Lecture 12

Data Manipulation: Strings

Text: Chapter 12 (4th edition)
Chapter 11 (5th edition)

A String is a sequence of bytes, usually containing the ASCII codes for printable characters.

They can be defined with the DB directive:

```
        SCHOOL   DB   ’Queens College’
        DEPT     DB   ’Computer Science’
```

Since strings may be quite long, special instructions are provided :

MOVS

Byte, word or doubleword — ES:DI $\leftarrow$ DS:SI

LODS

Load byte, word, doubleword — AL $\leftarrow$ byte
AX $\leftarrow$ word
EAX $\leftarrow$ doubleword

STOS

Store byte, word, doubleword — AL $\rightarrow$ byte
AX $\rightarrow$ word
EAX $\rightarrow$ doubleword

CMPS

Compare byte, word, doubleword using unsigned comparison of ASCII codes.

SCAS

Compare contents of AL, AX or EAX with a memory location.

REP

Repeat prefix, causing a string instruction to be executed multiple times.

Three operands needed

- source
- destination
- length (assumed to be 1 when REP is not used)

```
SomeSchool  DB  14 DUP (?)
ThisSchool  DB  'Queens College'

                LEA DI,SomeSchool
                LEA SI,ThisSchool
                MOVSB
```

Destination:
ES:DI

Source:
DS:SI

To move an entire string, the CX register is loaded with the length of the string and the REP prefix is used.

```
SomeSchool  DB  14 DUP (?)
ThisSchool  DB  'Queens College'

                MOV CX,14
                LEA DI,SomeSchool
                LEA SI,ThisSchool
                CLD
                REP MOVSB
```

Number of bytes to move

Move the bytes left to right

Destination:
ES:DI

Source:
DS:SI

Note that the destination segment is the ES, not the DS. Usually the ES address is encoded by the programmer to be the same as the DS:

```
MOV  AX,DATASEG
MOV  DS,AX
MOV  ES,AX
```

CLD (and STD)

The direction flag indicates whether the string instruction should add or subtract one from the DI and SI registers.

CLD Clear Direction Flag.	0 $\rightarrow$ add
STD Set Direction Flag.	1 $\rightarrow$ subtract

MOVSB Move string byte-by-byte. SI and DI are changed by 1.

MOVSW Move string word-by-word. SI and DI are changed by 2

```
 TITLE  P12MOVST (COM)Use of MOVS string
        .MODEL SMALL
        .CODE
        ORG     100H
BEGIN:  JMP     SHORT MAIN
; ---------------------------------------------------
NAME1   DB      'Assemblers'
NAME2   DB      10 DUP(' ')
NAME3   DB      10 DUP(' ')
; ---------------------------------------------------
MAIN    PROC    NEAR        ;Main procedure
        CALL    C10MVSB     ;MVSB subroutine
        CALL    D10MVSW     ;MVSW subroutine
        MOV     AX,4C00H    ;Exit to DOS
        INT     21H
MAIN    ENDP
;               Use of MOVSB:
C10MVSB PROC    NEAR
        CLD                 ;Left to right
        MOV     CX,10       ;Move 10 bytes,
        LEA     DI,NAME2    ;NAME1 to NAME2
        LEA     SI,NAME1
        REP MOVSB
        RET
C10MVSB ENDP
;               Use of MOVSW:
D10MVSW PROC    NEAR
        CLD                 ;Left to right
        MOV     CX,05       ;Move 5 words,
        LEA     DI,NAME3    ;NAME2 to NAME3
        LEA     SI,NAME2
        REP MOVSW
        RET
D10MVSW ENDP
        END     BEGIN
```

CMPS: Compare String

- Alphanumeric comparison
- Direction flag used
- Compares the operand in (ES:DI) to the operand in (DS:SI) and sets the flag registers.

CMPS without the REP prefix will only compare two bytes.

```
      LARRY  DB     ’Larry’
      CURLY  DB     ’Curly’

             LEA    DI,LARRY
             LEA    SI,CURLY
             CMPSB
             JNE    DIFRNT
             …
```

```
L A R R Y
>
C U R L Y
```

With the REP prefix, it will compare ALL the bytes

```
       LARRY  DB     'Larry'
       CURLY  DB     'Curly'

              MOV    CX,5
              LEA    DI,LARRY
              LEA    SI,CURLY
              REP    CMPSB
              JNE    DIFRNT
              …
```

```
 L A R R Y
 > < = > =   final FLAGS setting
 C U R L Y
```

The JNE instruction will fail (not jump) because the FLAGS indicate that the result of the last comparison done was “equal”.

CONDITIONAL REPETITION

REPE	Repeat while equal. stops comparing when the first non-match is found.
REPNE	Repeat while not equal stops comparing when the first match is found.

Use of REPE (Repeat on Equal):

```
      LARRY  DB     'Larry'
      CURLY  DB     'Curly'

             MOV    CX,5
             LEA    DI,LARRY
             LEA    SI,CURLY
             REPE   CMPSB
             JNE    DIFRNT
             …
```

```
 L A R R Y
>                REPE stops
 C U R L Y


 L A R R Y
> < =            REPNE stops
 C U R L Y
```

SCAS: Scan String

- Look for a specific byte (word or doubleword) contained in the AL (AX, EAX) register.
- Auto increment/decrement of SI,DI depending on direction flag.

```
TITLE   P12SCAST Use of SCAS string operation
        .MODEL   SMALL   ; .COM
        .CODE
        ORG      100H
BEGIN:  JMP      SHORT MAIN
; ----------------------------------------------
NAME1   DB       'Assemblers'          ;Data item
; ----------------------------------------------
MAIN    PROC     NEAR    ;Main procedure
        CLD              ;Left to right
        MOV      AL,'m'
        MOV      CX,10   ;Scan NAME1
        LEA      DI,NAME1 ;for 'm'
        REPNE    SCASB
        JNE      H20     ;If found,
```

Note that at this point, the “m” in “Assemblers” has been found, the CX register contains 5 and the DI register contains the address of the “b” (one past the “m” since it was already increased by the SCASB instruction).

```
H20:
        MOV      AH,4CH
        INT      21H     ;Exit to DOS
MAIN    ENDP
        END      BEGIN
```

Exercises - Lecture 12

1. Write a program where the user is asked to type in two three-letter words. The program should print an appropriate message indicating which word is smaller (alphabetically), such as:

```
CAT is smaller than RAT
PAN is bigger than NAP
POP is the same as POP
```

2. Presuming the data segment contains

```
FUZZY      DB         'peach'
DIMPLED    DB         'orange'
```

and the following code is executed

```
           mov        cx,5
           lea        si,FUZZY
           lea        di,DIMPLED
           cld
```

what will the condition flags indicate if the next instruction is

	greater	less	equal
`cmpsb`			
`rep    cmpsb`			
`repe   cmpsb`			
`repne  cmpsb`			