

MATA49

Programação de Software Básico

Comparação de Dados

Leandro Andrade
leandrojsa@ufba.br

Comparação de Dados

- CMP
 - A comparação é feita através da subtração entre os dois operandos, porém o resultado **não é armazenado** em nenhum registrador
 - Um dos operandos deve ser um registrador
 - Usado para comparação de inteiros e caracteres
 - Sintaxe: CMP destino, fonte

Comparação de Dados

- CMP
 - Resultado = destino – fonte
 - Resultados modificam os bits do registrador FLAGS
 - Overflow (OF), Sign (SF), Zero (ZF), Carry (CF)
 - A mudança nos flags variam se estamos tratando de números com ou sem sinal

Comparação de Dados

- **CMP**
 - Comparação de números sem sinal
 - Zero Flag e Carry Flag

CMP Results	ZF	CF
Destination < source	0	1
Destination > source	0	0
Destination = source	1	0

Comparação de Dados

- CMP
 - Compara de número com sinal
 - Altera os flags Sign, Overflow e Zero

CMP Results	Flags
Destination < source	SF \neq OF
Destination > source	SF = OF
Destination = source	ZF = 1

vc Notas & Assinamentos via

Comparação de Dados

- CMP

- Exemplos:

```
MOV    AX, 5
```

```
CMP    AX, 10    ; ZF = 0 and CF = 1
```

```
MOV    AX, 1000
```

```
MOV    CX, 1000
```

```
CMP    CX, AX    ;ZF = 1 and CF = 0
```

Comparação de Dados

- CMP

- Exemplos:

```
MOV    EAX, 30
```

```
CMP    EAX, -40    ; SF == OF
```

```
MOV    EBX, -10
```

```
CMP    EBX, 10     ; SF != OF
```

Saltos Condicionais e Incondicionais

Saltos Incondicionais

- JMP
 - Realiza um salto para uma instrução com um nome (label) indicado
 - Realiza saltos para instruções após e antes do JMP
 - Sintaxe:
 - JMP <Destino>

Saltos Incondicionais

- JMP

- Exemplo:

- MOV EAX, 0

- Incremento:

- INC EAX

- JMP Incremento ;Loop infinito

Saltos Condicionais

- Podemos condicionar os saltos aos resultados valores dos bits do FLAGS
- Podemos condicionar os saltos de acordo com comparações entre inteiros (CMP)
- Comandos diferentes para avaliação de flags diferentes
- Sintaxe Geral:
 - J<cond> <destino>

Saltos Condicionais

- Condicionais Simples
 - Análise a partir dos flags:

Mnemonic	Description	Flags / Registers
JZ	Jump if zero	ZF = 1
JNZ	Jump if not zero	ZF = 0
JC	Jump if carry	CF = 1
JNC	Jump if not carry	CF = 0
JO	Jump if overflow	OF = 1
JNO	Jump if not overflow	OF = 0
JS	Jump if signed	SF = 1
JNS	Jump if not signed	SF = 0
JP	Jump if parity (even)	PF = 1
JNP	Jump if not parity (odd)	PF = 0

Saltos Condicionais

- Condicionais Simples

- Exemplo:

```
if ( EAX == 0 )  
    EBX = 1;  
else  
    EBX = 2;
```

could be written in assembly as:

```
1      cmp     eax, 0           ; set flags (ZF set if eax - 0 = 0)  
2      jz      thenblock       ; if ZF is set branch to thenblock  
3      mov     ebx, 2           ; ELSE part of IF  
4      jmp     next            ; jump over THEN part of IF  
5 thenblock:  
6      mov     ebx, 1           ; THEN part of IF  
7 next:
```

Saltos Condicionais

- Condicionais Simples:

- Exemplo:

- if (EAX >= 5)

- EBX = 1;

- else

- EBX = 2;

Saltos Condicionais

- Exemplo:

```
    cmp     eax, 5
    js      signon          ; goto signon if SF = 1
    jo      elseblock       ; goto elseblock if OF = 1 and SF = 0
    jmp     thenblock       ; goto thenblock if SF = 0 and OF = 0
signon:
    jo      thenblock       ; goto thenblock if SF = 1 and OF = 1
elseblock:
    mov     ebx, 2
    jmp     next
thenblock:
    mov     ebx, 1
next:
```

Saltos Condicionais

- Condicionais de comparação
 - Avaliação a partir da instrução
 - `CMP <vleft>, <vright>`

Signed		Unsigned	
JE	branches if <code>vleft = vright</code>	JE	branches if <code>vleft = vright</code>
JNE	branches if <code>vleft ≠ vright</code>	JNE	branches if <code>vleft ≠ vright</code>
JL, JNGE	branches if <code>vleft < vright</code>	JB, JNAE	branches if <code>vleft < vright</code>
JLE, JNG	branches if <code>vleft ≤ vright</code>	JBE, JNA	branches if <code>vleft ≤ vright</code>
JG, JNLE	branches if <code>vleft > vright</code>	JA, JNBE	branches if <code>vleft > vright</code>
JGE, JNL	branches if <code>vleft ≥ vright</code>	JAE, JNB	branches if <code>vleft ≥ vright</code>

Saltos Condicionais

- Condicionais de comparação
 - O Exemplo anterior pode ser expressado como:

```
1          cmp     eax, 5
2          jge     thenblock
3          mov     ebx, 2
4          jmp     next
5  thenblock:
6          mov     ebx, 1
7  next:
```

Saltos Condicionais

- Exercício:
 - Faça um código em assembly equivalente ao código em C abaixo:

```
if(eax >= ebx){  
    eax = 1;  
}else if(eax == 0){  
    eax = -1;  
}
```

Instrução LOOP

- Repete um bloco de instruções um número específico de vezes
- O número de vezes é contabilizado pelo valor numérico do registrador ECX
 - A cada iteração o ECX é decrementado em 1 e verifica-se se novo valor de ECX é zero.
 - Caso seja diferente de zero é feito um salto para início do bloco
- Sintaxe: LOOP <destino>

Instrução LOOP

- Exemplo:

```
mov    ax,0
```

```
mov    ecx,5
```

```
L1:
```

```
inc    ax
```

```
loop   L1
```

Instrução LOOPZ e LOOPE

- São idênticas (possuem o mesmo opcode)
- Semelhantes ao LOOP, porém além de verificar se $ECX > 0$, observam se Zero Flag (ZF) é igual a 1
- Sintaxe: LOOPZ/LOOPE <destino>
 $ECX = ECX - 1$
if $ECX > 0$ and **ZF == 1**, jump to destination

Instrução LOOPNZ e LOOPNE

- Segue a mesma lógica do LOOPZ/LOOPE, só que Zero Flag deve ser 0

$ECX = ECX - 1$

if $ECX > 0$ and $ZF \neq 0$, jump to destination

Instrução LOOP

- Exercício:
 - Faça um código em assembly equivalente ao código em C abaixo:

```
do{
    EAX++;
    if (EAX == ECX){
        EDX = 10;
    }else{
        EDX = 20;
    }
}while (EAX > EBX)
```