

## 27. Esercitazione Assembly

mercoledì 10 giugno 2020 17:09

**Architettura Intel x86-32**

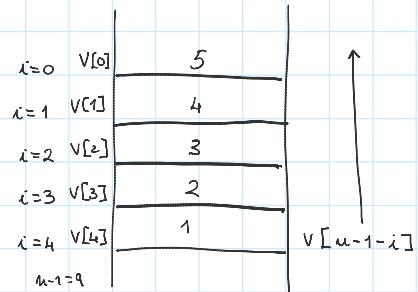
**8 Registri generali:**

EAX:	31	15	7	AH A X AL
EBX:	31	15	7	BH B X BL
ECX:	31	15	7	CH C X CL
EDX:	31	15	7	DH D X DL
ESI:	31	15	7	SI
EDI:	31	15	7	DI
EBP:	31	15	7	BP
ESP:	31	15	7	SP

**Memoria:**

Indirizzo	Valore
$2^{32} - 1$	FFFFFFFFFFh
...	...
1373	0000055Dh
1372	0000055Ch
1371	0000055Bh
1370	0000055Ah
...	...
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; azzeramento 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+1]; 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
incr:
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 incr
fine:
MOV [c], DI
printw word [c] ; bisogna specificare il valore della memoria
exit 0
```

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

