

Entendendo LODS, STOS e MOVS

Sejam os seguintes valores de uma memória (vetor):

Posição de Memória	06	07	08	09	10	11	12
Valor	50	32	38	03	23	01	12

Executando as instruções

MOV SI,07 ; SI vem Source Index

LODSB ; AL = 32 ou AL=Memoria[7]

Assim, se SI apontava para 07 após uma instrução LODSB, AL seria agora igual a **32**.

Executando as instruções

MOV DI, 11 ;DI vem Destin Index

MOV AL,50

STOSB ;Memoria[11]=50

Movemos o valor que estava em AL para a posição 11 da memória apontada por DI

Posição de Memória	06	07	08	09	10	11	12
Valor	50	32	38	03	23	50	12

OBS.: Quando usamos LODSB/STOSB, usamos AL. Isto porque estaremos mexendo com um número de **8** bits (um byte), apenas. Podemos armazenar um número de 8 bits em AL, AH, ou AX, mas não podemos armazenar um número de **16** bits em AH ou AL porque eles são **REGISTRADORES DE 8 BITS**.

Como resultado, quando usarmos LODSW ou STOSW, nós devemos usar AX e não AL, já que estaremos pegando/colocando um número de **16** bits.

- MOVSB – Move um byte
- MOVSW – Move uma word

Como exemplo, vamos pegar um byte de um vetor onde SI aponta e mandá-lo para um vetor onde DI aponta.

Memoria1 apontada por SI:

Posição de Memória	06	07	08	09	10	11	12
Valor	50	32	38	03	23	50	12

Memoria2 apontada por DI:

Posição de Memória	06	07	08	09	10	11	12
Valor	10	11	20	02	67	00	12

MOV SI,07
MOV DI,11
MOVSB ; Memoria2[DI]=Memoria1[SI]

Em DI:

Posição de Memória	06	07	08	09	10	11	12
Valor	10	11	20	02	67	32	12

- See more at: <http://www.krull.com.br/programacao/programacao-assembly/asm-tutorial-1-adam-hyde/#sthash.Vh3ZkYp0.dpuf>