

# Assembly

Friday, 25 August 2023

17:52

## 1) Esercizio 4 (ciclo for)

Dato il frammento di codice assembly

```
.data
numeri:    .word 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
risultato:  .word 0
```

scrivere un programma assembly che calcoli la somma di 10 numeri disponibili in memoria all'etichetta "numeri"

```
.data
numeroLoop: .word 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
risultato:   .word 0
lunghezza:   .word 10
.text
main:
    li $t1, 10 # Numero totale
    li $t2, 0 # Indice attuale
    lw $t7, lunghezza
    mul $t7, $t7, 4

loop:
    lw $t3, risultato # Prendo il risultato
    lw $t4, numeroLoop($t2) # Prendo ciò che devo sommare
    addi $t2, $t2, 4 # Incremento per dopo
    add $t3, $t3, $t4 # Sommo il risultato
    sw $t3, risultato # Rimetto il risultato nello stack
    bne $t7, $t2, loop # Se siamo al limite del for, vai alla fine

fine:
    li $v0, 1 # Scrivi il risultato nello schermo
    lw $a0, risultato
    syscall
```

## 2) Scrivere un programma assembly che legge 3 valori che si trovano in memoria all'etichetta "valori" e memorizza in memoria, nella locazione subito dopo il terzo valore:

- il risultato della somma dei tre valori, se il primo valore è positivo
- il prodotto dei tre numeri, se il primo valore è negativo
- l'AND del secondo e del terzo numero, se il primo è uguale a zero

```
.data
valori: .word 1, 2, 3, 10
.text
main:
    li $t1, 8 # Prendo il terzo valore
    lw $t0, valori($t1)
    li $t1, 12 # Prendo l'indirizzo del 4 valore che sarà il nostro output
    la $s0, valori($t1)
    lw $s1, valori($zero) # Prendo il primo valore
```

```

        li $t1, 4 # Il secondo
        lw $s2, valori($t1)
        slti $t1, $t0, 0 # Se il 3 valore è più piccolo di 0
        bnez $t1, prodotto # Va a prodotto
        seq $t1, $t0, $zero
        beqz $t1, somma # Va a somma se non è uguale a 0
        j andBr # Senò fa l'and
somma:
        add $v0, $s1, $s2
        j fine
prodotto:
        mul $v0, $s1, $s2
        j fine
andBr:
        and $v0, $s1, $s2
        j fine
fine:
        sw $v0, 0($s0) # Prendo il risultato della funzione e lo metto nello stack

```

Dato il seguente frammento di codice:

```
.data 0x10010000
```

Quale valore assume l'etichetta "t"?

```
ciao:
```

```
.ascii "Ciao Ciao."
```

Quale valore assume l'etichetta "eol"?

```
mamma:
```

```
.ascii "Mamma"
```

```
t:
```

```
.space 6
```

```
eol:
```

```
.ascii "1"
```

Okay che esercizio del cazzo

Conti i caratteri. Ciao Ciao. = 10

Siccome è ascii, il compilatore aggiunge un \n quindi 11

Mamma = 5

Quindi 16

E quindi l'indirizzo di t sarà  $100100000_2 + 16_{10}$

Per eol basta aggiungere 6