输入一串既有大写又有小写的字母串为source，使用lods，stos指令，串中的大写变为小写，同时将源串移动到目的串dest

```
DOS
BOX DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT
File Edit Search View Options Help
----- C:\F7.ASM -----
[
data segment
[source db 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
[count equ $-source
[dest db count dup(0)
[data ends
[icode segment
[assume cs:code, ds:data, es:data
[begin:
[mov ax, data
[mov es, ax
[mov ds, ax
[lea si, source
[lea di, dest
[mov cx, count
[ag:
[ lods source
[ or al, 20h
[ stos dest
[ loop ag
[ mov ah, 4ch
[ int 21h
[icode ends
F1=Help Line:2 Col:1
DOS
BOX DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
C:\>debug f7.exe
-u
06CE:0000 B8CA06 MOV AX,06CA
06CE:0003 BEC0 MOV ES,AX
06CE:0005 BED8 MOV DS,AX
06CE:0007 8D360000 LEA SI,[0000]
06CE:000B 8D3E1A00 LEA DI,[001A]
06CE:000F B91A00 MOV CX,001A
06CE:0012 AC LODSB
06CE:0013 0C20 OR AL,20
06CE:0015 AA STOSB
06CE:0016 E2FA LOOPW 0012
06CE:0018 B44C MOV AH,4C
06CE:001A CD21 INT 21
06CE:001C 188946FC SBB [BX+DI+FC46],CL
-g
Program terminated (007A)
-d 06ca: 0 1 40
06CA:0000 41 42 43 44 45 46 47 48-49 4A 4B 4C 4D 4E 4F 50 ABCDEFGHIJKLMNOP
06CA:0010 51 52 53 54 55 56 57 58-59 5A 61 62 63 64 65 66 QRSTUWXYZabcdef
06CA:0020 67 68 69 6A 6B 6C 6D 6E-6F 70 71 72 73 74 75 76 ghijklmnopqrstuv
06CA:0030 77 78 79 7A 00 00 00 00-00 00 00 00 00 00 00 00 wxyz.....
-q_
```

三、程序的执行过程：

寄存器的变化：(CX) = 1000H, (SP) = 1FEH, (BX) = 3H, (SP) = 1FCH,
IP压栈, (SP) = 1FAH, (SP) = 1F8H, (BP) = 1F8H, (BX) = 3H,
(CX) = 1000H → (CX) = 0, (BX) = 2H, (CX) = 1000H → (CX) = 0,
(BX) = 1H, (CX) = 1000H → (CX) = 0, (BX) = 0H, (BP) = 原BP的值,
(SP) = 200H

5000H:1F8H	BP
5000H:1FAH	IP
5000H:1FCH	03H
	00H
5000H:1FEH	00H
	10H
5000H:200H	