ASSEMBLY XV

Continuando o papo sobre o TASM, precisaremos aprender como manipular tipos de dados mais complexos do que WORD, BYTE ou DWORD. Eis a descrição das estruturas!

Uma estrutura é o agrupamento de tipos de dados simples em uma única classe de armazenamento, por exemplo:

```
STRUC   MyType
    A   DB  ?
    B   DW  ?
ENDS
```

A estrutura MyType acima, delimitada pelas palavras-chava STRUC e ENDS, foi construida com dois tipos de dados simples (BYTE e WORD) com os nomes de A e B. Note que as linhas acima apenas declaram a estrutura, sem alocar espaço na memória para ela. Criar uma 'instancia' dessa estrutura é tão simples quanto criar uma variável de tipo simples:

```
MyVar   MyType  <0,0>
```

A sintaxe é basicamente a mesma de qualquer declaração de variável em assembly, com a diferença de que o 'tipo' do dado é o nome (ou TAG) da estrutura - MyType - e os dados iniciais dos elementos da estrutura estão localizados entre os simbolos < e >. Na linha acima criamos a variável MyVar, cujos elementos são 0 e 0. Vamos a um exemplo de uso desse novo tipo:

```
;... Aqui entra o modelamento,...

DATASEG

MyVar   MyType  <0,0>

CODESEG

PROC    SetA        ; Poe valor em A na estrutura.
ARG     V : Byte
        mov     al,[V]
        mov     [MyVar.A],al
        ret
ENDP

PROC    SetB        ; Poe valor em B na estrutura.
ARG     V : Word
        mov     ax,[V]
        mov     [MyVar.B],ax
        ret
ENDP

;... Aqui entra o fim do código...
```

Simples, näo?

Mas, e se quisermos trabalhar com um vetor do tipo MyType? Vetores de tipos mais simples é facil:

```
DATASEG

MyVar1  dw  10 DUP (0)

CODESEG

PROC    Fill1
    mov     cx,10
    sub     bx,bx
@@FillType1:
    mov     [bx+MyVar1],0FFh
    add     bx,2
    dec     cx
    jnz     @@FillType1
    ret
ENDP
```

Aqui fiz da maneira mais dificil apenas para exemplificar um método de preenchimento de vetores. No caso, BX contém o item desejado do vetor. MyVar1 é o deslocamento do primeiro item do vetor na memória e CX a quantidade de itens do vetor. Note que temos um vetor de WORDS e precisaremos adicionar 2 (tamnho de uma WORD) para cara item do vetor. No caso da estrutura, isso fica um pouco mais complicado porque ela pode ter um tamanho näo múltiplo de 2 (o que complica o cálculo. Por exemplo, MyType (a estrutura) tem 3 bytes de tamanho. Eis a implementação (näo otimizada) para a rotina FillType para preenchimento de um vetor de MyType com 10 itens:

```
DATASEG
MyVar   MyType  10 dup (<0,0>)

CODESEG
PROC    FillType
        mov     cx,10
        sub     bx,bx   ; indice para localizar itens.
@@FillLoop:
        mov     [bx+MyVar.A],0FFh   ; * Instrução destacada...
        mov     [bx+MyVar.B],0FFFFh
        add     bx,3
        dec     cx
        jnz     @@FillLoop
        ret
ENDP
```

Essa rotina merece ser observada mais de perto:

Vejamos a instrução destacada na listagem acima... MyVar.A fornece o deslocamento de A, do primeiro item do vetor, na memória, enquanto isso BX fornece o indice do item desejado no vetor. Assim, BX+MyVar.A fornecerá o offset do elemento A do item da estrutura desejado.

Well... É isso...