The String Instructions

<u>Reference</u>: Assembly Language Programming and Organization of the IBM PC – Charles Marut – Chapter 11

Overview

- A memory string or string is an array of bytes or words.
- We will see instructions to:
    - Copy a string into another string
    - Search a string for a particular byte or word
    - Store characters in a string
    - Compare strings of characters alphabetically

Moving a String

- MOVSW ; word form of MOVSB
- Copies the contents of the word addressed by DS:SI to the word addressed by ES:DI. The contents of the source word are unchanged.
- After the word has been moved, both SI and DI are automatically incremented by 2 if DF=0, or decremented by 2 if DF=1.

Store String

- STOSB ; store string byte
- Moves the contents of AL register to the byte addressed by ES:DI. DI is incremented if DF=0 or decremented if DF=1.
- STOSW ; store string word
- Moves the contents of AX register to the word addressed by ES:DI. DI is incremented by 2 if DF=0 or decremented if DF=1.

Store String

- MOV AX, @Data
  MOV ES, AX
  LEA DI, string1
  CLD
  MOV AL, ‘A’
  STOSB
  STOSB

	Di ↓				
String1	‘H’	‘E’	‘L’	‘L’	‘O’
After STOSB					
		Di ↓			
String1	‘A’	‘E’	‘L’	‘L’	‘O’
After Second STOSB					
			Di ↓		
String1	‘A’	‘A’	‘L’	‘L’	‘O’

Load String

- LODSB ; load string byte
- Moves the byte addressed by DS:SI into AL. SI is incremented if DF=0 or decremented if DF=1.
- LODSW ; load string word
- Moves the word addressed by DS:SI into AX. SI is increased by 2 if DF=0 or decreased by 2 if DF=1.

Load String

```
.Data
  String1 DB 'ABC$'
.Code
  MOV AX, @Data
  MOV DS, AX
  LEA SI, String1
  CLD
  LODSB
  LODSB
```

Before LODSB					
	SI ↓				
String1	'A'	'B'	'C'		
After LODOSB					
		SI ↓			
String1	'A'	'B'	'C'		AL='A'
After Second LODSB					
			SI ↓		
String1	'A'	'B'	'C'		AL='B'

Scan String

- SCASB ; scan string byte
- Can be used to examine a string for a target byte. The target byte is contained in AL.
- SCASB subtracts the string byte pointed to by ES:DI from the contents of AL and uses the result to set the flags.
- DI is incremented if DF=0 or decremented if DF=1.
- SCASW ; scan string word
- The target word is in AX.

Scan String

```
.Data
 String1 DB 'ABC$'
MOV AX, @Data
MOV ES, AX
LEA DI, String1
CLD
MOV AL, 'B'
SCASB
SCASB
```

Before SCASB						
	Di					
String1	'A'	'B'	'C'		AL='B'	
After SCASB						
		DI				
String1	'A'	'B'	'C'		AL='B'	ZF=0
After Second SCASB						
			DI			
String1	'A'	'B'	'C'		AL='B'	ZF=1

Scan String

- If CX is initialized to the number of bytes in the string

REPNE SCASB ; Repeat SCASB while not equal to target

- Repeatedly subtract each string byte from AL, update DI and decrements CX until the target is found or CX=0.

Compare String

- CMPSB ; compare string bytes
- Subtracts the byte with address ES:DI from the byte with address DS:SI and sets the flags. Then both SI and DI are incremented or decremented depending on DF.
- CMPSW ; compare string words

Compare String

```
.Data
String1 DB ‘ACD$’
String2 DB ‘ABC’
.Code
MOV AX, @Data
MOV DS, Ax
MOV ES, Ax
LEA  SI,  String1
LEA DI, String2
CLD
CMPSB
CMPSB
```

Before CMPSB					
	SI ↓				
String1	‘A’	‘B’	‘C’		
	DI ↓				
String2	‘A’	‘C’	‘D’		
After CMPSB					
		SI ↓			
String1	‘A’	‘B’	‘C’		
		DI			ZF=1
String2	‘A’	‘C’	‘D’		
After CMPSB					
			SI ↓		
String1	‘A’	‘B’	‘C’		
			DI ↓		ZF=0
String2	‘A’	‘C’	‘D’		

Thanks.......