The String Instructions

Chapter 11

By Rezwana Reaz Rimpi

Modified by (slightly): Mahjabin Nahar

General form of String Instructions

Instruction	Byte form	Word form	Source	Destination
Move String	MOVSB	MOVSW	DS:SI	ES:DI
Load String	LODSB	LODSW	DS:SI	AL or AX
Store String	STOSB	STOSW	AL or AX	ES:DI
Compare String	CMPSB	CMPSW	DS:SI	ES:DI
Scan String	SCASB	SCASW	AL or AX	ES:DI

CLD : Clear Direction FLAG; sets DF = 0;
STD: sets DF = 1;
After execution of each of above instruction SI and DI automatically increased (if CLD is called) or decreased (if STD is called)
For byte instructions SI and DI increased/decreased by 1 byte
For word instructions SI and DI increased/decreased by 2 byte

Copy String1 to String2

```
.data
Str1 db 'Hello'
Str2 db 5 DUP(?)
.code
Main proc
    Mov ax, @data
    Mov ds, ax
    Mov es, ax
    Lea SI, str1
    Lea DI, str2
    CLD
    movsb
    movsb
    movsb
    movsb
    movsb
Main endp
```

DS:SI points to source string

ES:DI points to destination string

You can replace these 5 lines just with 2 lines

```
Mov cx, 5
REP Movsb
```

Equivalent to ⟺

```
Mov cx, 5
Copy:
    Movsb
    Loop copy
```

Copy String1 to String2 in reverse order

```
.data
Str1 db 'Hello'
Str2 db 5 DUP(?)
.code
Main proc
        Mov ax, @data
        Mov ds, ax
        Mov es, ax
        Lea SI, str1 + 4
        Lea DI, str2
        STD
        Mov cx, 5
        Copy:
            Movsb
            Add DI, 2
        Loop copy
Main endp
```

DS: SI points to the last element of str1

ES:DI points to str2

Moves str1[4] to str2[0]
* decrements both SI and DI by 1

We need to decrement SI but increment DI by 1
* add 2 to DI

Use of Lodsb and Stosb

Input str : ABC
Output str : DEF

```
.data
Str1 db 'ABC'
Str2 db 3 DUP(?)
.code
Main proc
        Mov ax, @data
        Mov ds, ax
        Mov es, ax
        Lea SI, str1
        Lea DI, str2
        CLD
```

```
Mov cx, 3
Str_loop:
            lodsb
            add al, 3
            stosb
Loop str_loop

Main endp
End main
```

Moves str1[SI] to AL

Moves AL content to str2[DI]

Conditional REP Instruction

- REPE : Repeat while equal
- REPZ : Repeat while zero

Compare 2 strings: Use of cmpsb, repz, repe

```
.data
Str1 db 'ABC'
Str2 db 'ABD'
.code
Main proc
        Mov ax, @data
        Mov ds, ax
        Mov es, ax
        Lea SI, str1
        Lea DI, str2
        CLD
```

```
Mov cx, 3
Comp_if_equal:
      cmpsb
      jne  Not_equal
      loop Comp_if_equal

Equal :

Not_equal:
Main endp
End main
```

Another way:

```
Mov cx, 3
repe cmpsb
jne  Not_equal

Equal :

Not_equal:

Main endp
End main
```

Automatically repeats cmpsb instruction as long as compared bytes are found equal

Use of scasb

Compares an element pointed to by ES: DI with AL (if scasb is used) or AX (if scasw is used)

```
.data
Str db 'ABC'
.code
Main proc
        Mov ax, @data
        Mov es, ax
        Lea DI, str

        Mov al, 'B'

        CLD
```

```
Mov cx, 3
repne scasb
je  Found


Found:
 ; B is found

Main endp
End main
```

Automatically repeats scasb instruction as long as str[DI] does not match the content in AL