

```

1 #conteggio dei bit a 0 in un long
2
3 .GLOBAL _main
4
5 .DATA
6 # Abbiamo bisogno di un long per contare i dati e uno per svolgere il conteggio
7 dato:      .LONG 0xF0F0F0101
8 conteggio: .BYTE 0x00
9
10 .TEXT
11 # Inizializziamo i registri
12 # Definiamo un ciclo poichè dovremo analizzare tutta la variabile.
13 # Si fa il confronto per verificare se EAX è nullo. A un certo punto avremo solo bit
    uguali a 0
14 # Utilizzo il registro DL per salvare tutte le volte il valore del CF (con la shift
    spostiamo sempre il bit "espulso" nel carry flag): resetto DL e incremento con ADC
    (ovviamente mi interessa solo il CF).
15 # Se il valore del flag è diverso da 0 (e quindi uguale ad 1) salto l'incremento
16
17 _main:  NOP
18         MOVB $0x00, %CL
19         MOVL dato, %EAX
20
21 comp:   CMPL $0x00, %EAX      # while(eax != 0) {
22         JE fine
23
24         SHRL %EAX            # bit = eax[0]
25         MOVB $0x00, %DL      # DL = 0
26         ADCB $0x00, %DL      # DL = DL + bit
27         CMPB $0x00, %DL      # if(DL == 0) cl++;
28         JNE step
29         INC %CL
30
31 step:   JMP comp              # }
32
33 fine:   MOVB %CL, conteggio
34         RET
35

```