

METODI DI INDIRIZZAMENTO DI UN DATO

È il modo in cui l'istruzione ci dice dove reperire gli operandi

Esistono 7 modi diversi di indirizzamento:

1. Immediato
2. Indirizzamento a registro
3. Indirizzamento diretto
4. Indirizzamento indiretto
5. Indirizzamento indiretto con spiazzamento
6. Indirizzamento indiretto indicizzato
7. Indirizzamento indiretto indicizzato con spiazzamento

Nei nostri esempi abbiamo utilizzato i primi tre.

1. Indirizzamento immediato

In questo metodo il valore dell'operando è già indicato nell'istruzione stessa.

Ad esempio.

- Add num1, F5 ; somma al contenuto della locazione di memoria num1 il valore immediato F5
- Mov ax, 0FFFFh ; carica in ax il valore immediato FFFFh

↑
NB. ogni numero esadecimale che inizia con una lettera deve essere preceduto dallo 0

2. Indirizzamento a registro

Gli operandi delle istruzioni sono dei registri.

Ad esempio: mov ax,bx

3. Indirizzo diretto

Uno degli operandi è l'indirizzo di memoria a cui è destinato il dato oppure l'indirizzo di memoria da cui prelevare.

Ad esempio:

- mov al, num 1 dove num1 è una variabile ed è l'offset dell'operando in memoria centrale
eseguendo l'istruzione con TD num 1 viene visualizzata con **[indirizzo]**

4. Indirizzamento indiretto

Quando l'offset si trova in uno di seguenti registri base o indice: **BX, BP, SI e DI**.

L'accesso in memoria avviene attraverso uno di questi registri e quindi questo metodo risulta più veloce in quanto l'offset è già in CPU.

Non si possono usare gli altri registri, ad esempio AX.

Esempio. Scrivere un programma in assembly che stampi a video un carattere posto in memoria, il cui indirizzo è nel registro BX.

<pre>.MODEL SMALL .STACK .DATA Carattere DB 'S' .CODE Inizio: MOV DX,@data MOV DS,DX ;DS è impostato per puntare al DATA ; SEGMENT LEA BX,Carattere ;BX punta a Carattere ; equivale a ;mov bx, offset carattere MOV DL,DS:[BX] ; oppure mov DL, [BX] ;stampa al carattere a video MOV AH,02h INT 21h ;Visualizzazione di Carattere mov ah,4Ch ;Terminazione e rientro al DOS int 21h END Inizio</pre>	<pre>.MODEL SMALL .STACK .DATA Carattere DB 'S' .CODE Inizio: MOV DX,@data MOV ES,DX LEA BX,Carattere ;BX punta a Carattere MOV DL,ES:[BX] ;stampa a video il carattere MOV AH,02h INT 21h ;Visualizzazione di Carattere mov ah,4Ch ;Terminazione e rientro al DOS int 21h END Inizio</pre>
---	---

ISTRUZIONE LEA

Sintassi: **LEA** operando 1, operando2

Carica in operando 1 l'offset di operando 2. E' equivalente a : **MOV** operando1, **offset** operando2

5. Indirizzamento indiretto con spiazzamento

Con spiazzamento si intende una costante a 8 o 16 bit che la CPU somma al contenuto dei registri base o indice. Utile per lavorare su vettori in memoria centrale.

;un esempio di indirizzamento indiretto con spiazzamento e vettore stringa
;L'indirizzo effettivo dell'operando viene determinato
;sommando uno spiazzamento al contenuto di un registro base.

.model small

.stack

.data

vet db 'ciao mondo' ; in memoria: 63 69 61 6F 20 6D 6F 6E 64 6F

.code

inizio:

mov ax,@data

mov ds,ax

;Imposta AL con il terzo carattere del messaggio (elemento stringa con indice 2)

