

27. Esercitazione Assembly

mercoledì 10 giugno 2020 17:09

Architettura Intel x86-32

8 Registri generali:

EAX:

EBX:

ECX:

EDX:

ESI:

EDI:

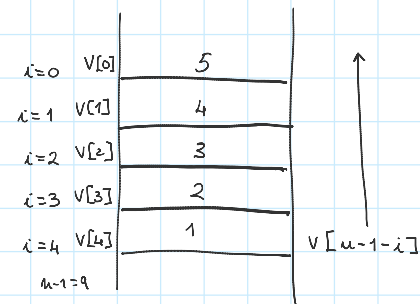
EBP:

ESP:

Memoria:

Indirizzo	Valore	8bit
$2^{32} - 1$	FFFFFFFFh	←→
...
1373	0000055Dh	10h
1372	0000055Ch	21h
1371	0000055Bh	32h
1370	0000055Ah	45h
...
1	00000001h	
0	00000000h	

Numeri rappresentati in formato Little Endian (vedi esempio 10213245h).



ESERCIZIO 1

Scrivere una funzione Assembly che, dato un vettore V di word (16 bit) di dimensione n, conta il numero di terne elementari per le quali vale la proprietà:

- V[i] è pari allora deve essere minore della somma dei due numeri successivi
- V[i] è dispari allora deve essere maggiore della differenza fra il successivo e il numero dopo

```
%include "utils.nasm"
section .data
    V dw 3,4,-2,8,1,5,6
    n equ ($-V)/2 ; la lunghezza di un vettore o di una stringa si ottiene con ($-variabile)/bitDati
section .bss
    c resw 1 ; questa è una variabile dove salverò il numero di terne. La sua grandezza è arbitraria se non specificata dalla traccia

section .text
    global _start
_start:
    ; conviene operare sui registri per poi spostare i valori ottenuti nella variabile creata solo al termine
    ; poiché comunicare sempre con la memoria è costosa
    XOR ESI,ESI ; azzeramento del registro che useremo come traccia dei passaggi
    XOR DI,DI ; azzero il registro contatore del numero di terne. Poiché il numero sarà a 16 bit posso usare solo DI anziché EDI
    MOV EDX,n
    SUB EDX,2 ; con questi due passaggi ottengo in EDX il numero di terne da analizzare
ciclo:
    CMP ESI,EDX ; se ESI>=EDX (terne analizzate>=terne da analizzare)
    JGE fine ; se il primo è maggiore del secondo salta e fine
    MOV AX,[V+ESI*2] ; sposto il valore V[i] in AX (poiché a 16 bit). V rappresenta l'indirizzo del primo elemento.
    ; la notazione indica indirizzoPrimoElemento+elementoDellArray*numeroBitElemento
    ; tale notazione serve per avere l'accesso ad un elemento il cui indirizzo dipenderà dal numero di
    ; operazioni fatte
    MOV BX,[V+ESI*2+2] ; lettura di V[i+1]
    ; la notazione serve per indicare:
    ; indirizzoPrimoElemento+elementoDellArray*numeroBitElemento+numeroDiLocazioniRispettoAlValorePrecedente
    ROR AX,1 ; opero lo shift così il bit eliminato va a finire nel Carry Flag che posso controllare in una verifica con JC (CF==1) o JNC(CF==0)
    JNC pari ; se CF==0 allora il numero è pari, altrimenti se continua è dispari
    SUB BX,[V+ESI*2+4] ; sottraggo il valore di V[i+2] al registro che conteneva già V[i+1]
    ROL AX,1
    CMP AX,BX
    JLE avanti
    JLE avanti
    inc:
    INC DI ; incremento il registro che ha il numero di terne
    avanti:
    INC ESI
    JMP ciclo
pari:
    ADD BX,[V+ESI*2+4] ; sommo il valore di V[i+2] al registro che conteneva già V[i+1]
    ROL AX,1
    CMP AX,BX ; verifico se V[i]<V[i+1]+V[i+2]
    JGE avanti
    JMP inc
fine:
    MOV [c],DI
    printw word [c] ; bisogna specificare il valore della memoria
    exit 0
```

1370	3	
1371		
1372	4	V[1]
1373		
1374	-2	V[2]
1375		
1376	8	V[3]
1377		
1378	1	V[4]
1379		
1380	5	V[5]
1381		
1382	6	V[6]
1383		

|GAIA BERTOLINO|
INGEGNERIA INFORMATICA