

Por: Frederico Pissarra

Mais instruções de comparação...

| CMPSB e CMPSW

Essas instruções comparam (da mesma forma que CMP) o conteúdo da memória apontada por DS:SI com o conteúdo apontado por ES:DI, afetando os flags. Com isso, soluciona-se a limitação da instrução CMP com relação aos dois operandos como referências à memória!

Lembre-se que DS:SI é o operando implícito FONTE, enquanto ES:DI é o destino. A comparação é feita de ES:DI para DS:SI. A rotina abaixo é equivalente a CMPSB:

```

+-----+
| MOV AL,ES:[DI]
| CMP AL,[SI]
| INC SI
| INC DI
+-----+

```

Existe um pequenino erro de lógica na rotina acima, mas serve aos nossos propósitos de ilustrar o que ocorre em CMPSB.

SI e DI serão incrementados (ou decrementados, depende do flag de direção) depois da operação, e o incremento (ou decremento) dependerá da instrução... Lembre-se que CMPSB compara Bytes e CMPSW compara Words.

| SCASB e SCASW

Essas instruções servem para comparar (da mesma forma que CMP o faz) o conteúdo da memória apontado por DS:SI com o registrador AL (no caso de SCASB) ou AX (no caso de SCASW). Os flags são afetados e SI é incrementado (ou decrementado) de acordo com a instrução usada.

```

+-----+
|Comparando blocos de memória:
+-----+

```

Podemos usar CMPS? e SCAS? (onde ? e' B ou w) em conjunto com REP para compararmos blocos (CMPS?) ou procurar por um determinado dado num bloco (SCAS?). A diferença aqui é que podemos fornecer uma condição de comparação ou busca.

Acrescentando o modificador REP, precisamos dizer à uma dessas instruções a quantidade de dados que queremos manipular... fazemos isso através do registrador CX (assim como fizemos com LODS? e STOS?):

```

;Certifica-se do sentido crescente!
CLD

;Obtém o segmento da linha de comando e coloca em DS
MOV  AX,SEG LINHA_DE_COMANDO
MOV  DS,AX

;Obtém o offset inicial da linha de comando
MOV  SI,OFFSET LINHA_DE_COMANDO

;Procura, no máximo por 128 bytes
MOV  CX,128

;Procuraremos por um espaço.
MOV  AL,' '

REPNE SCASB

```

Esse fragmento de código ilustra o uso de SCASB com blocos. O modificador REPNE significa (REPete while Not Equal - Repete enquanto não for igual). REPNE garante que o byte vai ser procurado por toda a linha de comando até que o primeiro espaço seja encontrado. Se não houver espaços na linha, então, depois de 128 bytes de procura, o registrador CX estará zerado (já que é decrementado a cada byte comparado).

Esta é outra característica das instruções que manipulam blocos (as que são precedidas de REP, REPNE ou REPE): O contador é decrementado a cada operação da instrução associada (no nosso caso SCASB), bem como os demais operandos implícitos (SI no caso acima) é incrementado a cada passo.

Se quisermos encontrar o primeiro byte DIFERENTE de espaço na rotina acima, basta trocar REPNE por REPE (Repete enquanto for IGUAL).

REPE e REPNE não foram mencionados antes porque não funcionam com LODS? e STOS?.

□