lea bx,vet

mov al,[bx+2] ; al viene caricato 61

;stampo a video

MOV DL, al ;memorizzo il carattere da stampare in DL cioè a

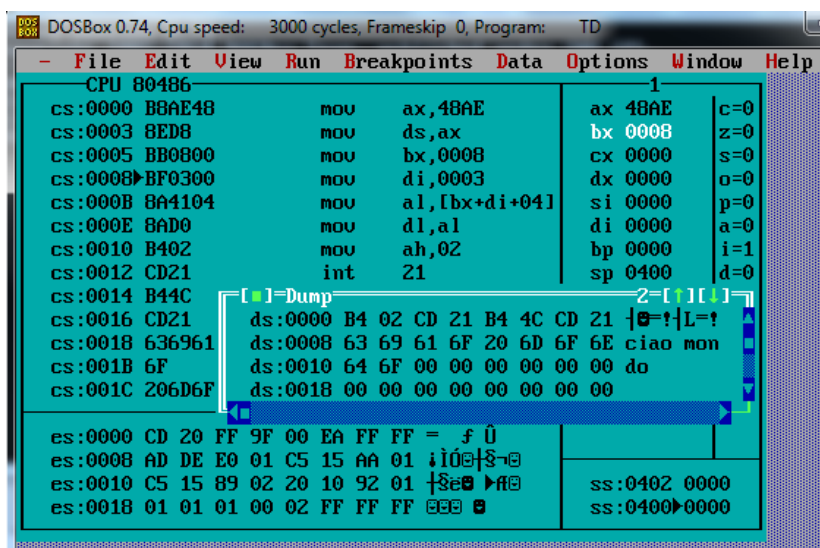
MOV AH, 02h ; stampa a video il carattere in DL

INT 21h

mov ah,4ch

int 21h

end inizio



6. Indirizzamento indiretto indicizzato

L'offset viene calcolato sommando il contenuto di un registro base BX o BP con il contenuto di un registro indice SI o DI. Molto utile per lavorare su vettori in memoria centrale.

```
.model small
.stack
.data
    vet db 0,2,3,4,8

.code
inizio:
    mov ax,@data
    mov ds,ax

    lea bx,vet
    mov si,2
    mov ax,[bx+si]

    mov ah,4ch
    int 21h
end inizio
```

7. Indirizzamento indiretto indicizzato con spiazzamento

L'offset viene calcolato sommando il contenuto di un registro base BX o BP con il contenuto di un registro indice SI o DI e con una costante.

```
.model small
.stack
.data
    vet db 'ciao mondo'      ; in memoria: 63 69 61 6F 20 6D 6F 6E
.code
inizio:
    mov ax,@data
    mov ds,ax

    ;Imposta AL con l'ottavo carattere del messaggio (di indice 7=3+4)
    lea bx,vet
    mov di,3
    mov al,[bx+di+4]        ; al viene caricato 6D

    ;stampo a video
    MOV DL, al              ;memorizzo il carattere da stampare in DL cioè n
    MOV AH, 02h             ; stampa a video il carattere in DL
    INT 21h

    mov ah,4ch
    int 21h
end inizio
```

Esercizio1. Scrivere i dati 1234h e 12h, rispettivamente nei registri AX e AL con il metodo indiretto.

<pre>;indirizzamento indiretto .model small .stack .data wDato dw 01234H bDato db 012H .code inizio: mov ax,@data mov ds,ax mov bx, offset wDato mov ax,[bx] mov bx, offset bDato mov al,[bx] mov ah,4ch int 21h end inizio</pre>	<pre>;indirizzamento indiretto .model small .stack .data wDato dw 01234H bDato db 012H .code inizio: mov ax,@data mov ds,ax lea bx,wDato mov ax,[bx] lea bx,bDato mov al,[bx] mov ah,4ch int 21h end inizio</pre>
---	---

- Osservare come vengono memorizzato la variabile word in memoria[34 12]

Esercizio2. Completa il seguente programma seguendo i commenti in verde.

<pre>.model small .stack .data vet db 2,4,6,8,10,12,14,16,18,20 .code inizio: mov ax,@data mov ds,ax ;Imposta AL con il primo elemento (=2)del vettore con il metodo indiretto lea bx,vet mov al,[...] ;Imposta AL con il terzo elemento (=6) del vettore con metodo indiretto con spiazzamento mov al,[...] ;Imposta AL con il sesto elemento (=12) del vettore con metodo indiretto indicizzato mov si,... mov al,[bx+si] ;Imposta AL con l'ultimo elemento (=20) del vettore con metodo indicizzato con spiazzamento mov al,[... + si + ...] mov ah,4ch int 21h end inizio</pre>
